KIET
GROUP OF INSTITUTIONS
Connecting Life with Learning

A Project Report

on

Deceptive Content Analysis

submitted as partial fulfillment for the award of

BACHELOR OF TECHNOLOGY DEGREE

SESSION 2022-23

in

Computer Science and Engineering

By

Utkarsh Midha (1900290100181)

Ujjwal Kumar (1900290100178)

Tripti Saloni (1900290100172)

Under the supervision of

Prof. Hriday Kumar Gupta

KIET Group of Institutions, Ghaziabad

Affiliated to

Dr. A.P.J. Abdul Kalam Technical University, Lucknow

(Formerly UPTU)

May, 2023

DECLARATION

We hereby declare that this submission is our own work and that, to the best of our knowledge and belief, it contains no material previously published or written by another person nor material which to a substantial extent has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgment has been made in the text.

Signature

Name: Utkarsh Midha, Ujjwal Kumar, Tripti Saloni

Roll No.: 1900290100181, 1900290100178, 1900290100172

Date: 26/05/2023

CERTIFICATE

This is to certify that Project Report entitled “Deceptive Content Analysis” which is submitted by Utkarsh Midha, Ujjwal Kumar & Tripti Saloni in partial fulfillment of the requirement for the award of degree B. Tech. in Department of Computer Science & Engineering of Dr. A.P.J. Abdul Kalam Technical University, Lucknow is a record of the candidates own work carried out by them under my supervision. The matter embodied in this report is original and has not been submitted for the award of any other degree.

.

26/05/2023 Prof. Hriday Kumar Gupta

Date: **Supervisor Name**

(Designation)

ACKNOWLEDGEMENT

It gives us a great sense of pleasure to present the report of the B. Tech Project undertaken during B. Tech. Final Year. We owe special debt of gratitude to Prof. Hriday Kumar Gupta, Department of Computer Science & Engineering, KIET, Ghaziabad, for his constant support and guidance throughout the course of our work. His sincerity, thoroughness and perseverance have been a constant source of inspiration for us. It is only his cognizant efforts that our endeavors have seen light of the day.

We also take the opportunity to acknowledge the contribution of Dr. Vineet Sharma, Head of the Department of Computer Science & Engineering, KIET, Ghaziabad, for his full support and assistance during the development of the project. We also do not like to miss the opportunity to acknowledge the contribution of all the faculty members of the department for their kind assistance and cooperation during the development of our project.

We also do not like to miss the opportunity to acknowledge the contribution of all faculty members, especially faculty/industry person/any person, of the department for their kind assistance and cooperation during the development of our project. Last but not the least, we acknowledge our friends for their contribution in the completion of the project.

Date: 26/05/2023
Signature:
Name : Utkarsh Midha, Ujjwal Kumar, Tripti Saloni
Roll No.: 1900290100181, 1900290100178, 1900290100172

ABSTRACT

The spreading of misinformation on the web nowadays represents a serious issue, as its influence on people's opinions may be significant. Fake news represents a specific type of misinformation. While its detection was mostly performed manually in the past, automated methods using machine learning and related fields became more critical. On the other hand, deep learning methods became very popular and frequently used methods in the field of data analysis in recent years. Our main idea was to train different types of neural network models using both entire texts from the articles and to use just the title text. In this study, we tried to accomplish that using LSTM and natural language processing. There are lots of machine learning models but these two have shown better progress. The models were trained and evaluated on the Fake News dataset obtained from the Kaggle competition.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

LSTM	Long Short Term Memory
RNN	Recurrent Neural Network
CNN	Convolutional Neural Network
LISW	Linguistic Inquiry and Word Count
SNS	Social Networking Services
RST	Rhetorical Structure Theory
MVAE	Multimodal Variational Autoencoder
HSDF	Hierarchical Discourse-level Structure
POS	Part-of-speech

CHAPTER 1
INTRODUCTION

1.1 INTRODUCTION

Fake news is untrue information presented as news. It often has the aim of damaging the reputation of a person or entity or making money through advertising revenue. Once common in print, the prevalence of fake news has increased with the rise of social media, especially the Facebook News Feed. During the 2016 US presidential election, various kinds of fake news about the candidates widely spread in the online social networks, which may have a significant effect on the election results. According to a post-election statistical report, online social networks account for more than 41.8% of the fake news data traffic in the election, which is much greater than the data traffic shares of both traditional TV/radio/print medium and online search engines respectively. Fake news detection is becoming increasingly difficult because people who have ill intentions are writing the fake pieces so convincingly that it is difficult to separate from real news. What we have done is a simplistic approach that looks at the news headlines and tries to predict whether they may be fake or not. Fake news can be intimidating as they attract more audience than normal. People use them because this can be a very good marketing strategy. But the money earned might not live up to fact that it can harm people.

1.2 PROJECT DESCRIPTION

The main objective is to detect fake news, which is a classic text classification problem with a straightforward proposition, Improving the efficiency of the existing model with the help of Deep Learning and NLP. To overcome the issue of Fake News we will implement our idea on a website to differentiate between “Real” News and “Fake” News.

CHAPTER 2

LITERATURE REVIEW

2.1 Fake News Detection using RNN-LSTM

Disseminating false information through fake news aims to deceive people. The prevalence of fake news is a societal issue that cannot be completely eradicated, whether on an individual level or through popular web-based social platforms like Facebook and Twitter. Fake news plays a significant role in inciting riots, promoting violence, causing mass lynching incidents, and triggering various social and economic disruptions. Fabricated news articles can intentionally mislead readers, promote one-sided perspectives, serve specific agendas, or simply provide entertainment. It's important to acknowledge that this pervasive problem affects individuals worldwide and has persisted throughout history. Detecting false information often relies on advanced deep learning techniques like Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNNs). One specific type of RNN known as Long Short-Term Memory (LSTM) excels at analyzing sequential data to identify patterns associated with false information.

The BI-DIRECTIONAL LSTM-RNN model is employed for detecting spurious news by scrutinizing the constructed headline of a news article and establishing the correlation between the headline and the body text to appraise the veracity of the content.

- A Recurrent Neural Network (RNN) is a type of neural network that assimilates antecedent information as input for the current prognostication. RNNs encompass a mnemonic component to uphold preceding learning, and a recurrent covert layer is engaged to handle variables whose activations hinge upon antecedent time steps.

- Long Short-Term Memory (LSTM) is a variant of RNNs that proficiently captures protracted dependencies. LSTM algorithms are particularly adept at mitigating the predicament of the vanishing gradient, which can impede learning in intricate networks.

- Bi-directional LSTM-RNN models are invaluable in perpetuating information from both antecedent and subsequent time steps and across manifold strata in a profound network. They are highly suitable for endeavors like voluminous-scale sequential text prediction and text categorization.

The news reports are initially subjected to comprehensive pre-processing. Each news article is assigned a binary value, with 1 denoting fake news and 0 indicating genuine news. The two

columns, 'title' and 'text,' are merged to create a new column, facilitating immediate pre-processing. The NLTK Porter stemmer is employed to eliminate punctuation and stop words from the input news articles, while gensim filters out words consisting of fewer than two characters. The titles and content of the news reports are transformed into padded sequences of words, delimited by spaces. Tokenization is then applied to further break down these sequences into lists of individual tokens. The resulting vector representation of the data is subsequently partitioned into training, validation, and testing datasets, with a test size of 0.2. The training process of the models is based on a comprehensive collection of diverse news stories. The validation dataset is crucial for fine-tuning the models, while the trained models are deployed to predict the labels of news articles using the test data. Two distinct models, namely LSTM and Bi-LSTM, are trained, enabling a thorough comparison between their performances.

As a result, the prediction mechanism is effectively implemented through the utilization of advanced RNN techniques, delivering highly accurate predictions of fake news based on the input variables. The proposed approach incorporates various elements, including the application of NLTK Porter stemming for comprehensive natural language processing (NLP), word tokenization, word embedding techniques, and the incorporation of both unidirectional and bidirectional LSTM models within the RNN framework.

2.2 A Survey on Natural Language Processing for Fake News Detection

Fact-checking involves evaluating the accuracy of statements made by public figures such as politicians and pundits (Vlachos and Riedel, 2014). The distinction between fake news detection and fact-checking is often blurred since both aim to assess the truthfulness of claims. While fake news detection primarily focuses on news events, fact-checking has a broader scope. Thorne and Vlachos (2018) provide a comprehensive overview of this topic. In many studies, fake news detection is approached as either a classification or regression problem.

Classification:

One common approach is to treat fake news detection as a binary classification problem. However, accurately categorizing news into two classes (fake or real) is challenging due to cases where news contains a mix of factual and false information.

Regression:

Fake news detection can also be framed as a regression task, where the output is a numerical score indicating the level of truthfulness. Evaluation is typically performed by comparing predicted scores with ground truth scores or employing Pearson/Spearman correlations.

Datasets:

An important challenge in automated fake news detection lies in the availability and quality of datasets. Public fake-news datasets can be categorized into three types: claims, entire articles, and Social Networking Services (SNS) data. Claims consist of one or a few sentences that require validation. Entire articles encompass multiple related sentences constituting comprehensive information. SNS data, similar in length to claims, includes structured data from accounts and posts, often containing non-textual information.

Methods for Fake News Detection:

The initial step involves preprocessing, including tokenization, stemming, and word generalization or weighting. Techniques like Term Frequency-Inverse Document Frequency (TF-IDF) and Linguistic Inquiry and Word Count (LIWC) are commonly used to convert tokenized texts into features. Supervised learning methods are predominantly employed in existing research, with a focus on models such as Support Vector Machine (SVM) and Naive Bayes Classifier (NBC) for classification.

Multi-class Fake News Detection framework (MMFD):

This framework combines Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) models. CNN analyzes local patterns within claim texts, while LSTM captures temporal dependencies in the entire text. The final hidden outputs of both models are concatenated and passed through a Fully Connected Network. Attention mechanisms are integrated to enhance accuracy. This model leverages the strengths of both CNN and LSTM, with LSTM performing better on longer sentences.

Rhetorical Structure Theory (RST):

RST, sometimes in conjunction with the Vector Space Model (VSM), is utilized for fake news detection. RST is an analytical framework for examining the coherence of a story. By defining the semantic roles of text units, such as Circumstance, Evidence, and Purpose, this framework systematically identifies the central idea and analyzes the characteristics of the input text to detect fake news based on coherence and structure.

Collecting Evidence:

The Recognizing Textual Entailment (RTE) method is commonly employed to gather and utilize evidence. RTE involves recognizing relationships between sentences. By employing RTE methods to gather sentences that support or oppose the input from sources such as news articles, it becomes possible to predict the correctness of the input.

Nine Requirements for Fake News Detection Corpus:

1. Availability of both truthful and deceptive instances.
2. Accessibility of digital textual formats.
3. Verifiability of "ground truth" labels.
4. Homogeneity in article lengths.
5. Consistency in writing style.
6. Predefined timeframe for news collection.
7. Consideration of the manner of news delivery.
8. Pragmatic concerns in dataset construction.
9. Consideration for language and cultural differences.

Based on observations, several recommendations are proposed for future datasets. These recommendations aim to address the challenges and limitations in fake news detection:

I. Combine hand-crafted features with neural network models: Investigate the effectiveness of integrating hand-crafted features, such as linguistic and contextual information, with neural network models. This combination can potentially enhance the performance and interpretability of fake news detection systems.

II. Incorporate non-textual data: Explore the inclusion of non-textual data, such as metadata, user engagement metrics, or visual content, to complement the textual information. Non-textual data can provide valuable cues for detecting fake news and improving the overall accuracy of the models.

III. Extend verification methods: Expand the methods used for verification by considering additional sources and types of evidence. For example, beyond relying solely on textual entailment, incorporate other techniques like image analysis, fact-checking databases, or domain-specific knowledge bases to gather more diverse and comprehensive evidence.

IV. Enhance language and culture considerations: Take into account the linguistic and cultural variations that exist in different regions and languages. Fake news detection models should be adaptable and robust across diverse languages and cultural contexts to ensure their effectiveness in global settings.

To achieve a comprehensive and accurate fake news detection system, it is crucial to continuously explore and refine these recommendations while advancing the understanding of the intricate dynamics of misinformation.

2.3 The Role of Social Context for Fake News Detection

The rapid growth of social media platforms has transformed the way people consume news, thanks to their convenience, accessibility, and affordability. However, this widespread adoption has also led to a surge in the dissemination of fake news, which refers to deliberately misleading information presented as news. Detecting and combating fake news has become crucial not only to ensure the public receives accurate information but also to maintain the integrity of the news ecosystem.

Nowadays, social media has surpassed traditional news outlets as the primary source of news for many individuals. Unfortunately, this shift has given rise to a proliferation of fake news across various online platforms, driven by motives such as financial gain and political manipulation.

The impact of fake news can be highly detrimental to both individuals and society at large. Firstly, it can mislead people, causing them to believe and spread false information. This distortion of reality can have far-reaching consequences. Secondly, the presence of fake news can influence public perception and attitudes towards genuine news, leading to a skewed understanding of important events and issues. Lastly, the widespread circulation of fake news erodes trust in the overall news ecosystem, undermining the credibility of reliable sources of information.

Therefore, it is imperative to develop effective strategies to detect and combat fake news on social media platforms. This paper focuses on two key aspects: (1) the development of a mathematical framework to model the complex relationships involved in fake news detection and extract meaningful features from news articles, and (2) leveraging this framework to enhance the accuracy of fake news detection.

The proposed framework, known as TriFN, comprises five major components: the news content embedding module, the user embedding module, the user-news interaction embedding module, the publisher-news relation embedding module, and the semi-supervised classification module.

