Dissertation on

"Understanding and Tackling the Problem of Fake News"

Submitted in partial fulfillment of the requirements for the award of degree of

Bachelor of Technology

in

Computer Science & Engineering

Submitted by:

AKHILESH NIRNA	**01FB15ECS028**
AVIRAL JOSHI	**01FB15ECS062**
HARDIK MAHIPAL SURANA	**01FB15ECS116**

Under the guidance of

Internal Guide
Dr. S Natarajan
Professor
PES University

January – May 2019

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

FACULTY OF ENGINEERING

PES UNIVERSITY

(Established under Karnataka Act No. 16 of 2013)

100ft Ring Road, Bengaluru – 560 085, Karnataka, India

PES UNIVERSITY

(Established under Karnataka Act No. 16 of 2013)
100ft Ring Road, Bengaluru – 560 085, Karnataka, India

FACULTY OF ENGINEERING

CERTIFICATE

This is to certify that the dissertation entitled

'Understanding and Tackling the Problem of Fake News'

is a bonafide work carried out by

AKHILESH NIRNA	**01FB15ECS028**
AVIRAL JOSHI	**01FB15ECS062**
HARDIK MAHIPAL SURANA	**01FB15ECS116**

In partial fulfillment for the completion of eighth semester project work in the Program of Study Bachelor of Technology in Computer Science and Engineering under rules and regulations of PES University, Bengaluru during the period Jan. 2019 – May. 2019. It is certified that all corrections / suggestions indicated for internal assessment have been incorporated in the report. The dissertation has been approved as it satisfies the 8th semester academic requirements in respect of project work.

Dr. S Natarajan Professor	Dr. Shylaja S S Chairperson	Dr. B K Keshavan Dean of Faculty

External Viva

Name of the Examiners	**Signature with Date**
1. _____	_____
2. _____	_____

DECLARATION

We hereby declare that the project entitled **"Understanding and Tackling the Problem of Fake News"** has been carried out by us under the guidance of Dr. S Natarajan, Professor, Dept. of CSE and submitted in partial fulfillment of the course requirements for the award of degree of **Bachelor of Technology** in **Computer Science and Engineering** of **PES University, Bengaluru** during the academic semester January – May 2019. The matter embodied in this report has not been submitted to any other university or institution for the award of any degree.

01FB15ECS028	**AKHILESH NIRNA**
01FB15ECS062	**AVIRAL JOSHI**
01FB15ECS116	**HARDIK MAHIPAL SURANA**

ACKNOWLEDGEMENT

We would like to express our heartfelt gratitude to Dr. S. Natarajan, Professor, Department of Computer Science and Engineering, PES University for his continuous guidance, assistance and encouragement throughout the development of this project.

We are grateful to the project coordinator, Prof. Jyothi R, for organizing, managing and helping with the project review sessions.

We take this opportunity to thank Dr. Shylaja S S, Chairperson, Department of Computer Science and Engineering, PES University, for all the knowledge and support we have received from the department.

We would like to thank Dr. B.K. Keshavan, Dean of Faculty, PES University for his help.

We would also like to thank Dr. K.N.B. Murthy, Vice-Chancellor, PES University for his support and guidance.

We are deeply grateful to Dr. M. R. Doreswamy, Chancellor, PES University and Prof. Jawahar Doreswamy, Pro Chancellor, PES University, for providing us various opportunities and enlightenment in every step of the way.

Finally, this project could not have been completed without the continual support and encouragement we have received from our family and team members.

ABSTRACT

To create a web-application and REST API that provides the ability to comprehend content from sources like images, article text, website links etc. and mine for patterns which act as metrics that aid in the detection of false information aka fake news

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

DEFINITIONS, ACRONYMS AND ABBREVIATIONS

Definitions

1. Phishing: method to entice people to reveal their personal/confidential data.
2. REST: architectural style for web services.

Abbreviations

1. CNN: Convolutional Neural Networks
2. CRUD: Create Read Update Delete
3. TRP: Target Rating Point
4. API: Application Programming Interface
5. TLD: Top Level Domain
6. RSS: RDF Site Summary
7. EXIF: Exchangeable image file format
8. REST: Representational State Transfer

CHAPTER 1

INTRODUCTION

1.1 Overview

Fake News is a term which has caught much attention in the recent past. Many organizations and individuals refer to it as “news that is intended to mislead people and sway public opinion”. While the term itself may seem rather benign and of no impending danger, the opposite is true. Scholars and experts who research Fake News consider it to be a grave threat to journalism, democracy and freedom of expression. Its pernicious nature has ebbed away peoples’ trust in institutions, endangered organizations and individuals to the benefit of a few. Furthermore, its growth has been amplified by the growth of technology. Technology has provided the perpetrators of fake news with new tools at their disposal to make Fake News seem more authentic and increase the speed at which it is spread. To make matters worse, news outlets especially news channels have a tilt towards broadcasting sensational news and as a consequence of this: Fake News which is often sensational is more likely to be spread by news outlets. Moreover, a stern competition to maintain T.V ratings has propelled the new debate format which is entertaining but often lacks objectivity. Since this format gets more viewership and does not need the rigor of fact checking or the expense of an investigation, it is not hard to see why news channels have levitated towards it and compromised their journalistic standards (which were noticeably better in the past), and therefore exacerbating the problem of Fake News. To help better explain the point on how Fake News compromises “journalism, democracy and freedom of expression” let us look at some examples and analyze their outcomes.