The news content embedding module aims to transform the textual content of news articles into a latent feature space, capturing more nuanced and informative representations. Similarly, the user embedding module extracts latent features from user social interactions, considering the impact of user characteristics on their engagement with news content. The user-news interaction embedding module leverages partial labels and user credibility to learn feature representations that capture the dynamic interactions between users and news articles. The publisher-news relation embedding module incorporates publisher partisan bias labels to regularize the feature representations of news articles. Finally, the semi-supervised

classification module trains a classification function to predict the labels of unlabeled news articles, utilizing the available labeled data and the learned feature representations.

By integrating these components, the TriFN framework offers a comprehensive approach to fake news detection, leveraging the intricate relationships between news content, users, and publishers. This holistic approach aims to enhance the accuracy and effectiveness of fake news detection on social media platforms, thereby safeguarding the integrity of the news ecosystem.

Ensuring the accuracy and reliability of news shared on social media platforms is crucial, as the widespread dissemination of fake news can misinform and deceive users. Evaluating the effectiveness of fake news detection algorithms requires the use of well-established metrics such as Accuracy, Precision, Recall, and F1 score, commonly employed in evaluating classifiers in related domains. This study aims to compare the performance of the innovative TriFN framework, designed for detecting fake news, with several state-of-the-art methods.

To extract meaningful features for the identification of fake news, we employ a range of representative techniques, including the utilization of Rhetorical Structure Theory (RST). RST constructs a hierarchical tree-like structure that captures the rhetorical relations between words within a text, facilitating a deeper understanding of the underlying narrative. Additionally, the Linguistic Inquiry and Word Count (LIWC) method is employed to extract lexicons categorized by psycholinguistic characteristics, providing valuable insights into the linguistic patterns associated with fake news. Furthermore, we incorporate the Castillo approach, which focuses on extracting diverse features from users who have shared specific news items on social media. This contextual information sheds light on user behavior and engagement patterns, thereby aiding in the detection process.

In our investigation, we also explore the performance of combined feature sets, namely RST+Castillo and LIWC+Castillo, which leverage both the content of news articles and the social context in which they are shared. By integrating these diverse feature representations, we aim to enhance the accuracy and efficacy of fake news detection.

Additionally, we emphasize the significance of early detection in combating the spread of fake news. Early detection algorithms consider a limited social context within a specific timeframe from the original news posting, providing timely alerts and minimizing the dissemination of fake news on social media platforms.

Our evaluation yields noteworthy observations. Firstly, we find a positive correlation between the duration of time since the original news posting and the performance improvement of algorithms that utilize social context information. This indicates that increased social engagements and interactions on social media platforms provide valuable signals for the detection of fake news. Secondly, our proposed TriFN framework consistently outperforms other methods in terms of accuracy and F1 score on both datasets. The incorporation of user-

news interactions within the TriFN framework proves to be a crucial factor in capturing meaningful feature representations. Lastly, even in the early stages following the publication of fake news, TriFN demonstrates commendable performance, showcasing its effectiveness in detecting fake news at its inception.

Looking ahead, there are several promising avenues for future research. Firstly, exploring innovative approaches to early detection of fake news is imperative due to the dynamic nature of its propagation on social media. Secondly, there is a need to delve deeper into the psychological aspects and intentions behind the creation and dissemination of fake news, with a focus on developing robust methods for extracting features that capture these nuances. Lastly, identifying and addressing the involvement of low-quality or malicious users in the spread of fake news is a critical step towards implementing effective intervention and mitigation strategies.

2.4 Deep Learning and Natural Language Processing for fake news detection

False information, misleading content, and distorted facts are all part of the deceptive realm known as fake news. It is a deliberate tactic employed to manipulate public opinion, gain personal advantages, or incite discord among people. Shockingly, a recent study revealed that an overwhelming 86% of individuals have fallen victim to fake news, with the social media giant Facebook being the primary platform for its dissemination. One notable incident where fake news had severe repercussions was during the Citizen Amendment Act (CAA) in 2020. A team of dedicated journalists conducted a survey to shed light on the extent of the issue.

Models:

- **N-grams language models**

This method reduces the computing potential but can also lead to the tokenization missing some of the characteristics in the dataset before training the deep learning model.

- **Gated Recurrent Units**

GRUs which are the newest breed of the memory elements used in the hidden layers are a revision of LSTM modules containing fewer gates, sigmoid and hyperbolic tangent activations which makes it computationally faster than LSTM.

- **M.L. Models**

Different machine learning models have also been used for this matter like the use of naïve bayes which provided an accuracy of 74 %. Deep learning models have emerged as a breakthrough in combating the limitations of machine learning techniques when it comes to detecting fake news. One of the primary challenges faced by machine learning frameworks in this domain is the need for a substantial dataset, which often leads to issues of overfitting. However, deep learning models have revolutionized the landscape by leveraging sophisticated neural networks to extract intricate features automatically. This enables them to effectively categorize and discern fake news while avoiding the pitfalls of overfitting. By harnessing the power of deep learning, researchers and practitioners have significantly enhanced their ability to identify and combat the proliferation of false information in today's digital age.

Methods

For fake news classification technique different deep learning architectures are used each possessing different characteristics as compared to the other architectures and each possessing its own set of distinct parameters.

✧ Artificial Neural Networks

In artificial neural network weighted connection is made between different layers of the neural network namely input layer consisting of several nodes where the preprocessed news vector of certain dimension is passed as an input to the neural network, Second is the hidden layer which contains the neurons that are responsible for performing calculations to give the desired output which is the binary variable indicating 0 for the news being real or 1 for the news being fake or vice versa. The output layer, which constitutes the third layer of the neural network architecture, plays a vital role in presenting the processed information obtained from the preceding hidden layers. It serves as the final stage where the results or outputs of the neural network are displayed. This layer is responsible for showcasing the outcomes of the computations performed within the neural network, which can take various forms depending on the specific application. These outputs may encompass predictions, classifications, or any other relevant information derived from the network's learning process. Ultimately, the output layer serves as a crucial component in delivering the final output of the neural network, making it accessible and meaningful for subsequent analysis or decision-making purposes.

There are two modes in which training of a neural network is done:

Forward propagation: The neural network operates in direction from the input layer to output layer to calculate the predicted output.

Backward propagation: During the training period the observed output that is the result whether the news is fake or not is compared with the actual output and error is calculated with the help of certain loss function and the neural network operates in the direction from the output layer to the input layer and updates the weights and bias in order to achieve better predictions.

✧ Convolutional Neural Network

They are useful in classification of the pictures but using them with word embedding helps to predict amazing results as these embedding techniques convert the input vector as a matrix of features which is same in the case of images where pixels are stored and processed in the form of a matrix.

Convolutional networks, also known as Convolutional Neural Networks (CNNs), operate differently from Artificial Neural Networks (ANNs) by incorporating additional layers that enhance their performance in tasks such as image recognition and computer vision.

One of the key components in a CNN is the convolutional layer. This layer applies convolution operations using filters or kernels to extract local features and capture spatial

patterns within the input data. By sliding these filters across the input, the convolutional layer learns to identify important features such as edges, textures, and shapes in an image.

Along with the convolutional layer, CNNs include a pooling layer. This layer helps reduce the spatial dimensions of the feature maps, down sampling the data and extracting the most relevant information from different regions. Pooling operations like max pooling or average pooling select the most prominent features, ensuring computational efficiency and allowing the network to be invariant to small spatial variations.

After pooling, the data is passed through a flattening layer, which reshapes the multidimensional feature maps into a one-dimensional vector. This step is crucial for connecting the output of the convolutional layers to the subsequent fully connected layers.

The final stage involves fully connected layers, similar to those found in traditional ANNs. These layers employ weights and biases to perform computations and make predictions based on the learned features. Fully connected layers capture higher-level abstractions and relationships in the data, enabling complex decision-making and classification tasks.

In summary, convolutional networks extend the capabilities of ANNs by incorporating specialized layers like convolutional, pooling, and fully connected layers. These components allow CNNs to effectively learn and extract features from input data, particularly in the realm of image analysis and computer vision.

✧ **Recurrent Neural Network**

It is similar to ANN but these neural networks are cyclic in nature. For getting an output from the nodes of the hidden layer two things are taken in consideration: First is the input to that node of the hidden layer and the second layer of a neural network is known as the hidden layer, positioned between the input layer and the output layer. It operates as an intermediary layer that processes input data and produces hidden representations or features.

Comprising a collection of neurons or nodes, the hidden layer performs computations on the input data using weighted connections and activation functions. Each neuron in the hidden layer receives input from the previous layer, computes a weighted sum of these inputs, and applies an activation function to generate an output. These outputs, often referred to as hidden states, are then passed on as inputs to the subsequent layer.

The primary purpose of the hidden layer is to capture and learn intricate patterns, complex relationships, and important features present in the input data. By employing multiple hidden layers, a neural network can progressively extract higher-level abstractions and make more sophisticated predictions.

The hidden state, which corresponds to the output obtained from the hidden layer, represents a transformed representation of the input data. It encapsulates encoded information that has undergone processing by the hidden layer, capturing relevant patterns and significant characteristics for the given task. This hidden state serves as the input for the subsequent layer, enabling a sequential and hierarchical processing of the data.

To summarize, the hidden layer of a neural network plays a crucial role in processing input data, extracting meaningful features, and generating hidden states. These hidden states, which contain valuable information, are then utilized as inputs for subsequent layers, ultimately contributing to the final output of the network. There are different architectures of the recurrent neural network like many to many architectures where there are many inputs that is inputs of variable length which generate an output of variable length which may or may not be similar to the length of the input.

One major problem while using deep RNN for classification of binary output is the vanishing and exploding gradient problem. Generally, the fake news classification encounters the vanishing gradient problem due to its variability of input (many to one architecture) and large input news vectors.

✧ CNN with LSTM architecture

Using memory elements along with the convolutional neural network incorporates the features of both the architecture namely CNN and RNN. For getting the binary output to determine whether the news is hoax or not, remembering the long term patterns along with eradicating and captivating some important features using the layers of CNN can be a strong and helpful tool. The main benefit of using this compound framework is that the LSTM gets to remember smaller number of words or time steps because the 4 layers of the CNN does all the filtering steps which acts like a pre processing module to the fully connected layer with memory elements present inside them.

In the realm of fake news classification, this research explores diverse architectures and methodologies. To address long-term dependencies, the inclusion of long short-term memory (LSTM) in the hidden layer of artificial neural networks is crucial. Combining LSTM with convolutional neural networks (CNN) boosts model accuracy, while incorporating dropout layers in hidden or fully connected layers enhances predictions of news content. When applied to image classification of news on social media, the fusion of CNN and LSTM produces promising results. Furthermore, bidirectional recurrent neural networks, equipped with LSTM, excel in capturing dependencies within extensive inputs, making them valuable for identifying fabricated news. Employing the softmax activation function allows estimation of the probabilities indicating the authenticity of news.

2.5 Understanding User Profiles on Social Media for Fake News Detection

Fake News Detection on Social Media The detection of fake news involves the analysis of both news content and social context. Fake news detection methods primarily focus on extracting features from these two sources. Linguistic features, such as lexical and syntactic elements, are used to capture unique writing styles and attention-grabbing headlines commonly found in fake news articles. Visual features, on the other hand, help identify manipulated images and identify specific visual characteristics associated with fake news. Additionally, social context plays a crucial role in fake news detection, as it involves examining user interactions, propagation patterns, and source credibility. By combining these diverse features, effective algorithms can be developed to combat the dissemination of fake news. **Measuring User Profiles on Social Media** User profiles can be broadly categorized into explicit and implicit features. Explicit features are readily available in user metadata, while implicit features provide valuable insights for specific tasks. We begin by introducing datasets containing news and user engagements, followed by the identification of user communities based on trust levels in news. Two datasets are created, encompassing news content and social context information.

Continuing with our investigation into Research Question 1, we aim to identify user subsets that exhibit different degrees of trust in fake and real news. By identifying these subsets, we can compare their profiles and extract useful profile features. User profile features are collected and analyzed from explicit and implicit aspects. Explicit features are obtained directly from social media site APIs, including follower count, verification status, and registration details. Implicit features, inferred from user metadata and online behaviors like historical tweets, include factors such as gender, age, and personality.

2.6 Exploring the Role of Visual Content in Fake News Detection

The proliferation of fabricated information on social media platforms has sparked a significant concern, leading to extensive research in the field of identifying and classifying fake news. To tackle the challenges associated with detecting deceptive content, various architectures and methodologies have been proposed. Among these approaches, the integration of deep learning models, particularly artificial neural networks (ANNs), has proven effective in overcoming the limitations of traditional machine learning models. ANNs enable handling large datasets required for accurate classification, mitigating the risk of overfitting.

To enhance the detection of fake news, researchers have explored the use of deep learning models with Long Short-Term Memory (LSTM) units incorporated in hidden layers. This combination has shown promising results, as LSTMs can capture long-term dependencies in data, particularly in Convolutional Neural Networks (CNNs). Furthermore, the inclusion of dropout layers in hidden or fully connected layers of CNNs has demonstrated improved predictive performance for news content. Additionally, the utilization of CNNs in conjunction with LSTMs for analyzing visual content in social media posts has yielded favorable outcomes. Another noteworthy approach involves bidirectional recurrent neural networks using LSTMs to handle the complexities of large input data and predict the authenticity of news. Activation functions like SoftMax are employed to estimate the probability of news being genuine or false.

However, the advancement of research in this domain is accompanied by several challenges. One significant hurdle is the scarcity of labeled data specifically dedicated to multimedia fake news. Constructing and releasing high-quality labeled datasets is crucial to drive progress in this field. Moreover, exploring weakly supervised settings and leveraging the connections between textual and visual content, as well as metadata, can enhance the detection of multimedia fake news and offer valuable insights into the decision-making process of algorithms.

Moving beyond neural networks, explicit and implicit user profile features play a vital role in various social media tasks such as assessing information credibility and linking user identities. Explicit features, derived directly from raw user metadata, are widely utilized. In contrast, implicit features, inferred from user meta information or online behaviors, provide valuable insights into user profiles for specific tasks. Analyzing explicit features such as follower count, verification status, and registration details, along with implicit features like gender, age, and personality traits, contributes to a comprehensive understanding of user profiles.

To effectively detect and classify fake news, different types of features can be considered. Forensic features focus on verifying the authenticity of visual content by detecting manipulations, compressions, and generation techniques. Semantic features examine the

distinct characteristics of fake news, such as captivating visuals and emotional provocations, which exploit individual vulnerabilities to facilitate the spread of fabricated information. Statistical features leverage the distribution patterns of visual content between fake and real news on social media platforms.

Multiple approaches have been proposed for the detection of multimedia fake news, falling into content-based and knowledge-based categories. Content-based approaches leverage cues from various modalities to identify fake news without relying on external datasets. Knowledge-based approaches, on the other hand, utilize external sources and reference datasets to fact-check and assess the integrity of news posts.

Despite the progress made in fake news detection, challenges persist, including the limited availability of labeled data and the need for robust methodologies. Constructing high-quality labeled datasets and exploring weakly supervised settings can enhance the development of effective detection models. Additionally, studying the connections between textual and visual content, as well as metadata, can provide valuable insights into the detection process and facilitate the explanation of algorithmic decisions.

In conclusion, combating the dissemination of fake news and developing reliable classification techniques require a multidimensional approach that encompasses deep learning models, user profile features, and the analysis of visual content. By leveraging these diverse methodologies and addressing the challenges in this field, we can contribute to the ongoing battle against fake news and its detrimental societal impacts.

2.7 Media-Rich Fake News Detection

The phenomenon of fake news encompasses the dissemination of misleading or false information with the intent to deceive readers. However, combatting this issue is far from simple. Automatic detection of fake news poses significant challenges due to the prevalence of multimedia content, including images and videos, which can be easily manipulated and forged. Additionally, the rise of social media platforms has made fake news easily accessible and capable of exerting a profound impact on society.

This research paper addresses multiple aspects related to the problem of detecting fake news. It provides a thorough exploration of the various platforms utilized for sharing news content, highlighting their effectiveness and extensive reach. Furthermore, the paper examines the different types of data that news articles can contain, investigating the influence of each data type on readers.

An essential focus of this paper is developing an understanding of the distinct categories of fake news. By categorizing fake news into specific groups, researchers gain valuable insights into the diverse nature of fabricated information. Additionally, the paper offers an overview of existing methods for detecting fake news, conducting a comprehensive comparison to analyze their strengths and limitations.

To facilitate further research in the field, the paper highlights the availability of existing datasets that can be utilized by researchers. These datasets play a crucial role in training and evaluating detection models, enabling the development of more robust and accurate solutions.

Moreover, the paper delves into various types of fake news, going beyond the traditional text-based approach. It explores visual-based, user-based, post-based, network-based, knowledge-based, style-based, and stance-based fake news, providing a comprehensive understanding of the complexities associated with detecting and combating fake news across different platforms and contexts.

Furthermore, the paper references several widely used fake news datasets, including Buzzfeed News, LIAR, and PHEME CREDBANK. These datasets serve as valuable resources for researchers, facilitating experimentation and serving as benchmarks for evaluating detection algorithms.

In conclusion, this paper offers a comprehensive exploration of the challenges involved in detecting fake news. By examining platform analysis, data types, fake news categories, existing detection methods, datasets, and various types of fake news, it lays a strong foundation for further research and development in this critical area.

2.8 MVAE: Multimodal Variational Auto encoder for Fake News Detection

The prevalence of fake news on social media networks poses a significant challenge in today's information landscape. With a vast number of news articles being shared daily, the lack of credibility and verification checks has given rise to an ecosystem driven by misinformation and disinformation. Detecting fake news is a complex task, particularly due to the inclusion of images and videos alongside textual content, making it easier to manipulate and deceive readers. Additionally, the widespread reach of social media platforms amplifies the impact of fake news stories.

In this paper, we delve into various aspects of fake news detection, aiming to address this pressing research problem. We provide a comprehensive overview of different platforms used for news dissemination, highlighting their effectiveness and wide usage. We examine the types of data found in news articles and discuss their respective impacts on readers. Understanding the various categories of fake news is also a crucial aspect covered in our study.

To tackle the challenge of fake news detection, we review existing methods and compare them from different perspectives. We also explore available datasets that researchers can utilize for their investigations into fake news detection. Furthermore, we analyze different types of fake news, such as those based on visuals, users, posts, networks, knowledge, style, and stance. Notably, we discuss notable fake news datasets, including Buzzfeed News, LIAR, and PHEME CREDBANK.

Inspired by the concept of autoencoders, we propose a novel approach for fake news detection that leverages a multimodal framework. Our model, the Multimodal Variational Autoencoder (MVAE), aims to learn shared representations by integrating visual and textual information. This enables the discovery of correlations across modalities within tweets. We present a detailed description of the MVAE, including its various components such as the encoder, decoder, and fake news detector. The encoder encodes information from both text and image modalities into a latent vector, while the decoder reconstructs the original image and text from this vector. The fake news detector utilizes the learned shared representation to predict the authenticity of news content. To optimize the MVAE, both the VAE and the fake news detector are trained jointly.

We evaluate the performance of our proposed model on real-world datasets and compare it to baseline models. The MVAE demonstrates superior performance, surpassing existing architectures in terms of accuracy and F1 scores. Notably, the attention mechanism employed in the MVAE, represented by the att-RNN model, significantly improves the model's

performance by considering relevant visual features related to the text. These findings highlight the effectiveness of our proposed approach in detecting fake news.

Despite these achievements, we acknowledge the scarcity of labeled data for fake news detection, which poses a challenge for further advancements in this research field. We encourage researchers to focus on constructing and releasing high-quality labeled datasets. Additionally, we emphasize the importance of studying fake news detection in a weakly supervised setting, leveraging the relationship between textual and visual content along with metadata to make more informed inferences.

In conclusion, our study contributes to the ongoing efforts in combating fake news by proposing the MVAE model and leveraging multimodal representations. By integrating visual and textual information and discovering correlations across modalities, our approach shows promising results in detecting fake news. We believe that our findings can contribute to the development of more effective techniques to address the pervasive issue of fake news in today's media landscape.

2.9 Learning Hierarchical Discourse-level Structure for Fake News Detection

In today's digital landscape, the rapid dissemination of false news articles through online platforms poses a significant threat to information reliability. However, our understanding of the linguistic characteristics of fake news remains limited. To address this issue, we propose the Hierarchical Discourse-level Structure for Fake News Detection (HDSF) framework, which aims to capture the hierarchical structure of news articles. By analyzing the discourse-level structure, we can gain insights into how fake and real news articles are organized.

The HDSF framework, described in this study, embraces the opportunities and challenges associated with understanding the structural aspects of news articles. We introduce several structure-related properties that allow us to analyze the hierarchical structures of news articles. Our framework leverages automated dependency parsing techniques at the sentence level to learn the hierarchical structure of a document.

To construct the hierarchical structure between discourse units, we utilize dependency parsing techniques that do not require annotated data. The resulting structure is represented as a dependency tree, which captures the relationships between different discourse units. Additionally, we extract a structurally rich representation for the entire document, incorporating a method similar to previous studies. Each sentence is assigned a representation that considers its structural characteristics. We identify three fundamental properties of the constructed discourse dependency trees: the number of leaf nodes, the preorder difference, and the parent-child distance.

Through our experiments, we observe a progressive decrease in the training error as the training process proceeds. Furthermore, the accuracy on the development set consistently improves during training, indicating that the framework optimizes its performance and learns to classify fake news documents accurately. Real news documents exhibit a higher degree of coherence in their hierarchical structures compared to fake news documents.

This study opens up new avenues for future research. We plan to explore more advanced properties derived from the discourse dependency trees to enhance the framework's performance. Additionally, investigating the hierarchical structure at the word-level presents an exciting research direction. Furthermore, we aim to investigate unsupervised methods for extracting discourse-level structures from fake and real news documents.

By delving into the hierarchical discourse-level structure of news articles, we aim to deepen our understanding of fake news and enhance the effectiveness of fake news detection. The insights gained from this research can contribute to combating the spread of misinformation in online media platforms.

2.10 Bidirectional LSTM Based on POS tags and CNN Architecture for Fake News Detection

Presently, the proliferation of false news articles across diverse online platforms poses a significant threat to the reliability of information. False news encompasses fabricated content intentionally generated to deceive readers. This definition emphasizes two critical aspects: authenticity and intent. Firstly, false news disseminates inaccurate information that can be verified as such. Secondly, it is purposefully crafted to mislead consumers. A broader perspective on false news examines either the authenticity or intent of the news content. False news predominantly spreads through traditional print and broadcast media, as well as online social media channels. Unethical practices, such as financial incentives for reporters, often underlie the creation and propagation of false news. Sensationalized headlines are commonly employed to boost readership and manipulate public opinion.

Part-of-speech (POS) tagging serves as a fundamental step in numerous Natural Language Processing (NLP) tasks, as it assigns appropriate POS tags to words in a text. These tags provide valuable insights into the syntactic categories of words, such as nouns, verbs, adjectives, adverbs, and more. Additionally, POS tagging facilitates analyses of similarities and dissimilarities in texts. Notably, extensive research on large Twitter datasets employing POS tags has revealed distinct distributions of positive, negative, and neutral texts. Convolutional neural networks (CNNs), akin to Support Vector Machines (SVMs) and logistic regression, have emerged as standard baseline models in computer vision tasks, including visual question answering, image classification, and image captioning. In the field of natural language processing, CNNs have demonstrated remarkable performance in sentence classification tasks.

Recurrent neural networks (RNNs) process input sequences by iteratively updating weight matrices through recursive function calls. At each time step, the recursive function incorporates the current input and the previous hidden state. However, RNNs face the challenge of the exploding or vanishing gradient problem during training, where the gradient either grows or decays exponentially. This limitation hinders RNNs' ability to capture long-range dependencies in textual input.

Two frequently employed variants of Long Short-Term Memory (LSTM) networks are bidirectional LSTM and multi-layer LSTM. A bidirectional LSTM consists of two LSTM units: one processes the input from left to right, while the other operates from right to left within the input sequence. Initially devised for image classification, convolutional neural networks have exhibited exceptional performance in tasks like object detection. They employ various operations, including convolution and pooling. The integration of independent CNN and LSTM models in an efficient manner can yield superior overall performance.

CHAPTER 3

PROPOSED METHODOLOGY

There are two parts to the ML Model building. Machine Learning is a part of our life that can help us in predicting. We are using two types of models in this case.

For the first part, we used LSTM. And the steps include:

1. **Loading the data:** We are loading a CSV file for the data sorting and training-testing part of the model. The CSV file is turned into an array for easier work purpose.
2. **Scanning and parsing**: Data is loaded from a CSV file. This consists of the body of selected news articles. It then contains a label field that indicates whether the news is real or fake. In this code block, we scan the CSV and clean the titles to filter out stop words and punctuation.
3. **Tokenization:** The tokenizer is used to assign indices to words, and filter out infrequent words. This allows us to generate sequences for our training and testing data.
4. **Embedding matrix:** Apply the embedding matrix. An embedding matrix is used to extract the semantic information from the words in each title.
5. **Model Building:** Building the model and finding out the accuracy via confusion matrix. The model is created using an Embedding layer, LSTM, Dropout, and Dense layers.

In the second part, we used is Natural Language Processing (NLP) using Logistic Regression. Here are the steps:

1. **Preparing The Data - Text Pre-processing**

   Cleaning and Normalization: Removing unnecessary elements and standardizing the text.

Stop Words Removal: Eliminating commonly occurring words with low semantic value.

Stemming and Lemmatization: Reducing words to their base or root form.

Tokenization: Breaking down the text into smaller chunks (words, sentences, etc.).

Other Pre-processing Steps: Removing special characters, punctuation, and numbers; handling capitalization and case sensitivity.

Stop Words: Understanding the role of stop words in natural language processing and techniques for their removal.

By following these data preparation steps, the raw text can be transformed into a processed format suitable for further analysis and modeling in natural language processing tasks. Each step plays a crucial role in cleaning, normalizing, and organizing the text data to ensure accurate and meaningful results in subsequent NLP tasks.

2. **Building the model**
3. **Fitting the model - Logistic Regression**

3.1 Requirement Analysis

3.1.1 Functional Requirements

The functions of software systems are defined in functional requirements and the behavior of the system is evaluated when presented with specific inputs or conditions which may include calculations, data manipulation and processing and other specific functionality.

- Our system should be able to read the data and preprocess data.
- It should be able to analyze the fake data.

- It should be able to group data based on hidden patterns.
- It should be able to assign a label based on its data groups.
- It should be able to split data into train set and test set.
- It should be able to train model using train set.
- It must validate trained model using test set.
- It should be able to classify the fake and real data.

3.1.2 NON-FUNCTIONAL REQUIREMENTS

Non-functional requirements illustrate how a system must behave and create constraints of its functionality. This type of constraints is also known as the system's quality features. Attributes such as performance, security, usability, compatibility are not the feature of the system, they are a required characteristic. They are "developing" properties that emerge from the whole arrangement and hence we can't compose a particular line of code to execute them. Any attributes required by the user are described by the specification. We must contain only those needs that are appropriate for our design.