1.2 Problem Importance and Validation

In 2017 an Indian e-commerce website named Snapdeal was severely affected by Fake News spread on the social media platforms esp. WhatsApp. Earlier that year, the C.E.O of Snapchat (a social media platform) Evan Spiegel was accused by a person of saying that “This app (Snapchat) is only for rich people. I don't want to expand into poor countries like India and Spain”. It was later revealed that the person who accused Spiegel was a former employee who was disgruntled with the company. As it turns out no such comments were made by the

C.E.O of Snapchat, however, fake / incorrect messages quickly spread on WhatsApp which confused Snapchat with Snapdeal and thus the latter received undue backslash in India, where quite a few people used only WhatsApp as their hub for information exchange and with no possibility of seeing counter arguments to the claims made by other people the social media platform harbors parochial echo chambers. As a consequence of which both companies suffered losses. Snapchat's stock value fell by 1.9% around that time and Snapdeal saw its losses grow from 3,300 to 5,100 crores in this mishap. It is clear from this example that the person who incorrectly accused Spiegel misused his right to freedom of speech and expression thus causing harm to Individuals and organizations.

Fake News has always been a problem, but it was recognized as a grave one only after the famous US presidential election of 2016 took place. This election in the eyes of many was marred by controversies and rampant Fake News being circulated on social media platform. Apart from campaign speeches the battle for convincing voters was fought primarily on social media. And with the amount of Fake News that was being spread at the time it became hard for ordinary citizens to establish trust with either presidential candidate, as the objectivity of the candidate's policies was blurred out by the fake tweets circulating on Twitter. It was later discovered that there were foreign powers (esp. in Russia) that were circulating the Fake News and systematically targeting the election. The current president of the USA (Donald Trump) was accused of having colluded with the Russians to spread Fake News about the other presidential candidate (Hillary Clinton). An investigation was launched into this matter and it has revealed no conclusive evidence of the accusations leveled against the President Trump and his administration. Regardless of the truth of the matter, it is clear that, if the spread of Fake News was stopped at the point of its inception the resulting chaos could have been avoided. The example above clearly demonstrate how Fake News can be a threat to `Democracy and weaken trust people in democratic institutions`.

Finally, this news piece form 2018 exemplifies how fake news can undermine journalists and affect journalism. In 2018 amidst the flood in Kerala, India, Fake News was conjured by few to create communal disharmony. Forged conversion rate cards (i.e. royalty paid to some people for converting people of a different religion into their own) were spread in Kerala. A

few news channels prematurely cling onto the news thinking it was good for their TRP without doing their due diligence and fact checking / investigating further into the matter. This only played into the hands of those who spread the Fake News and caused unnecessary outrage in the country. It is clear that modern day news channels due to their poor journalistic standards become prime targets for Fake News perpetrators as they are easy to manipulate as long as the news being spread covers some sensational topic.

1.3 Reason for Existence of Problem

While the imminent dangers of fake news are harder to comprehend, the intentions behind spreading such news are not. Two words can summarize why most fake news is spread : propaganda and benefit. All fake news, either directly or indirectly, is spread to benefit someone's vested interests and is disguised as there is an underlying propaganda that is being pushed onto people.

But why is it so easy for humans to fall into such traps and lies? To answer such a question, one must delve deeper into the human psyche and try to analyze how Fake News uses our inherent beliefs and biases against us.

1. "Confirmation bias is a tendency of people to trust information that confirms their preexisting beliefs and hypothesis", as people we generally tend to disregard information that does not fit our preconceived notions.
2. Furthermore, social media is a breeding ground for online Echo Chambers, these tend to "reinforce beliefs by communication and repetition within a closed system".
3. People tend to try and align with a preexisting norm and tend to bandwagon what others are doing. This has a far-reaching consequence when it comes to how easily fake news can spread.
4. Finally, another reason why people are susceptible to Fake News is due to peer pressure. A person whose friends feel a particular way about a topic is also likely to think the same way about it, because if he were to do the opposite, he might lose their company.

1.4 Current Solutions

Social media giants such as Facebook, Google, Twitter and WhatsApp are slowly taking note of this problem and are coming up with ways to help prevent Fake News from spreading. Furthermore, the US senate has questioned the C.E.O's of these social media platforms on the steps they are taking to ensure that Fake News is not being spread by them. Listed below are a few examples of how these organizations are tackling the problem.

- Facebook in 2017 had launched a new feature in its news feeds which authenticates the news article against fact checking websites such as Snopes.net and PolitiFact.com.

Fig. 1.1 Example of how Facebook Flags Fake News

- Users can voluntarily provide their opinion/feedback on a news article, which Google then uses to filter out potentially misleading content from their searches. This feature is present on youtube platform as well.

- Additionally Google News uses also uses a fact checking mechanism and indicates the credibility of a news article.