Some Non-Functional Requirements are as follows:

- Reliability
- Maintainability
- Performance
- Portability
- Scalability
- Flexibility

ACCESSIBILITY:

Availability is a general term used to depict how much an item, gadget, administration, or condition is open by however many individuals as would be prudent. In our venture individuals who have enrolled with the cloud can get to the cloud to store and recover their information with the assistance of a mystery key sent to their email ids. UI is straightforward and productive and simple to utilize.

MAINTAINABILITY:

In programming designing, viability is the simplicity with which a product item can be altered as:

- Correct absconds
- Meet new necessities

New functionalities can be included in the task based the client necessities just by adding the proper documents to existing venture utilizing ASP. Net and C# programming dialects. Since the writing computer programs is extremely straightforward, it is simpler to discover and address the imperfections and to roll out the improvements in the undertaking.

SCALABILITY:

Framework is fit for taking care of increment all out throughput under an expanded burden when assets (commonly equipment) are included. Framework can work ordinarily under circumstances, for example, low data transfer capacity and substantial number of clients

PORTABILITY:

Portability is one of the key ideas of abnormal state programming. Convenient is the product code base component to have the capacity to reuse the current code as opposed to making new code while moving programming from a domain to another. Venture can be executed under various activity conditions gave it meet its base setups. Just framework records congregations would need to be designed in such case.

3.1.3 HARDWARE REQUIREMENTS

- Processor : Any Processor above 500 MHz
- RAM : 4 GB
- Hard Disk : 500 GB
- System : Intel i3 6gen 2.4 GHz

Any system with above or higher configuration is compatible for this project.

3.1.4 SOFTWARE REQUIREMENTS

- Operating system : Windows 7/8/10/11
- Programming language : Python
- IDE : Jupyter Notebook /Google Colab Notebook
- Tools : Anaconda

3.2 IMPLEMENTATION

IMPORTING LIBRARIES

```
import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

import nltk

import re #It is for regular expression

from wordcloud import WordCloud


# Tokenizer to tokenize the text data

from tensorflow.keras.preprocessing.text import Tokenizer


# Pad sequences to pad those data set which are not long enough

# It is text data , some data could be 100-200 words .

# Suppose we are taking 700 words as constant width then we need to pad to make it 700 words

#  Because the deeplearning model takes constant lenght input

from tensorflow.keras.preprocessing.sequence import pad_sequences
```

```
# Importing our sequential model in which we will bw feeding the layers of our models
from tensorflow.keras.models import Sequential

#These are the keras layers dense , embedding, lstm , convulation1D,MaxPool1D
from tensorflow.keras.layers import Dense, Embedding,LSTM,Conv1D,MaxPool1D

# To split train and test data set
from sklearn.model_selection import train_test_split

# Used to test model performance
from sklearn.metrics import classification_report,accuracy_score
```

IMPORTING FAKE DATA SET

```
fake=pd.read_csv('https://raw.githubusercontent.com/UtkarshMidha/Fake-News-kaggle-dataset/master/Fake.csv')

fake.head()
```

	title	text	subject	date
0	Donald Trump Sends Out Embarrassing New Year'...	Donald Trump just couldn t wish all Americans ...	News	December 31, 2017
1	Drunk Bragging Trump Staffer Started Russian ...	House Intelligence Committee Chairman Devin Nu...	News	December 31, 2017
2	Sheriff David Clarke Becomes An Internet Joke...	On Friday, it was revealed that former Milwauk...	News	December 30, 2017
3	Trump Is So Obsessed He Even Has Obama's Name...	On Christmas day, Donald Trump announced that ...	News	December 29, 2017
4	Pope Francis Just Called Out Donald Trump Dur...	Pope Francis used his annual Christmas Day mes...	News	December 25, 2017

Table 3.2.1 Fake news dataset head

```
for i in fake.columns:

  print(fake[i].value_counts(),end="\n----------\n")

print(fake['subject'].value_counts())

# sns.set_theme(style="whitegrid")

# sns.axes_style("darkgrid")

plt.figure(figsize=(15,7))

sns.dark_palette("xkcd:golden", 8)

sns.countplot(x='subject',data=fake, palette="Set2")
```

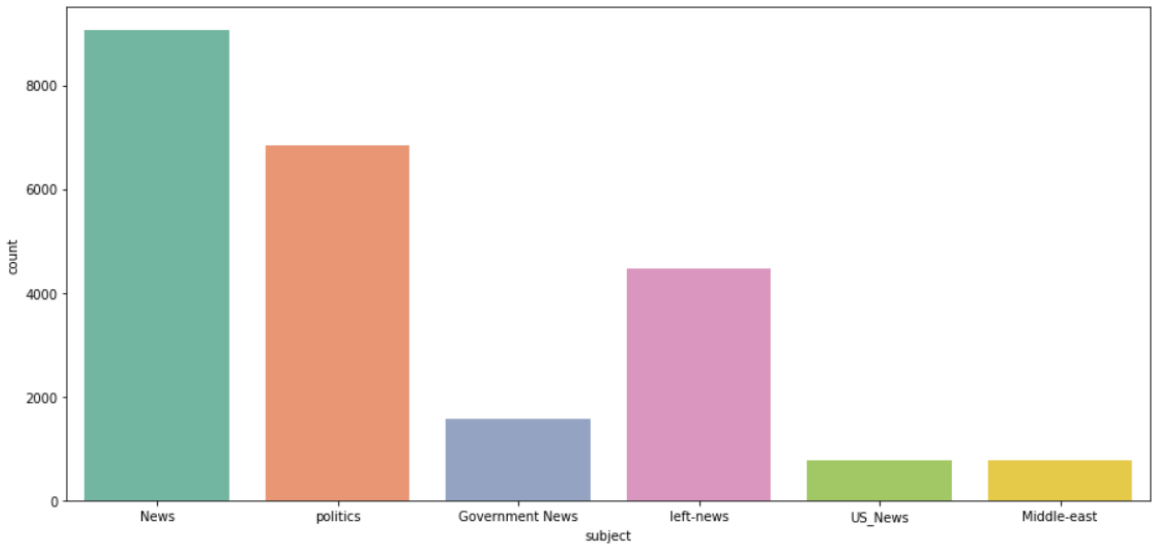

Fig 3.2.2 Fake news subject count

WORDCLOUD OF FAKE NEWS

```
# For wordcloud you need to mix all fake text data together

fake['text'].tolist()[:3]

# We need single text data i.e. join complete text data
```

```
text=' '.join(fake['text'].tolist())

wordcloud=WordCloud(background_color='black',width=1920,
height=1080,colormap='Blues').generate(text)

fig=plt.figure(figsize=(10,10))

plt.imshow(wordcloud)

plt.axis('off')

plt.tight_layout(pad=1)

plt.title("Fake News Word Cloud", fontsize=20)

plt.show()
```

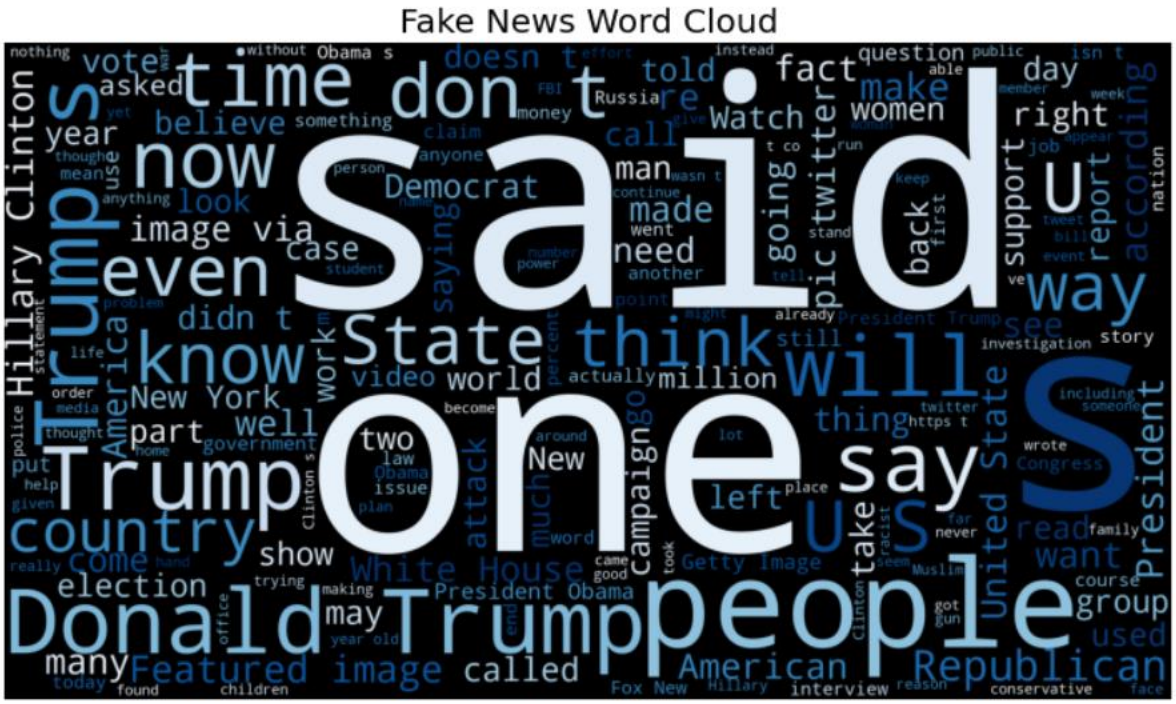

Fig 3.2.3 Fake news word cloud

IMPORTING REAL DATA SET

```
real=pd.read_csv('https://raw.githubusercontent.com/UtkarshMidha/Fake-News-kaggle-dataset/master/True.csv')

real.head()
```

	title	text	subject	date
0	As U.S. budget fight looms, Republicans flip t...	WASHINGTON (Reuters) - The head of a conservat...	politicsNews	December 31, 2017
1	U.S. military to accept transgender recruits o...	WASHINGTON (Reuters) - Transgender people will...	politicsNews	December 29, 2017
2	Senior U.S. Republican senator: 'Let Mr. Muell...	WASHINGTON (Reuters) - The special counsel inv...	politicsNews	December 31, 2017
3	FBI Russia probe helped by Australian diplomat...	WASHINGTON (Reuters) - Trump campaign adviser ...	politicsNews	December 30, 2017
4	Trump wants Postal Service to charge 'much mor...	SEATTLE/WASHINGTON (Reuters) - President Donal...	politicsNews	December 29, 2017

Fig 3.2.4 Real news dataset head

```
real['subject'].value_counts()
```

```
politicsNews    11272
worldnews       10145
Name: subject, dtype: int64
```

Fig 3.2.5 Subject value's count

WORDCLOUD OF REAL NEWS

```
# We need single text data i.e. join complete text data

text=' '.join(real['text'].tolist())

wordcloud=WordCloud(background_color='black',width=1920,
height=1080,colormap='Pastel1').generate(text)

fig=plt.figure(figsize=(10,10))

plt.imshow(wordcloud)

plt.axis('off')

plt.tight_layout(pad=1)
```

```
plt.title("Real News Word Cloud", fontsize=20)

plt.show()
```

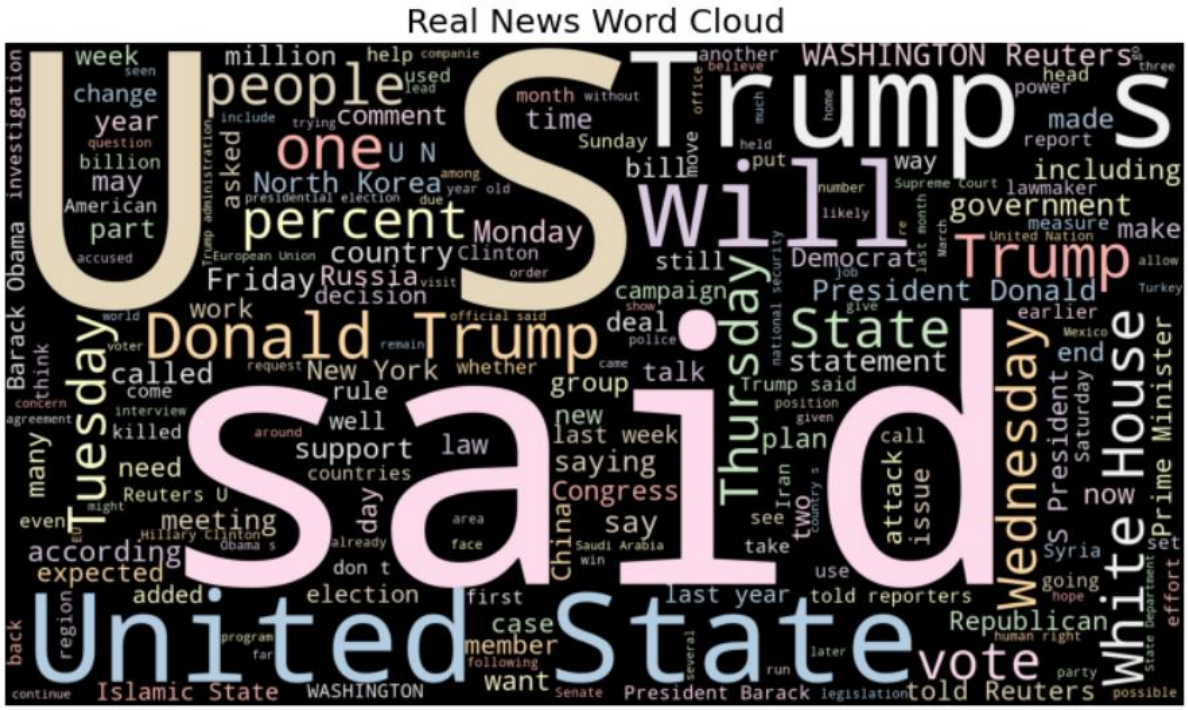

Fig 3.2.6 Real News Word Cloud

Cleaning the Data

```
real.sample(5)
```

	title	text	subject	date
19859	After Irma ravages Havana, city highlights hou...	HAVANA (Reuters) - After Hurricane Irma wrough...	worldnews	September 16, 2017
19080	Kurdistan supervisors begin counting votes in ...	ERBIL, Iraq (Reuters) - Voting stations set up...	worldnews	September 25, 2017
17838	Opposition magistrates holed up in Chile resid...	CARACAS (Reuters) - Five magistrates named by ...	worldnews	October 10, 2017
12458	New Polish PM: must defend national interest i...	WARSAW (Reuters) - Poland should defend its na...	worldnews	December 12, 2017
1585	U.S. Senate opposition to Obamacare repeal bil...	WASHINGTON (Reuters) - A proposal by U.S. Repu...	politicsNews	September 24, 2017

Fig 3.2.7 Random selection of data in Real News dataset

```
unknown_publishers = []

for index,row in enumerate(real.text.values):

  try:
```

```
    # To check if  hyphen (we are breaking text on hyphen)
    record=row.split('-',maxsplit=1)
    record[1] #We try to run it , if empty then it will move to exception
    # Let's know if short news are from twitter
    assert(len(record[0])<120)
  except:
    # Deal with the above 3 cases
    unknown_publishers.append(index)
print(len(unknown_publishers))
```

```
real.iloc[unknown_publishers].text
```

```
7        The following statements were posted to the ve...
8        The following statements were posted to the ve...
12       The following statements were posted to the ve...
13       The following statements were posted to the ve...
14        (In Dec. 25 story, in second paragraph, corre...
                              ...
20135     (Story corrects to million from billion in pa...
20500     (This Sept 8 story corrects headline, clarifi...
20667     (Story refiles to add dropped word  not , in ...
21246     (Story corrects third paragraph to show Mosul...
21339     (Story corrects to fix spelling in paragraph ...
Name: text, Length: 222, dtype: object
```

Fig 3.2.8 Unknown publisher list

```
# Drop 8970 row as it has no data
real.iloc[8970]
```

```
title      Graphic: Supreme Court roundup
text
subject                      politicsNews
date                        June 16, 2016
Name: 8970, dtype: object
```

Fig 3.2.9 Row with empty text field

```
real.drop(8970,axis=0) #axis=0 means drop from row

# Create a new column to add publisher info and handle cases where there is no publisher

publisher=[]

tmp_text=[]


for idx,row in enumerate(real.text.values):

  if idx in unknown_publishers:

    tmp_text.append(row)

    publisher.append('Unknown')

  else:

    record=row.split('-',maxsplit=1)

    publisher.append(record[0].strip()) # First index has publisher

    tmp_text.append(record[1].strip()) # strip is used to remove extra spaces

real['publisher']=publisher

real['text']=tmp_text


real.head()
```

	title	text	subject	date	publisher
0	As U.S. budget fight looms, Republicans flip t...	The head of a conservative Republican faction ...	politicsNews	December 31, 2017	WASHINGTON (Reuters)
1	U.S. military to accept transgender recruits o...	Transgender people will be allowed for the fir...	politicsNews	December 29, 2017	WASHINGTON (Reuters)
2	Senior U.S. Republican senator: 'Let Mr. Muell...	The special counsel investigation of links bet...	politicsNews	December 31, 2017	WASHINGTON (Reuters)
3	FBI Russia probe helped by Australian diplomat...	Trump campaign adviser George Papadopoulos tol...	politicsNews	December 30, 2017	WASHINGTON (Reuters)
4	Trump wants Postal Service to charge 'much mor...	President Donald Trump called on the U.S. Post...	politicsNews	December 29, 2017	SEATTLE/WASHINGTON (Reuters)

Fig 3.2.10 Real Dataset Table with publisher field

```
# Check if fake news also has empty text data

empty_fake_index=[index for index, text in enumerate( fake.text.tolist()) if str(text).strip()==""]


# The iloc() function in python is defined in the Pandas module that helps us to select a specific row or column from the data set

fake.iloc[empty_fake_index]


# We need to remove them

# As text is in the title so we merge text and title

real['text']=real['title']+ " " + real['text']

fake['text']=fake['title']+ " " + fake['text']

real['text']=real['text'].apply(lambda x:str(x).lower())

fake['text']=fake['text'].apply(lambda x:str(x).lower())
```

Preprocessing of Data

```
# There is no label for supervised learning in the real text data and fake text data

# So we add class
```

real['class']=1

fake['class']=0

real.columns

```
Index(['title', 'text', 'subject', 'date', 'publisher', 'class'], dtype='object')
```

Fig 3.2.11 Real dataset having class as a field

We need only text and class

real=real[['text','class']]

fake=fake[['text','class']]

Append these two together

data=real.append(fake,ignore_index=True)

data.sample(5)

	text	class
13110	lebanon's hariri rescinds resignation, drawing...	1
44047	trump faces off with cnn's jake tapper over ev...	0
11851	u.s. seeks ship ban over north korea violation...	1
41101	breaking: secret recordings about clinton foun...	0
37002	parents jailed and kids taken away for 90 minu...	0

Fig 3.2.12 Random Sample data from merges dataset

import preprocess_kgptalkie as ps

data['text']=data['text'].apply(lambda x:ps.remove_special_chars(x))

#The above line removes special character # eg :

ps.remove_special_chars('this .,is !@#$%%^&awesome)')

Vectorization

```
# To convert text to numerical data we use vectorization

# i.e. word 2 vector # https://www.tensorflow.org/tutorials/text/word2vec

import genism

y=data['class'].values

data['text'].tolist()

X = [d.split() for d in data['text'].tolist()]

# Taking dimension to be 100

DIM=100

w2v_model=gensim.models.Word2Vec(sentences=X,size=DIM,window=10 , min_count=1)

w2v_model.wv['india']
```

```
array([ 1.2766991 ,  2.2952096 ,  2.683091  , -0.9097098 , -0.22816421,
        0.46104145, -2.5216467 , -1.8398476 , -0.80013704,  2.6081471 ,
       -0.10403663,  2.4903958 , -1.4176043 ,  2.371989  , -0.05224562,
        1.7308472 ,  0.7346816 , -1.4048808 , -1.0184395 ,  1.3802904 ,
       -0.60796785, -1.0310636 ,  2.4042883 ,  0.7317703 , -0.9416885 ,
        1.6565094 , -0.9191489 , -0.34696656,  2.052084  ,  0.68073195,
       -1.5750781 ,  0.5120773 , -2.9663014 ,  0.08305793,  0.74701047,
        0.5356894 , -1.8746672 , -1.8937067 ,  3.2106106 ,  1.2956073 ,
        0.93291193, -1.3346196 , -0.31350818, -1.5797937 , -0.16749053,
        0.69901335, -0.5693547 ,  1.1610979 ,  1.0412673 ,  0.8051684 ,
        3.4016743 , -2.8989294 ,  3.4367797 , -2.4626105 , -1.7282575 ,
        1.7476593 , -1.594854  ,  0.70840704,  0.0260687 ,  0.5316434 ,
        0.81904507, -1.9694868 , -0.9200749 , -0.91580373, -1.1600637 ,
       -1.4788564 , -1.0928841 , -0.63876337,  0.20368016, -0.33307198,
       -1.4555008 ,  1.2209266 , -3.695195  ,  1.9468713 , -0.6230867 ,
       -2.632346  , -0.48098126, -1.2124912 , -0.24792801,  0.17974593,
        1.3263    , -0.0144372 , -2.3789935 ,  2.2481909 ,  1.156116  ,
        1.5509818 , -3.9409263 ,  2.6356177 , -0.9151728 ,  0.2235481 ,
       -1.4708967 ,  2.3635867 ,  0.4547821 , -0.30451322,  2.6915374 ,
       -0.59633017,  0.53405136, -0.3885622 ,  1.0585091 ,  1.1865733 ],
      dtype=float32)
```

Fig 3.2.13 word to vector of “India”

```
#For most similar to current word
w2v_model.wv.most_similar('usa')


# We can feed these vectors in machine learning model
# Then ML Model receates these vectors
# then there might be better accuracy


#We are going to create tokenizer
tokenizer=Tokenizer()
tokenizer.fit_on_texts(X)


# After tokenization the vector is converted into set of sequence
X= tokenizer.texts_to_sequences(X)
tokenizer.word_index
```

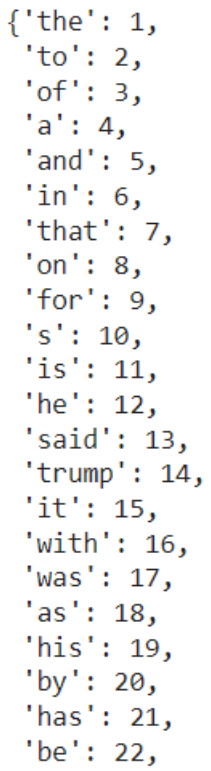

Fig 3.2.14 Tokenizer Word index

Now we will analyze the data in X

plt.hist([len(x) for x in X],bins =700, ec="skyblue") #A histogram displays numerical data by grouping data into "bins" of equal width.

plt.show()

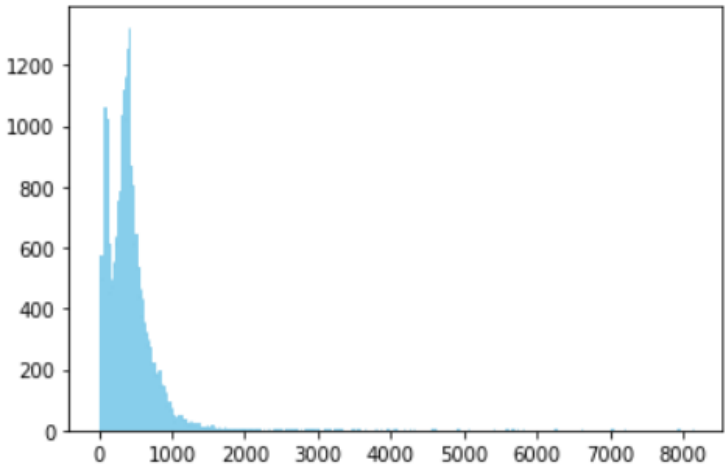

Fig 3.2.15 Distribution of length of data samples

```
# Lets see how many numbers are there

# So we create a numpy array

nos=np.array([len(x) for x in X])

len(nos[nos>1000])


maxlen=1000

X = pad_sequences(X,maxlen=maxlen)

# We have got every sequence greater than 1000


# index starts from index 1 so we do +1 so that we can have room for unknown word

vocab_size = len(tokenizer.word_index)+1

vocab=tokenizer.word_index
```

```
# We take the initial vector as an input to the machine learning model
def get_weight_matrix(model):
  weight_matrix=np.zeros((vocab_size,DIM))
  for word,i in vocab.items():
    weight_matrix[i] = model.wv[word]
  return weight_matrix
embedding_vectors = get_weight_matrix(w2v_model)
embedding_vectors.shape
```

Create ML MODEL

```
model = Sequential()
# trainable= false means no retraining of vectors
model.add(Embedding(vocab_size, output_dim=DIM , weights = [embedding_vectors] , input_length=maxlen , trainable=False))
model.add(LSTM(units=128))
# Activation function used is sigmoid as there are two classes 0 and 1 only
model.add(Dense(1, activation='sigmoid'))
model.compile(optimizer='adam' , loss='binary_crossentropy' , metrics=['acc'])
# Summary of the model
model.summary()
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding (Embedding)       (None, 1000, 100)         23191200

 lstm (LSTM)                 (None, 128)               117248

 dense (Dense)               (None, 1)                 129

=================================================================
Total params: 23,308,577
Trainable params: 117,377
Non-trainable params: 23,191,200
_________________________________________________________________
```

Fig 3.2.16 Sequential Model Summary

Now Split train and test data

X_train,X_test,y_train,y_test = train_test_split(X,y)

Keras can separate a portion of your training data into a validation dataset and evaluate the performance of your model on that validation dataset in each epoch.

model.fit(X_train,y_train , validation_split=0.3, epochs=6)

```
Epoch 1/6
737/737 [==============================] - 1215s 2s/step - loss: 0.1606 - acc: 0.9387 - val_loss: 0.1272 - val_acc: 0.9550
Epoch 2/6
737/737 [==============================] - 1204s 2s/step - loss: 0.0738 - acc: 0.9758 - val_loss: 0.0415 - val_acc: 0.9862
Epoch 3/6
737/737 [==============================] - 1194s 2s/step - loss: 0.0325 - acc: 0.9896 - val_loss: 0.0273 - val_acc: 0.9913
Epoch 4/6
737/737 [==============================] - 1196s 2s/step - loss: 0.0203 - acc: 0.9938 - val_loss: 0.0212 - val_acc: 0.9932
Epoch 5/6
737/737 [==============================] - 1183s 2s/step - loss: 0.0104 - acc: 0.9964 - val_loss: 0.0191 - val_acc: 0.9945
Epoch 6/6
737/737 [==============================] - 1167s 2s/step - loss: 0.0107 - acc: 0.9966 - val_loss: 0.0209 - val_acc: 0.9946
<keras.callbacks.History at 0x7efdbb3a2f90>
```

Fig 3.2.17 Training of Epochs for sequential model

Checking accuracy on Test Data set

y_pred = (model.predict(X_test)>=0.5).astype(int)

```
A=accuracy_score(y_test,y_pred)

print("Accuracy : "+str(A*100)+"%")

accuracy_score(y_test,y_pred)

print(classification_report(y_test,y_pred))
```

	precision	recall	f1-score	support
0	0.99	1.00	0.99	5909
1	1.00	0.99	0.99	5316
accuracy			0.99	11225
macro avg	0.99	0.99	0.99	11225
weighted avg	0.99	0.99	0.99	11225

Fig 3.2.18 Sequential model classification report

Testing on text data after training

```
x= ['this is a news']

len(tokenizer.texts_to_matrix(x)[0]) # Length is huge

x= ['this is a news']

x=tokenizer.texts_to_sequences(x)

x=pad_sequences(x,maxlen=maxlen)

(model.predict(x)>=0.5).astype(int)
```

```
1/1 [==============================] - 0s 50ms/step
array([[0]])
```

Fig 3.2.19 Correct prediction of Fake news

#Now x is similar to X_test

model.predict(x)

x=["'And when the former president announced Tuesday that he would make a comeback bid for the White House, response was similarly mixed, if not chilly. Only a handful of Republican members of Congress backed him, while others either said they hoped there would be more options in the primary or just scurried away from reporters' questions on Capitol Hill. Attention turned to polls showing that DeSantis would best him in hypothetical primary competitions, and even his own daughter and former adviser Ivanka Trump released a non-committal statement separating herself from his campaign without even an explicit endorsement.."']

x=tokenizer.texts_to_sequences(x)

x=pad_sequences(x,maxlen=maxlen)

(model.predict(x)>=0.5).astype(int)

```
1/1 [==============================] - 0s 40ms/step
array([[1]])
```

Fig 3.2.20 Correct prediction of Real News

What is Natural Language Processing (NLP)?