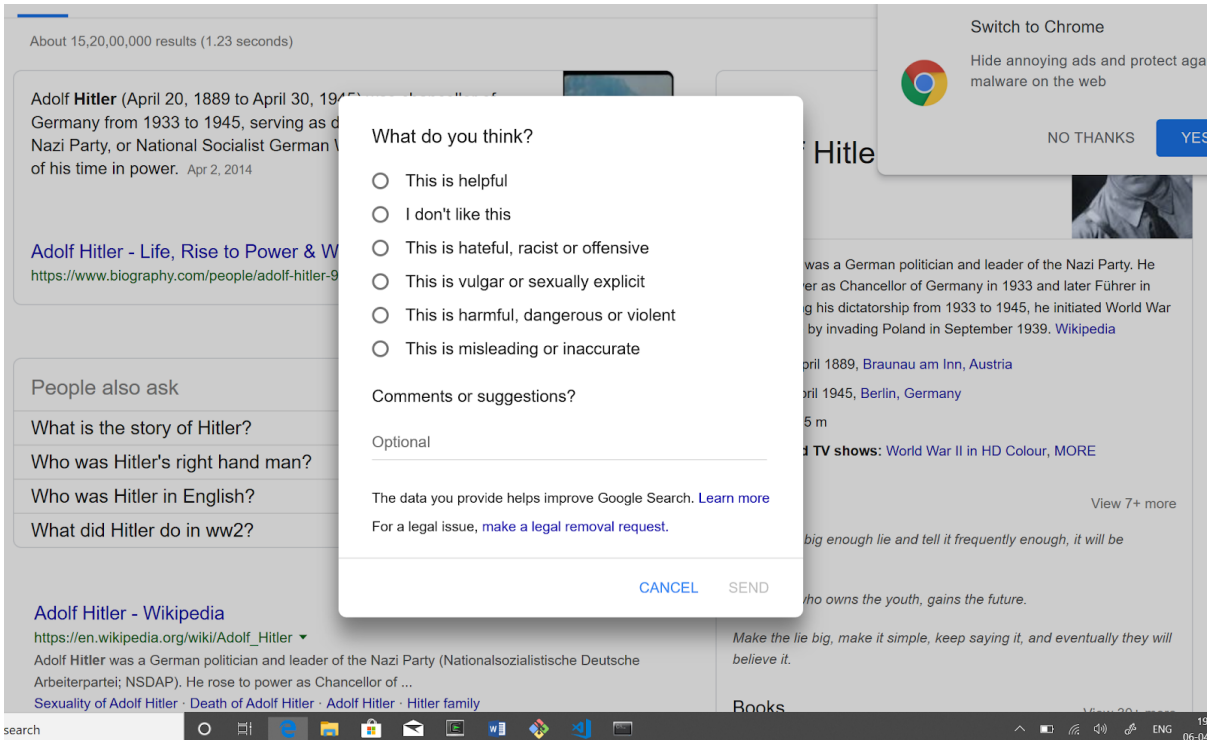

Fig. 1.2 Example of Google's Fact-Checking Mechanism

- WhatsApp has introduced a feature which lets users know if the message was written or forwarded by the sender. This can help a user identify the information he must place a grater trust on. As forwarded messages may not be read completely by the sender before he or she forwards the message they are more likely to contain misleading or incorrect information.

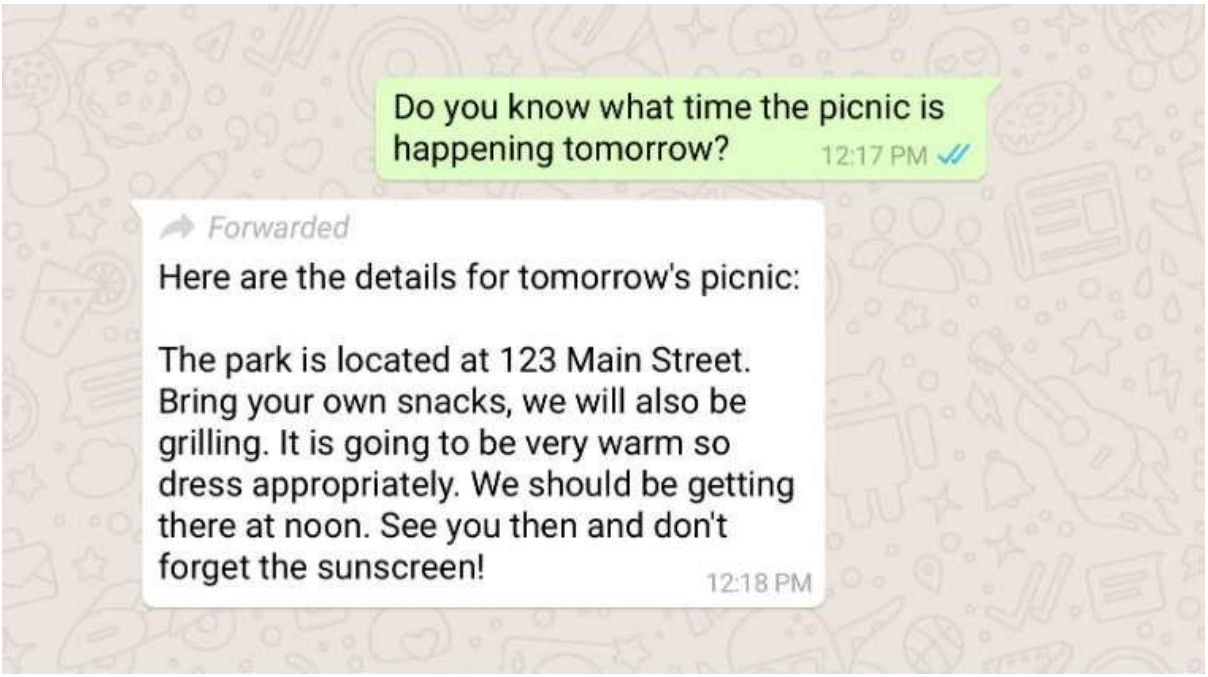

Fig. 1.3 Example of WhatsApp's Message Forwarding Mechanism

While these features do help combat Fake News to some extent their effectiveness can be disputed. Moreover, their usage seems very limited (merely to a news article). To better tackle the problem of Fake News a more comprehensive set of metrics need to be used, which cover various aspects of Fake News and not just the textual contents.

1.5 Project Objective

The inexorable wave of technological advancements has brought with it several challenges with respect to how news is regulated online. Fake news is defined as a concocted story made to deceive, often for some secondary gain by distortion of facts and is arguably one of the most serious challenges facing the news industry today. This project is intended to develop a novel framework for fake news classification. The aim is to provide a platform that enables end users to verify / reinforce their beliefs about the authenticity of a piece of news / statement in order to safeguard themselves from perpetrators of fake news which generally operate to promulgate their or someone else's vested interests.

1.6 Scope

The goal of this project is to provide a web surfer with a more comprehensive toolset, to not be consumed by the plethora of potentially misleading information online. This project aims to leverage artificial intelligence technologies, particularly machine learning and natural language processing to combat the problem of creation and propagation of fake news. Due to the probabilistic nature of these algorithms, it is difficult to ascertain the validity of our results.

In order to tackle the problem in a multi-faceted approach, we have defined a set of 6 carefully chosen metrics that will cover not only textual information but also images and websites, based on which we wish to evaluate the credibility of the given news content. Each of these metrics target a specific characteristic or pattern which can be identified using machine learning algorithms thereby helping to solve a piece of the puzzle.

We, having researched about the intricacies and challenges that can be faced to prevent the spread of Fake News aim to consolidate the results of each of the above mentioned metrics in a simple, easy-to-use web interface for the end user that can not only provide more context on a news/person and their beliefs but also save them from being used to benefit others and their propaganda. Furthermore, to make the website more easy to access by other third party application we have developed API's (Application Programming Interfaces) to access each of our metrics and research work.

This approach has the advantage of capturing more context about a news and can thus give better results when compared to existing approaches for detecting fake news. The purpose of this project is not to establish with certainty that a news is fake, but to instead provide a greater context and a deeper understanding of the authenticity of the news and then for the user himself to decide whether he should trust the claims made by the news article.

Traditional fake news detection methods have relied to a great extent on human labor and analysis. Fact checking can be really complex and might require validating many sources online which as of now is being done only by human fact checkers.

Another aim is to explore varied means of content consumption, verify the authenticity of each of them and examine the efficiency of the combined results in comparison to individual characteristics as found in the literature survey.

1.7 Outcomes

We aim to showcase our results through a web-services based application and REST API that provides the ability to comprehend content from sources like images, article text, website links etc. and mine for patterns which act as metrics that aid in the detection of false information aka fake news.

For each of the above mentioned metrics, we plan to provide a single / unified web platform that takes input from user in various formats and then displays the prediction that was arrived at by the listed components for the sub-problem they are trying to address. Their results will then be displayed in a compact, dashboard format.

Depicted below is an example of one of the ways our website could be used to detect fake news:

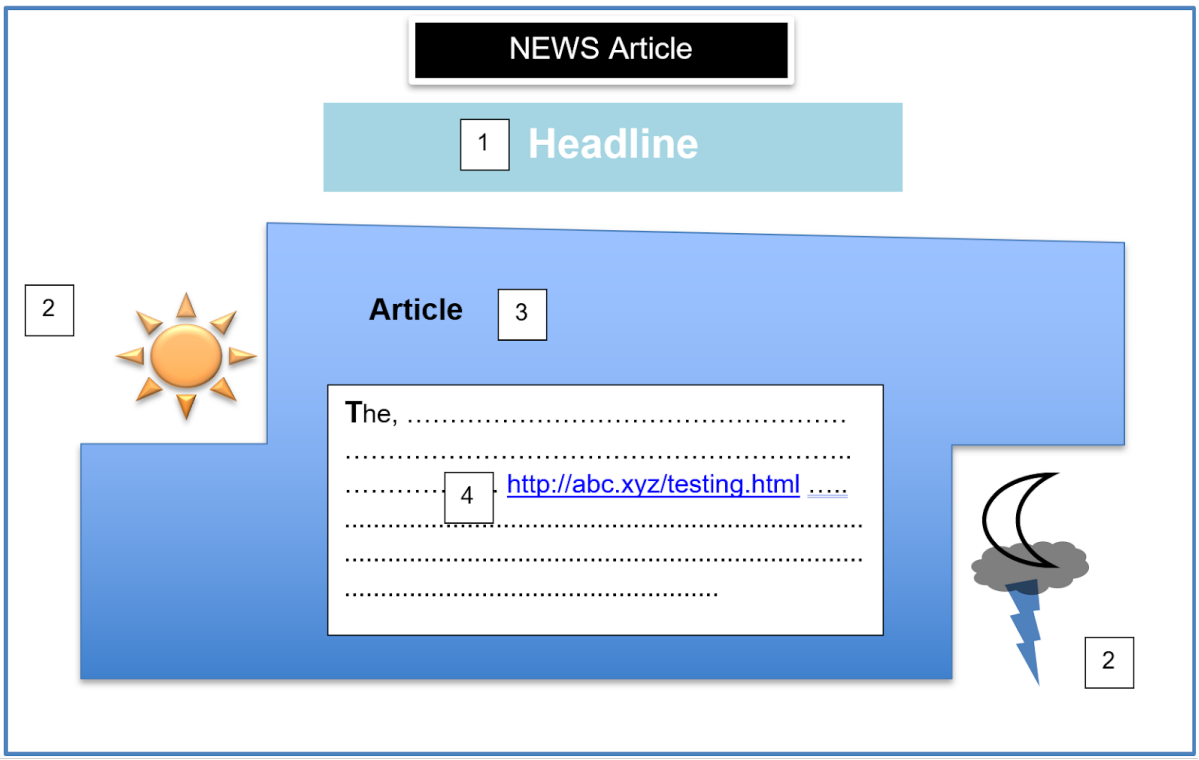

Fig. 1.4 Content Types in News Articles

Component(s)	Metric
1 & 3	Stance detection
2	Fake Image detection
1, 3	Credible Resources check
4	Fake Website detection

Table 1.1 Metrics for Different Input Types

An important dimension of this project work revolves around social networks / media platforms and therefore the web interface provides a user the ability to check for fake accounts, communities and tweets on Twitter.