Natural Language Processing (NLP) is basically how you can teach machines to understand human languages and extract meaning from text.

import numpy as np

import pandas as pd

```
import seaborn as sns

import matplotlib.pyplot as plt

import re

import nltk as nlp

from nltk.corpus import stopwords

from nltk.stem.porter import PorterStemmer

from sklearn.linear_model import LogisticRegression

from sklearn.metrics import accuracy_score

from sklearn.model_selection import train_test_split


# Reading datasets

fake_df = pd.read_csv("https://raw.githubusercontent.com/UtkarshMidha/dataset/master/Fake.csv")

true_df = pd.read_csv("https://raw.githubusercontent.com/UtkarshMidha/dataset/master/True.csv")


# Rows and columns of fake news dataset

print(f"Fake news dataset has: {fake_df.shape[0]} rows")

print(f"Fake news dataset has: {fake_df.shape[1]} columns")
```

Dealing with missing values if any

```
fake_df.isnull().sum()
```

```
true_df.isnull().sum()
```

Working on date column

```
# Adding 'Fake' column to our datasets then join them together

fake_df['Fake'] = 1

true_df['Fake'] = 0

df = pd.concat([fake_df, true_df])


# Extracting the year and the month

df['date'] = pd.to_datetime(df['date'], errors='coerce')

df['Year'] = df['date'].dt.year

df['Month'] = df['date'].dt.month
```

EDA - Exploratory Data Analysis

```
plt.style.use('ggplot')

plt.figure(figsize=(10,7))

sns.countplot(data=df, x='Fake')
```

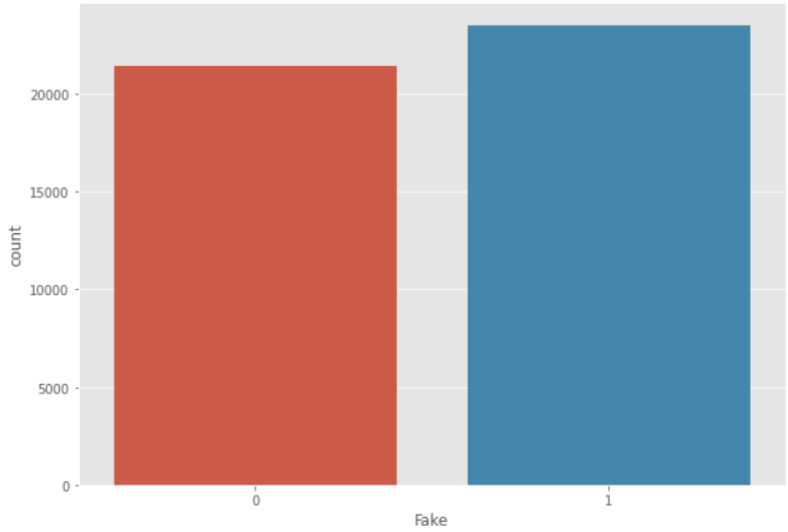

Fig 3.2.21 Real vs Fake plot

```
# Correlation betwwen year and news

plt.style.use('seaborn-pastel')

plt.figure(figsize=(10, 7))

sns.countplot(data=df, x='Year', hue='Fake')
```

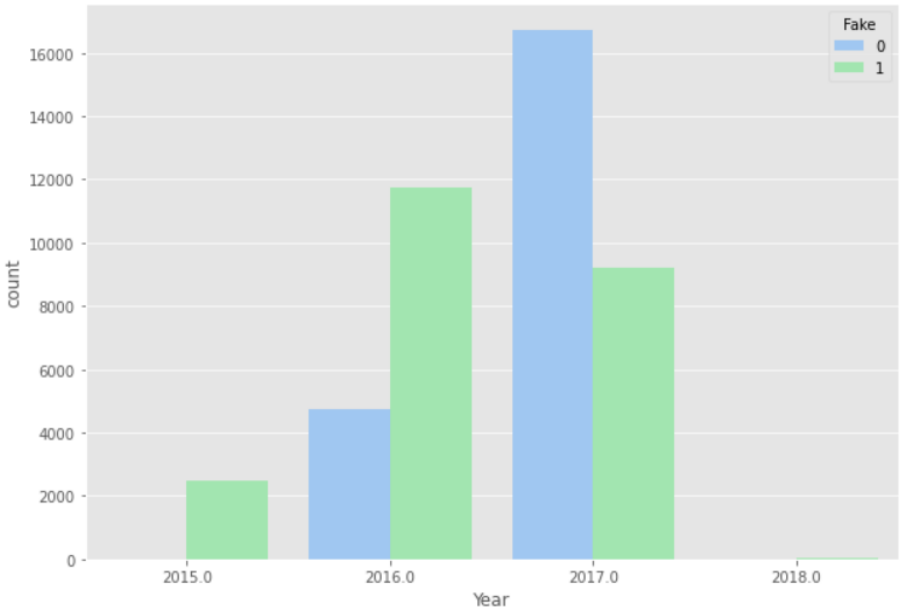

Fig 3.2.22 Correlation between year and news

Correlation between months and news

```
plt.style.use('bmh')

plt.figure(figsize=(12, 7))

sns.countplot(data=df, x='Month', hue='Fake')
```

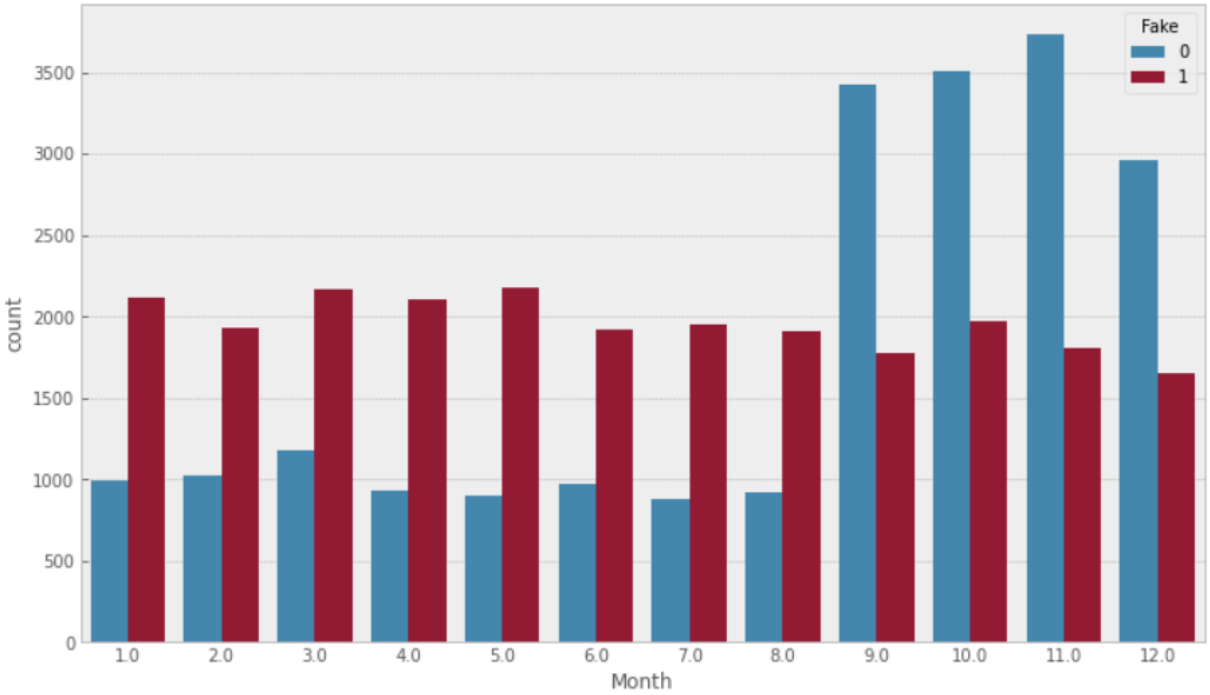

Fig 3.2.23 Correlation between months and news

Subjects count

```
df.subject.value_counts()

plt.style.use('seaborn-paper')

plt.figure(figsize=(12, 7))

sns.countplot(data=df, x='Year', hue='subject')
```

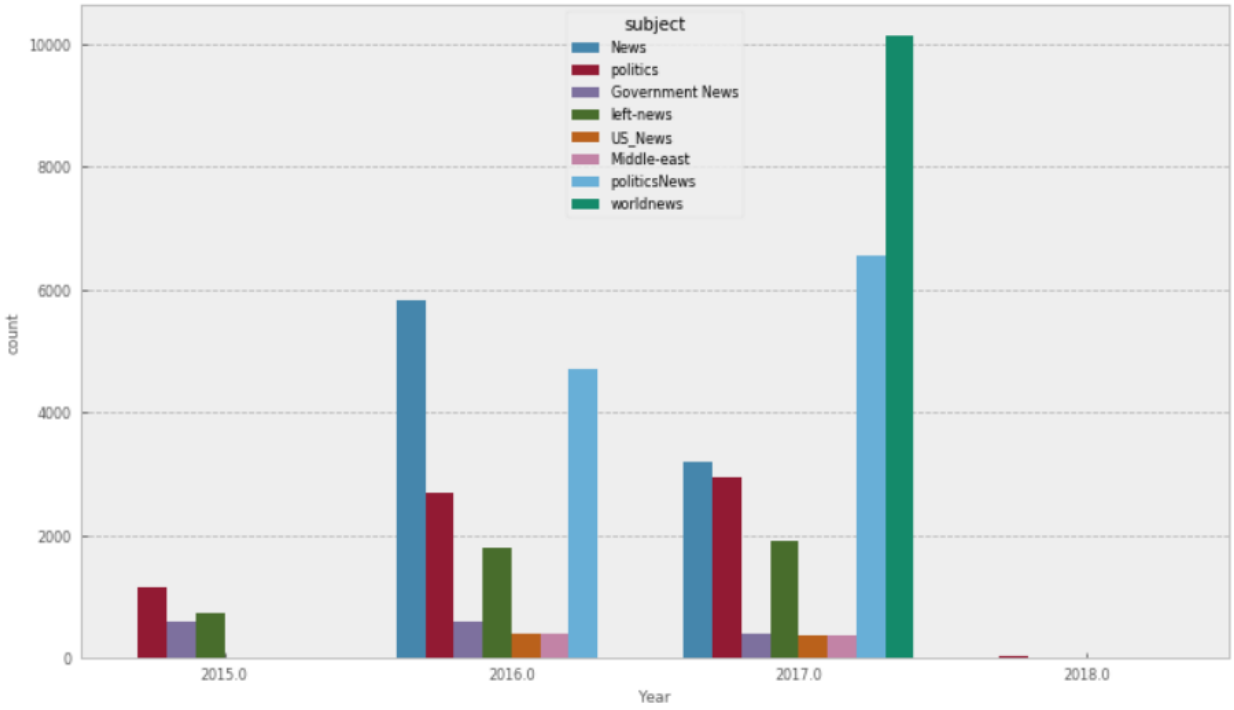

Fig 3.2.24 Correlation between subject and year

```
nlp.download('stopwords')

print(stopwords.words('english'))

# Joining the title and the content columns

df['full_text'] = df['title'] + ' ' + df['subject']

df.head()
```

	title	text	subject	date	Fake	Year	Month	full_text
0	Donald Trump Sends Out Embarrassing New Year'...	Donald Trump just couldn t wish all Americans ...	News	2017-12-31	1	2017.0	12.0	Donald Trump Sends Out Embarrassing New Year'...
1	Drunk Bragging Trump Staffer Started Russian ...	House Intelligence Committee Chairman Devin Nu...	News	2017-12-31	1	2017.0	12.0	Drunk Bragging Trump Staffer Started Russian ...
2	Sheriff David Clarke Becomes An Internet Joke...	On Friday, it was revealed that former Milwauk...	News	2017-12-30	1	2017.0	12.0	Sheriff David Clarke Becomes An Internet Joke...
3	Trump Is So Obsessed He Even Has Obama's Name...	On Christmas day, Donald Trump announced that ...	News	2017-12-29	1	2017.0	12.0	Trump Is So Obsessed He Even Has Obama's Name...
4	Pope Francis Just Called Out Donald Trump Dur...	Pope Francis used his annual Christmas Day mes...	News	2017-12-25	1	2017.0	12.0	Pope Francis Just Called Out Donald Trump Dur...