CHAPTER 2

RESEARCH BACKGROUND

2.1 Literature Survey

According to the Zhou et al.[1], fake news was “best highlighted” during critical months of the 2016 US Presidential Elections and the 2016 Brexit referendum where the top twenty frequently-discussed false election stories generated 8,711,000 shares, reactions, and comments on Facebook, ironically, larger than the total of 7,367,000 for the top twenty most-discussed election stories posted by 19 major news websites. This event led to increase in distrust towards the government. The effect of fake news is not restricted to politics but also in areas such as stock markets. Although fake news is not a new phenomenon, the interest in fake news has risen because it can be created and published online faster and cheaper compared to traditional news. Majority of the population in countries like US, UK rely on social media for gaining news information, thus providing a healthy environment for spreading false information.

Zhou et al.[1] provide a comprehensive guide to current state of Fake News. Their work provides a clear and concise definition of fake news for readers and discusses about fundamental theories that can be used to help understand the problem of fake news.

The paper lists 3 ways by which fake news can be studies, they are:

1. Knowledge-based study : Deals with the content of the news.
    1. Manual Fact checking
        1. Relies on domain experts to verify a given news content
        2. Highly accurate results
        3. Costly to do and scales poorly
    2. Automatic Fact Checking
        1. Uses computer algorithms. Relies on Natural Language processing and Information Retrieval techniques.
        2. Not as accurate as Manual Fact checking
        3. Very cheap to implement

2. Style-based study: Deals with the intent of the news article.
    1. Aims to study the style of content which constitutes deception. Studying the style can used to detect deception.
    2. The inherent assumption behind is that the style of content that is used to deceive readers is different from the style of an article with truthful content.

3. Propagation based study: Studies the spread of fake news on social media by using information about the users that were involved in spreading the news.
    1. Fake News propagation patterns
        1. Unconfirmed news gets more attention easily.
        2. Fake News spreads faster than truthful news.
        3. Fake News related to politics spreads even faster.

    2. Models for fake news propagation
        1. Epidemic diffusion model: Can be represented as finite state automaton where the states are the state of the news and the edges depict the transition rates b/w the states.

    3. Propagation fake news detection
        1. Similarity based: Uses the similarity b/w propagation paths to predict whether the news is fake or not.
        2. Representation based: A generated representation of the propagation path is used to classify news. The representation can be generated by either feature engineering or using deep learning techniques.

4. Credibility based study
    1. Credibility based studies first build a relationship between news articles and components such as publishers, users, posts. The contents of the

news are not directly analyzed instead auxiliary information is used. Zhou et al.[1] discuss the credibility of the following components.

1. Headline
2. News Source
3. News Comments
4. News Spreader

Finally, Zhou et al.[1] discuss about future research opportunities in the domains of:

1. “Fake News Early Detection”
2. “Identifying check worthy content”
3. “Cross-domain (-topic, -website, -language) Fake News Studies”
4. “Deep Learning for Fake News Studies”
5. “Fake News Intervention”

The work of Riedel et al.[2] is aimed at analyzing whether the body of an article agrees with its headline. The paper used a transformation technique called the TF-IDF short for term frequency - inverse document frequency. The cosine similarity of the TF-IDF vectors of the headline and article are concatenated with their TF vectors and fed into an MLP model for classification. The paper proposes a simple yet effective architecture for detecting fake news through stance detection. The results obtained by Riedel et al.[2] can further be improved by using a Neural Network architecture different from the MLP model. A 1D Convolution model can look for spatial relationships between words while reducing the number of parameters and hence increasing the efficiency of the model.

The work of Ruder et al.[3] proposes a tool for fake news classification by performing stance detection. The tool is made available online in the form of a web service. Along with classification the tool also provides a 2D scatter plot for visualization by aggregating news articles from various sources. The focus is on longer news articles and hence Bidirectional LSTMs are used to model the input data. We plan to provide web service similar to the one proposed by Ruder et al.[3] , to allow easy access.

Ajao et al.[4] propose a model, that uses a Hybrid Convolution + Recurrent model for classifying Twitter tweets. This model uses word embeddings to convert words into N dimensional vectors, these vectors are arranged in such a way that similar words lie close in the N dimensional place when compared to words that are more distant in meaning. The word vectors are then subject to 1D convolutions followed by a GRU (Gated Recurrent Unit) layer. The intuition is that the convolution layer can model short term dependencies while the GRU layer models long term dependencies. Also et al.[4] use the state of the art neural network architectures to detect fake news. The solution offered by Ajao et al.[4] can be extended to classify longer articles that might have long term dependencies that a MLP model cannot handle.

The work of Huh et al. [5] proposes a novel method of detecting fake images. Social media platforms such a Twitter only allow users to enter a few words of text, in such scenarios images become the main proponents of fake news. The work in this paper is mainly focused around analyzing the EXIF metadata present in JPEG images. By predicting the EXIF meta-data for sub patches in an image and checking the predicted EXIF values of each patch a splice can be detected. The model makes use of Convolution Neural Networks which are known for their great prowess in dealing with image data. The findings of Huh et al. [5]boast very promising results. We think that using a similar model for analyzing the authenticity of images along with predicting stance of articles accompanying the image can be used to build a robust and comprehensive fake news detection system.

The work of Zhang et al.[6] formulates the task of evaluating the trustworthiness of news by analyzing the news article, the news subject as well as the news creators. The proposed fake news detector module makes use of 2 main components. They are:

1. Hybrid Feature Learning : In this part of the module, explicit and latent feature extraction is performed. That is, first an explicit vocabulary of unique words and subjects from the PolitiFact dataset is built. Then correlations between extracted words / subjects with labels true/fake from each article are represented as feature vectors. In the latent feature extraction phase, hidden patterns and representations of text using Recurrent Neural Networks is used to generate a latent feature vector by

concatenating the output of the last recurrent layer.

2. Gated Diffusive Unit : This unit is responsible for forming associations between features extracted from the article and the information regarding subjects and creators. The model outperforms state of the art models such as DeepWalk and Line on the PolitiFact Dataset.

The work of Peters et al.[7] proposes a unique method of obtaining word embeddings. Historically all the word embedding models used a dataset to train a recurrent model to predict nearby words (Skip-Gram) and extracted the word embeddings from the last LSTM (Long Short Term Memory) layer. This paper however generates the embeddings of words by taking a weighted sum of many LSTM layers, thereby providing greater syntactic as well as semantic information. The ELMo (Embedding Language Model) achieves state of the art accuracy on most NLP related tasks. Using the work of this paper instead of GLoVe or Word2Vec embeddings can help improve the results obtained in (paper no. 4) and to our best knowledge such an approach is yet to be tried.

While detection of fake news is important, its early detection is more necessary than ever. The work by Liu et al. [8] proposes a novel architecture that can be used to detect fakes while it is propagating on Twitter. The model analyzes the propagation path of news stories and the characteristics of users involved in spreading the news to make judgments on the authenticity of the news. On Twitter, user characteristics are hard to manipulate, they are the function of the history of contribution by the user of that account, and in case of Fake News, generally, accounts that lack certain features are responsible for spreading fake news. The proposed model takes user characteristics of 'N' number of users as input and uses a hybrid architecture, consisting of recurrent layer and 1D convolution layer in parallel and predicts whether the news is fake or not. The work of this paper can be used effectively to mitigate the harm done by fake news by stopping the news from spreading in its early stages. Liu et al.'s [8] approach can be integrated to work with text and images, and hence a more comprehensive decision regarding the authenticity of the news can be reached.

Building a sentence encoder generally involves training a model for a Language modeling task and then extracting the encoding from the outermost layer. This task however requires a lot of training data and time. To alleviate this, the work by Ranjan et al. [9] proposes a training method that uses Fake Sentence Detection as a training task for sentence encoding. The model is fed corrupt sentences and is asked to determine whether the sentence is compatible as per a language or not. Surprisingly, structuring the training task in this fashion not only takes less data and time for the model to train, but it also achieves higher accuracy on several NLP related tasks. The unique training method proposed by Ranjan et al. [9] has the scope to be implemented for training models for different tasks. We would like to carry this forward by exploring relevant use cases of this training mechanism in our work.

Convolution Neural Networks are effective tools for detection image forgery but are not infallible. The work by Gragnaniello et al.[10] explores further into this territory. Historically, many techniques have been used to fool image forgery detection tools. Most widely used were the SPAM (Subtractive Pixel Adjacency Matrix) features followed by an SVM (Support Vector Machine) and deep models such as Bayar2016, Cozzolino2017, Xception. These models use gradient minimizing techniques to make a forged image appear real to a image forgery detection tool. While SVM used some hand engineered features, the other were an end to end deep learning solution. The authors of this paper also used GANs (Generative Adversarial Networks) to attack forgery detection systems. The results obtained by Gragnaniello et al.[10] showed that all the methods of attack except GANs did not scale, that is, only the image generated by GANs was able to fool multiple forgery detection classifiers. Whereas other approaches, when trained for a particular classifier are not able to trick other classifiers. The findings of Gragnaniello et al.[10] are crucial in order to build a robust Image Splicing detector as proposed by Huh et al. [5]. By using an ensemble of forgery detection tools we can ensure that our fake news system is not susceptible to fraudulent images.

The contribution by Tosik et al.[11] does not lies in the analysis of factors affecting the detection of fake news while building a model. This paper's work provides a guideline of what works and what doesn't. For example, the paper finds the usage of simple overlap features such as n-grams very effective. Distance and similarity measures such as TF-IDF

cosine similarity, WMD (Word Mover's Distance) proved to be effective features. Hamming distance and cosine similarity between count vectors were not useful. Surprisingly sentiment features did not improve accuracy of the model. Incorporating the word count of each body-heading pair proved to be somewhat useful and balancing dataset generally involves a precision-recall trade off. We plan to incorporate the features suggested by Tosik et al.[11] in our proposed model. This will help us save valuable time while exploring relevant features.