Fig 3.2.25 table with full text field

```
# Target variable and features

y = df['Fake']
```

```
X = df.drop('Fake', axis=1)


# Stemming process
def stemming_process(y):
    first_step = re.sub(r"^[A-Za-z]", ' ', y).lower()
    second_step = first_step.split()
    porter_stemmer = PorterStemmer()
    result = []
    for w in second_step:
        if w not in stopwords.words('english'):
            result.append(porter_stemmer.stem(w))
    return ' '.join(result)


df['full_text'] = df['full_text'].apply(stemming_process)


# Converting X and y to numpy arrays
X = df['full_text'].to_numpy()
y = df['Fake'].to_numpy()


# Converting the text into numerical values
from sklearn.feature_extraction.text import TfidfVectorizer
```

```
vectorizer = TfidfVectorizer()

vectorizer.fit(X)

X = vectorizer.transform(X)


# Building the model

from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25, random_state=42)


# Fitting the model - Logistic Regression

lg = LogisticRegression()

lg.fit(X_train, y_train)


# Predictiion

prediction = lg.predict(X_test)


# Score

accuracy = accuracy_score(prediction, y_test)

print(f"Model precision: {accuracy}")
```

```
Model precision: 0.9998218262806237
```

Fig 3.2.26 Model2 precision

CHAPTER 4

RESULTS AND DISCUSSION

The Logistic Regression model produces 99.98% accuracy. When we input the news text on the interface, it correctly identifies the news most of the time. We tested this by using news from The Onion. The Onion is a satire ‘news’ portal that posts fake funny news. When we pasted some of the news from the site, those were correctly identified as fake. But when we wanted to test the news from BBC or New York Times, those were correctly identified as real. The accuracy of the LSTM model was 99.28%, so we conclude that Logistic Regression Model is more accurate as compared to LSTM model.

CHAPTER 5

CONCLUSION AND FUTURE SCOPE

Our project can ring the initial alert for fake news. The model produces worse results if the article is written cleverly, without any sensationalization. This is a very complex problem but we tried to address it as much as we could. We believe the interface provides an easier way for the average person to check the authenticity of a news. Projects like this one with more advanced features should be integrated on social media to prevent the spread of fake news.

There are many future improvement aspects of this project. Introducing a cross checking feature on the machine learning model so it compares the news inputs with the reputable news sources is one way to go. It has to be online and done in real time, which will be very challenging. Improving the model accuracy using bigger and better datasets, integrating different machine learning algorithms is also something we hope to do in the future.

REFERENCES

1. Samarth Mengji, Saransh Ambarte, Sai Viswa Teja Arumilli, Shankar Mhamane, Rashmi Rane “Fake News Detection using RNN-LSTM”, International Journal for Research in Applied Science & Engineering Technology, (2021).

2. Ray Oshikawa, Jing Qian, William Yang Wang, “A Survey on Natural Language Processing for Fake News Detection”, Proceedings of the 12th Language Resources and Evaluation Conference (LREC 2020).

3. Kai Shu, Suhang Wang, Huan Liu, “Beyond News Contents: The Role of Social Context for Fake News Detection”, In Proceedings of 12th ACM International Conference on Web Search and Data Mining (WSDM 2019).

4. Kai Shu, Suhang Wang, Huan Liu, “Understanding User Profiles on Social Media for Fake News Detection”, FakeMM’18 Workshop, (2018).

5. Juan Cao, Peng Qi, Qiang Sheng, Tianyun Yang, Junbo Guo, Jintao Li, “Exploring the Role of Visual Content in Fake News Detection, Disinformation, Misinformation, and Fake News in Social Media. 2020

6. Dhruv Khattar, Jaipal Singh Goud, Manish Gupta, Vasudeva Varma, “MVAE: Multimodal Variational Autoencoder for Fake News Detection”, In Proceedings of the 2019 World Wide Web Conference (WWW ’19).

7. Shivam Parikh, Media-Rich Fake News Detection: A Survey, MIPR 2018At: Miami, FL, (April 2018).

8. Manoj Balwant, "Bidirectional LSTM Based on POS tags and CNN Architecture for Fake News Detection", 10th ICCCNT 2019At: IIT kanpur, (December 2019).

9. Hamid Karimi, Jiliang Tang, "Learning Hierarchical Discourse-level Structure for Fake News Detection", Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), (June 2019).

10. Yunfei Long, Qin Lu, Rong Xiang, Minglei Li, Chu-Ren Huang, "Fake News Detection Through Multi-Perspective Speaker Profiles", Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers), (November 2017).

11. Anastasia Giachanou, Guobiao Zhang, Paolo Rosso, "Multimodal Multi-image Fake News Detection", IEEE 7th International Conference on Data Science and Advanced Analytics (DSAA) (2020).

12. Marwan Albahar, "A hybrid model for fake news detection: Leveraging news content and user comments in fake news", IET Information SecurityVolume 15, Issue 2Mar 2021 Pages147-204.

13. Xishuang Dong; Uboho Victor; Lijun Qian, "Two-Path Deep Semisupervised Learning for Timely Fake News Detection", IEEE Transactions on Computational Social Systems Volume: 7, Issue: 6, (December 2020).

14. Qing Liao; Heyan Chai; Hao Han; Xiang Zhang; Xuan Wang; Wen Xia; Ye Ding, "An Integrated Multi-Task Model for Fake News Detection", IEEE Transactions on Knowledge and Data Engineering Volume: 34, Issue: 11, (November 2022).

APPENDIX1

Fake News Detection using LSTM and Logistic Regression

Utkarsh Midha[1[0009-0003-2161-3439]], Ujjwal Kumar[1[0009-0000-2826-7282]], Tripti Saloni[1[0009-0006-5600-9428]] and Hriday Kumar Gupta[1[0000-0001-6115-225X]]

[1] Department of Computer Science and Engineering, KIET Group of Institutions, Ghaziabad, India

Abstract. In this informational age, whether we examine a bit of writing or watch the news on television, we look for a sinceresupply. Internet plays a major role for providing any information or news, people rely on this very much, but the internet and social media are both full of false information. Misinformation or news that has been edited and posted on social media with the intention of harming a person, business, or enterprise is referred to as fake news. This false information can affect a person life adversely. Disasters might result from the spread of false information in urgent circumstances. The need of fake news identification is a must. The spread of fake news necessitates the developmentof computer algorithms to identify it. We have therefore included Machine Learning algorithms and techniques like NLTK, LSTM in order to prevent the harm that can be caused by spread of false information through technology.

Keywords: LSTM; Logistic Regression; Stemming; Lemmatization; Tokenization; NLP.

1 Introduction

Fake news is information that has been deliberately misinformed or falsely [22] claimed to be real. These tales aretypically written to sway people's evaluations, forward a political agenda, or create confusion, and they can regularlyconvey in cash for internet courses. This issue was chosen since it is turning into a significant social problem. It is ensuing in a poisonous on-line environment and riots and lynchings on the streets. Examples consist of political fakenews, stories about sensitive subjects like faith, religion, rumour that salt and garlic may treat corona, and every othercomparable messages we encounter on social media. We can all see the harm that can result from fake information, which is why there is a determined need for a tool which can affirm sure news, whether it is faux or real, and offer individuals with a sense of authenticity primarily based on which they can decide whether to act. With so much noisefrom fake news and fake statistics, if people lose faith in records, they will no longer be able to access even the mostimportant information, which can even occasionally be life-changing or even fatal. Hence identification of informationbecomes a necessary step or process to prevent any harm that can be caused by faux news.

2 Related Work

Due to social media, spreading false records has end up pretty simple in the contemporary era. To counteract this, we are going to expand a model that uses ML and NLP ideas and algorithms to identify whether or not a particular piece of information is genuine or not in order to determine whether the news is real or not.

2.1 Stemming

Along with a pre-processing stage in Text Mining applications [2], stemming is a common precondition for Natural Language Processing (NLP) [15] activities. In truth, it plays a crucial function in almost all information retrieval systems. of a section or subsection is not indented. The simple aim of [13] the stage stemming is generally to lessen word's various grammar structures, which are as its verb, adverb, adjective, and noun forms, to its most basic form. We are able to mention stemming goals[24] to lessen a phrase's morphological patterns and once in a while derivationally related forms to a fundamental form that is shared with the aid of all.

2.2 Lemmatization

Locating a word's normalized shape is referred to as lemmatization [8]. Lemmatization is a normalization technique [5], to find a transformation to use to a phrase on the way to achieve its normalized form has similarities to this. One of the approach specializes inword ends, particularly which word suffix should be saved or introduced to supply the normalized form. This examination contrasts the outcomes of two-word lemmatization algorithms, one based on if-then regulations and the other on ripple down regulations induction strategies. It discusses why the Ripple Down Rules (RDR) [25] approach is ideal for the motive and then addresses the problem of lemmatization of terms from Slovene free text. While studying from a corpus of lemmatized Slovene words, the RDR approach yields rules that are simpler to comprehend and feature better classification accuracy than the ones obtained through rule learning in earlier research.

2.3 Tokenization

The process of adding linguistic annotation [23] to certainly occurring text can be seen of as a chain of adjustmentsmade to the authentic textual context, each of which removes the surface distinctions. It means that the input text is split into tokens that are feasible for further analysis.[3] One of the preliminary stages of this transition in the course of nlp is tokenization. It refers to division of text which is taken as input, it is to a computer only a single characters' string into tokens. Then, these tokens are utilized in later stages [16] of natural language processing such morphological analysis, wordclass tagging, and parsing. Tokenization frequently includes identifying both sentence and token boundaries [11] due to the fact these subsequent treatments , are usually designedto work on particular sentences. A number of challenging issues are raised by tokenization when it is in an automated [9] TPS or text processing system demanding situations, handful of them have complete great solutions, no matter being occasionally inspected and quickly discarded. It is not the initial step in the abstraction [19] process. Most differences in typesetting, when are compared to an genuine printed piece of document, the texts are filtered to removepage layout ,font size, graphics, photos and font style.

2.4 Logistic Regression

Logistic regression is a statistical analysis method [4] to predict a binary outcome, such as yes or no, based on prior observations of a data set. Odds ratios are obtained using logistic regression [12] when there are multiple explanatory variables. Apart from thereaction (response) parameter [17] given that the process is a binomial, is noticeably similar to the multiple linear regression. The result is the effect of each and every variable on the probability ratio of the observed significant occurrence. The main advantage is that it is possible to prevent confusing effects of variables by examining how all the variables are related.

2.5 Word Cloud

A "word cloud" is a depiction of word frequency [10]. It shows the most common words in a text document. [1] The word size in the accompanying graphic increases with the phrase's frequency [22] of occurrence in the studied text. Using word clouds as a quick method to decide the primary idea of written content is growing. For instance, they have been used to visualize the content of political speeches in business, education, and politics.

2.6 Natural Language Processing

The collection of techniques known as "natural language processing" enables computers to understand human language. Natural language processing has permeated our daily lives over the past ten years in a number of ways: On the internet and in social media, automatic machine translation is widely used; text classification prevents our email inboxes from overflowing with spam; Search engines have developed to a high level of linguistic complexitybeyond string matching and network analysis; dialog systems offer an increasingly popular and efficient means of obtaining and exchanging information [19].

3 Proposed System

3.1 LSTM Model

A recurrent neural network's layers are constructed from long short-term memory (LSTM) units (RNN). A cell,an output and input gate along with a forget gate makes up an unit of long short-term memory [20]. The in charge of "remembering" values over a long period of time such that the relationship between words at the beginning of text can affect how those words appear later in the sentence is the cell. Traditional neural networks seem to have a significant flaw in that they cannot remember or keep track of the whole thing that occurs before they are put into action, which prevents the desired influence of words from the previous phrase from having any impact at the finalwords.

Dataset Overview. The data was obtained through the Kaggle platform. It possesses the subsequent attributes: title: the title of a news article, text: the article's material, subject: the article's subject, and date: the article's publication date. 21417 news articles in all make up the dataset used for model testing and training. A blend of true and fake news is used to createthe dataset.

The following image depicts LSTM cell structure[26].

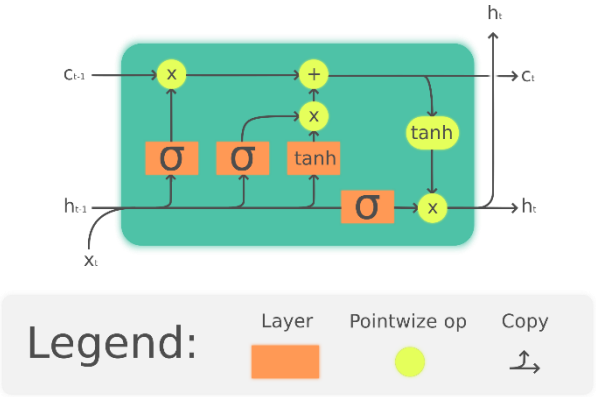

Fig. 1. LSTM Architecture(LSTM Model)

Pre- processing. The data set needs to be processed before it can be transformed into the appropriate format. As the raw data had no labels, so we classified real and fake news, datasets and merged them. Firstly, we eliminated special characters and NULL field data values from our datasets.