Fake news has multi-source information such as News content, Social context and Dynamic information. The work of Shu et al. [12] claims that the datasets available right now are incomplete and tend to miss out on one or the other types of information mentioned above. Therefore Shu et al. [12] construct a multidimensional data repository that consists of two datasets incorporating the above mentioned data as dimensions in the repository that is openly available, especially for the ones doing research work as it helps them understand fake news propagation. The dataset construction can be explained as follows:

1. News Content: This was extracted from PolitiFact(A website that performs fact checking on political news.) and GossipCop(A website that performs fact checking on reports related to entertainment stories and rates them on a scale of 0 to 10, where a rating of less than 5 indicates fake news.) using crawlers.
2. Social Context: This information was obtained by using twitter's Advanced Search API where the news headlines as well as user replies and other behaviors were fetched.
3. Dynamic Information: As the word "Dynamic" suggests, the data store is periodically updated with fresh news data as well as user replies by associating the information with time-stamp.

Thus we can further extend the dataset by including other sources as well as news topics other than entertainment and politics. An optimization can also be done which involves removing noise such that the repository includes only important words/features related to fake news.

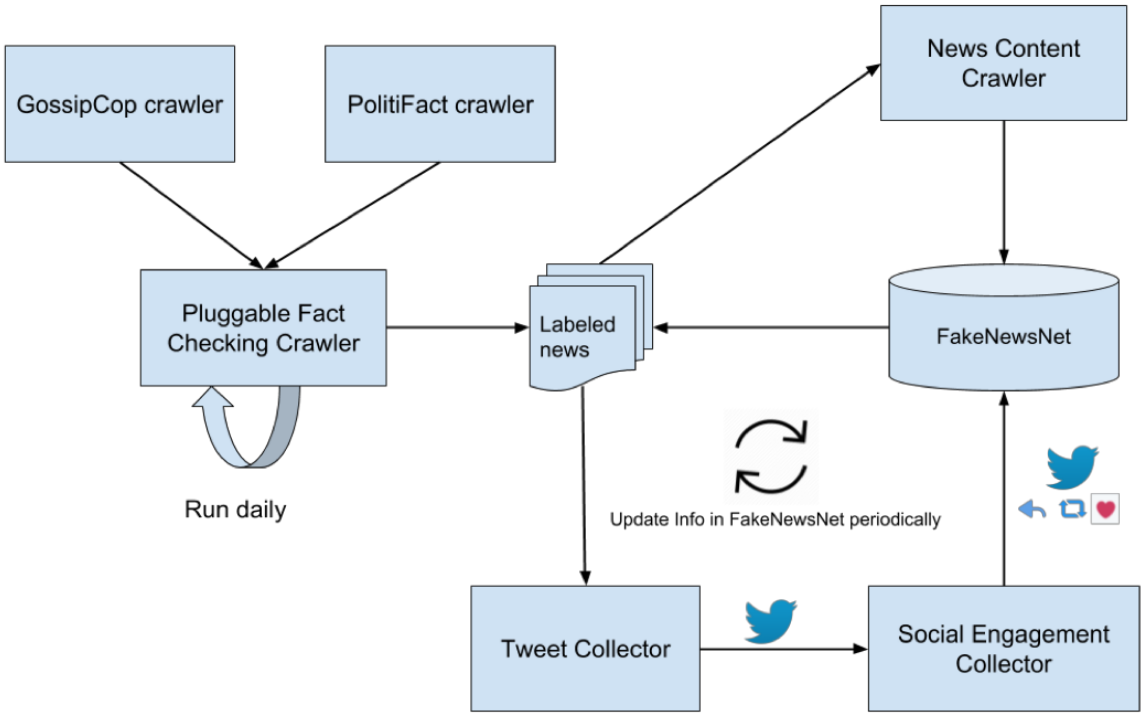

Fig. 2.1 LeSiE Framework

Social media is a platform to share individual's views and for any News based organization or Non - news based organization to make announcements. It is impossible to verify veracity of each and every article by experts, thus Choy et al. [13] propose a framework LeSiE(Lexical structure, Simplicity, Emoticon) that allows a layperson to identify the likeliness of a news article being fake. This paper describes the framework as follows:

1. Lexical Structure: Choy et al. [13] claim that forged content tends to have unique lexical features such as active verbs, implicatives, hedges, etc. Thus it exploits such features to identify fake content.
2. Simplicity: Choy et al. [13] state that fake news tends to be simple in nature i.e. the content tends to tell simple stories with lower average sentence and word length. The Coleman - Liau index was used for measuring simplicity.
3. Emotion: As mentioned by Choy et al. [13] , neutral content is mostly associated with true news, this framework uses negative and positive emotions to detect fake news. Thus, dictionaries were used to detect the sentiments.

The framework was tested on dataset of 11,523 headlines with the following distribution: There were 955 ‘pants-on-fire’ headlines, 2,257 ‘false’ headlines, 1,891 ‘barely-true’ headlines, 2,362 ‘half true’ headlines, 2,213 ‘mostly true’ headlines, and 1,845 ‘true’ headlines. For Simplicity measure: Mostly True/True categories were found to be statistically significant than other categories. For Emotions measure: It was tested against R’s LM and QDAP dictionary where results for LM was similar to simplicity results. For Lexical Structure test, Stanford NLP Algorithm was employed resulting in significant differences between the defined categories and the same was observed for verb, number, name and adjective classes.

As Liu et al. [8] worked on early detection of fake news, work by Vicario et al.[14] focuses in the same scenario but exploits the fact that humans tend to acquire information with respect to his/her own belief system (also known as confirmation bias) and selective manifestation. Thus this aims to build a generalized framework by identifying polarizing content in order to identify potential fake news i.e. early detection. The proposed skeletal structure is defined as follows:

1. Collecting fake and official news data.
2. Entities (for semantic information) and sentiments are extracted.
3. Features are derived using the previous step.
4. State-of-the-art algorithms are used for classification.