Word Embedding. Word to Vector representation: The method cannot accept input in text format, so we ought to convert it into numeric form. To do this, we use word to vector encoding. In word to vector representation the dimension is set to be 100. Each word vector undergoes tokenization. Word tokenization adds text to a list that is then referred to as a vocab list.The list of all the terms used in the narrative serves as the output for this stage.

Model. The model is given the word embedding's output. The machine learning model used in this instance is a sequential version with embedding as the first layer and values for vocabulary size, the number of features, and sentence length. Next comes the LSTM, which has 128 neurons per layer, then the dense layer with sigmoid activation function sincewe only need one output at the end. To prevent overfitting, we added a drop out layer among the Adam optimizer foradaptive estimation and binary cross entropy to calculate loss. The model is then trained and tested.

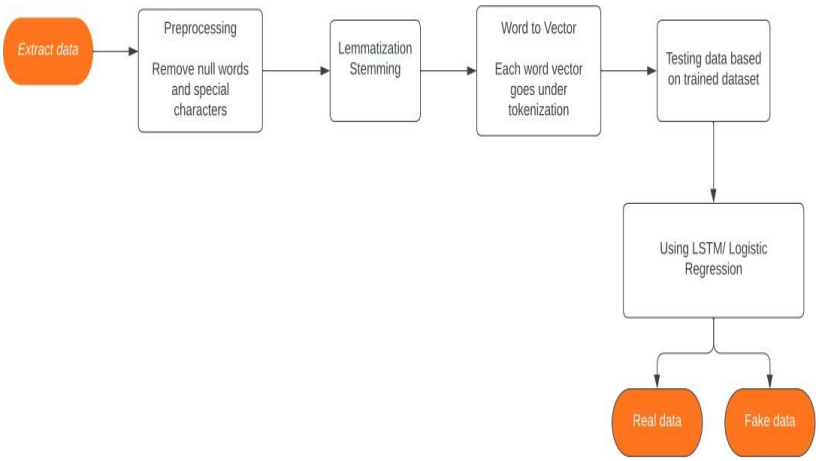

Fig. 2. Proposed System Module

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding (Embedding)       (None, 1000, 100)         23191200

 lstm (LSTM)                 (None, 128)               117248

 dense (Dense)               (None, 1)                 129

=================================================================
Total params: 23,308,577
Trainable params: 117,377
Non-trainable params: 23,191,200
_________________________________________________________________
```

Fig. 3. LSTM Model

Accuracy. The accuracy of the model is 99.4 %.

```
A=accuracy_score(y_test,y_pred)
print("Accuracy : "+str(A*100)+"%")
```

```
Accuracy : 99.42984409799554%
```

```
accuracy_score(y_test,y_pred)
```

```
0.9942984409799555
```

3.2 Logistic Regression (LR) model

In the supervised learning space, one of the most widely used machine learning algorithm is logistic regression. Itis used to interpret a collection of predetermined independent variables [7] and a categorical dependent variable. Toforecast the state of a categorical dependent variable, one make use of logistic regression [14]. The outcomes must therefore be distinct or categorical values. Instead of an exact value between 0 and 1, it promises a probabilistic valuebetween 0 and 1. True or false, 0 or 1, yes or no, and so on are all possible outcomes [6].

The following is the model curve between logistic and linear regression[27].

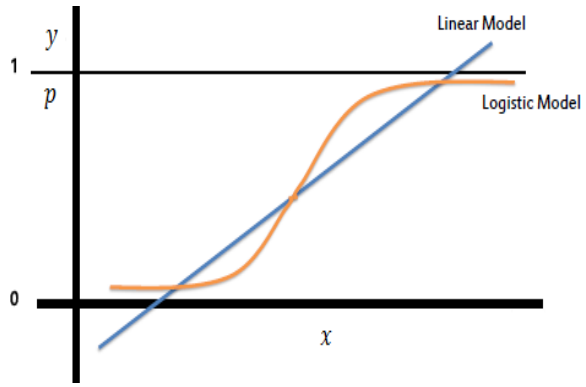

Fig. 4. Model Curve of Logistic VS Linear Regression

Implementation details. The data set needs to be pre-processed to be able to be converted into the precise format. First, we purged the dataset of all NAN and NULL values. Then the process of stemming is completed, followed by vectorization.

Model. The model used is the logistic regression. This model offers greater accuracy, it offers an accuracy of 99.98%.

```
accuracy = accuracy_score(prediction, y_test)
print(f"Model precision: {accuracy}")
```

```
Model precision: 0.9998218262806237
```

4 Results

The accuracy of the Logistic Regression model is 99.98%. Maximum of the time, while we enter news textual content into the interface, it accurately detects the news. We placed this to the test using articles from The Onion. A parody "news" internet site called The Onion publishes made-up, a laugh testimonies. Some of the news that we copied and pasted from the website was appropriately detected as bogus. However, while we tried to verify the news from the BBC or the New York Times, those have been recognized as genuine. Since the LSTM model's accuracy was 99.42%.

5 LSTM vs Logistic Regression Model

LSTM gives the accuracy of 99.42% whereas Logistic Regression has 99.98% accuracy. We may infer that theLogistic Regression Model is more accurate than the LSTM model.

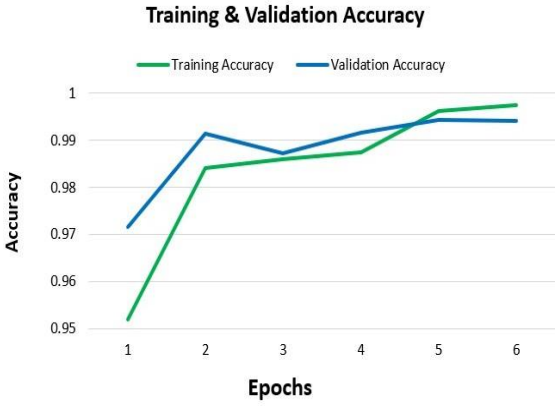

Fig. 5. Model Accuracy Chart

Fig. 6. Model Loss Chart

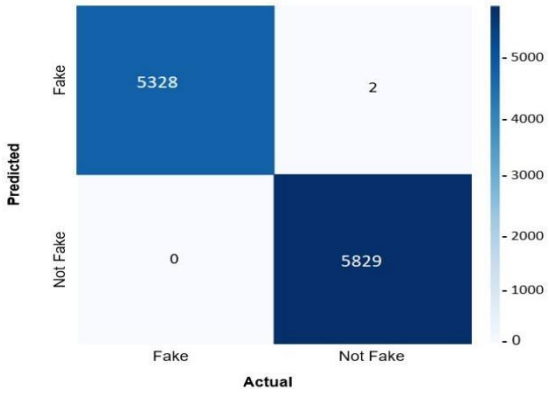

Fig. 7. Logistic Regression Confusion Matrix

6 Limitations

While the data shown here imply a model, some external factors, such as the news's author, location of origin, and time stamp, which may additionally have an impact on the model's output, have been now not taken into considered. There aren't many datasets and research papers available for detecting fake news. The dataset may be tainted, that may lead to detrimental findings. The accuracy of the final results is impacted by the shorter length of the headline or overall report. Training time in faux news grows as module layers increase.

7 Applications

Journalism: Since newspapers and news channels are the primary sources of reliable information, this detection can be used toconfirm the news before it is broadcast.

Social Media: It is simple to distort any news or information in modern's social media environment. Such fabricated information misleads the target audience. Knowing if news is phoney or authentic is essential. This document offers a spread of facts detection and information categorization procedures.

8 Conclusion

With the help of our mission, bogus news can be detected early. If the piece is cleverly written and without any sensationalization, the model produces worse outcomes. We made every effort to address this problem, in spite of how hard it's miles. We think that the consumer-friendly interface makes it simpler for the everyday consumer to confirm the veracity of a information object. To forestall the propagation of false information on social media, initiatives like this one with extra sophisticated features should be carried out. This project has a whole lot of room for future improvement. One approach is to add a go-checking functionality to the machine learning model so it is able to examine the information inputs with the reliable news sources. It desires to be carried out online and in real time, which will be very difficult. We additionally intend to integrate various machine learning techniques in the future, as well as decorate the model accuracy making use of extra and better datasets.

9 Future Scope

This paper has demonstrated the effectiveness of LSTM and NLP techniques for detecting fake news. However, there are several areas for future research that could expand upon this work. Firstly, model accuracy could be improved by exploring ways to fine-tune the models or incorporate additional features. Secondly, the robustness of LSTM and NLP models to adversarial attacks could be evaluated to ensure that they are effective in real-world scenarios. Thirdly, multimodal features could be incorporated to develop more comprehensive fake news detection models. Fourthly, the potential of transfer learning using pre-trained models could be investigated to further improve performance. Lastly, the real-world applications of fake news detection using LSTM and NLP techniques could be explored, such as integration into social media platforms or news websites. Pursuing these future research directions could significantly advance the field of fake news detection and contribute to the fight against misinformation.

10 Acknowledgement

We would like to extend our appreciation to Hriday Kumar Gupta for his invaluable support and guidance throughout the writing process. Additionally, we are grateful to the anonymous reviewers for their insightful feedback, which helped to enhance and refine the presentation of our paper.

References

1. Soesanto, A. M., Chandra, V. C., & Suhartono, D. Sentiments comparison on Twitter about LGBT. Procedia Computer Science, 216, 765-773 (2023).
2. Sahu, S. S., & Pal, S. Building a text retrieval system for the Sanskrit language: Exploring indexing, stemming, searching issues. Computer Speech & Language, 101518 (2023).
3. Carlström, Klara. "Implementation of interpretable methods for paraphrasing and text disambiguation." (2023).
4. TechTarget. (n.d.). Logistic regression. SearchBusinessAnalytics. Retrieved April 25,2023, from https://www.techtarget.com/searchbusinessanalytics/definition/logistic-regression
5. Bafitlhile, K. D. A context-aware lemmatization model for setswana language using machine learning (2022).
6. K. K. Hiran, R. K. Jain, D. K. Lakhwani and D. R. Doshi, Machine Learning: Master Supervised and Unsupervised Learning Algorithms, (2021).
7. L. Yang, J. Li, W. Lu, Y. Chen, K. Zhang and Y. Li, "The influence of font scale on semantic expressionof word cloud," Journal of Visualization, (2020).
8. M. Orešković, An Online Syntactic and Semantic Framework for Lexical Relations Extraction Using Natural Language Deterministic Model, (2019).
9. S. Ananth, D. Radha, D. Prema and K. Nirajan, "Fake News Detection using Convolution Neural Network in Deep Learning," International Journal of Innovative Research in Computer and Communication Engineering, (2019).
10. B. Kuyumcu, C. Aksakalli and S. Delil, "An automated new approach in fast text classification," in International Conference on Natural Language Processing and Information Retrieval, (2019).
11. D. M. J. Lazer, M. A. Baum, Y. Benkler, A. J. Berinsky, K. M. Greenhill, F. Menczer, M. J. Metzger, B. Nyhan, G. Pennycook, D. Rothschild, M. Schudson, S. A. Sloman, C. R. Sunstein and E. Thorson, "The science of fake news," (2018).
12. A. Singh, N. Thakur and A. Sharma, "A review of supervised machine learning algorithms," International Conference on Computing for Sustainable Global Development, (2016).
13. V. Gurusamy and D. Kannan, "Preprocessing Techniques for Text Mining", (2014).
14. R. Atenstaedt, "Word cloud analysis of the BJGP," British Journal of General Practice, (2012).
15. M. A. G. Jivani , "A Comparative Study of Stemming Algorithms," Int. J. Comp. Tech. Appl., (2011).
16. S. Sperande, "Understanding logistic regression analysis," Biochemia Medica, vol. 24, (2014).
17. R. M. Yeung and W. M. Yee, "Logistic Regression: An advancement of predicting consumer purchase propensity," The Marketing Review, vol. 11, (2011).
18. E. Clark and . K. Araki, "Text Normalization in Social Media: Progress, Problems and Applications for a Pre-Processing System of Casual English," Procedia - Social and Behavioral Sciences, vol. Volume27, (2011).
19. Prakash M Nadkarni, "Natural language processing: an introduction," Journal of the American Medical Informatics Association, vol. 18, no. 5, (2011).
20. A. BAYAGA, "MULTINOMIAL LOGISTIC REGRESSION: USAGE AND APPLICATION IN RISK ANALYSIS," Journal of applied quantitative methods, (2010).
21. G. Laboreiro, L. Sarmento, J. Teixeira and E. Oliveira, "Tokenizing micro-blogging messages using atext classification approach," (2010).
22. L. &. S. N. &. B. M. Suanmali, "Fuzzy Logic Based Method for Improving Text Summarization.," (2009).
23. D. Palmer, "Tokenisation and sentence segmentation," in Handbook of Natural Language Processing, (2007).
24. J. Plisson, N. Lavrac and D. Mladenic, "A Rule based Approach to Word Lemmatization," in Proceedings of IS, (2004).
25. G. Grefenstette, "Tokenization," in *Syntactic Wordclass Tagging*, Springer, Dordrecht, (1999).
26. https://github.com/guillaume-chevalier/Linear-Attention-Recurrent-Neural-Network/tree/master/inkscape_drawings, last accessed 2023/04/27.
27. Saif, Mohammed & Hosam Raheem, Saif. (2020). Determine of The Most Important Factors That Affect The Incidence Of Heart Disease Using Logistic Regression Model. 14. 174-183, last accessed 2023/04/27