The framework was tested on Facebook pages associated with Official Italian newspapers and Italian websites that generate unsubstantial information. The classification was performed on 4 popular Machine Learning algorithms KNN, SVM, Logistic Regression and Neural Networks(Multilayer perceptron). It was found out that Logistic Regression and Neural Networks performed the best out of the four mentioned. We can extend this framework to other social networking sites such as twitter and also include images as one of the features playing a crucial role in fake news identification.

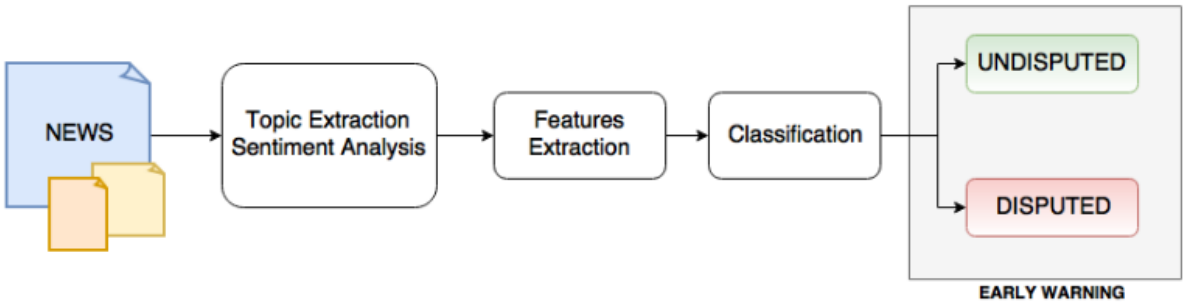

Fig. 2.2 Identification of Polarizing Content

The work of Von Luxburg et al. [15] focuses on explaining the spectral clustering techniques from scratch including the various representations of data before implementing the technique and thus proposes algorithms for different scenarios as well as the advantages and disadvantages of this technique wrt different clustering algorithms. Von Luxburg et al. [15] introduce the basic terminologies and the concepts associated for learning spectral clustering such as:

- Graphs and weights
- Graph Laplacians
- Eigenvalues and Eigenvectors

It provides us with 3 types of similarity graphs:

1. Epsilon - neighborhood: those points are connected whose pair-wise distances are smaller than some value epsilon.
2. K - nearest neighbor: connect vertices Vi and Vj if either Vi or Vj lies in the k - nearest neighbors of either one of them.
3. Fully connected: connects those points which have positive similarity and weights them with Sij.

It proposes the two type of graph laplacians:

- Unnormalized graph laplacians: L = D - W (L: the laplacian. D: degree matrix, W: weight matrix)

- Normalized graph laplacians:

$$L_{sym} = D^{-(½)}LD^{-(½)}$$

$$L_{rw} = D^{-1}L$$

It finally proposes an algorithm for both normalized and unnormalized graph laplacians.

In the day and age of internet media, people tend to know and be connected to one another through the concept of shared characteristics, often binding them into a dense interconnected social network. Over time, it has been found that the influence of this network has increasingly driven the content that is viewed by us and its effect on our perception of the world. In their paper, Frank et. al [16] showcase the importance of such online communities, their distinct features and also propose a method to identify and separate a larger network into sub-graphs based on the concepts of cohesion and separation.Here, sub-graphs are found to consist of entities that share similar features (depending on the data used) and be sufficiently different from their peers in the network through dense internal and sparse external connections. The ability to identify nodes in the network that are heavily connected to one another and poorly connected to nodes of differing characteristics is a simple, yet effective means to identify communities. Since it is not always practically possible to achieve optimal values for both cohesion and separation, the paper also identifies a method for establishing a compromise using ‘cohesive cores’ and their peripheries.

This concept is particularly interesting due to the possible complexities in large social networks which have a high user base. Some of these complexities are:

1. *Existence of hierarchies in a graph*: Some real-world groups are showcased by communities that contain the boundaries of other communities. For example, the broader community of sports may contain more focused communities such as those of football or cricket. This is the case when communities form a hierarchy, i.e. when larger communities contain smaller ones. The paper tells that the smallest communities of a hierarchy can be rather cohesive but super-communities only if their sub-communities are not very well separated. In another scenario, it is possible that there is higher overlap among nodes, apart from the boundary nodes. Due to the reduction in

cohesion, maximizing this metric may cause important communities to be excluded by the algorithm

2. *Achieving a compromise*: As stated earlier, it is difficult to choose the ideal metric for identifying correct sub-networks. This is because, depending on the characteristics of the network, there are times when cohesion is the better choice in contrast to separation. Thus, there is a need to achieve bi-objective optimization. The paper refers to external studies which identify a resolution to this complexity
3. *Size of cohesion measure*: This is a classic example seen in imperfect communities, wherein unevenly distributed links cause a large community to be perceived as multiple smaller communities. To resolve this, multiple researchers have proposed to add a measure for cohesion using the minimal conductance of all possible divergences.

To achieve a unified solution to these problems, Frank et. al [16] presented the concept of a 'cohesive core', which is essentially a node which when connected to multiple peripheral nodes is considered to be not-well-separated. Separation can be improved by including the core node's periphery into the community thereby reducing internal cohesion. This outward approach to first identify core nodes, expand the community by including peripheral nodes and compromise between overall cohesion and separation scores achieves multiple communities, which are then ranked by their size. In a complex problem such as fake news, considering the effects of content that is shared rapidly and widely, identifying the possible communities where this content is originating from and deciphering the credibility of the user and the community at large using the characteristics of the users that make the community is an important area to consider.

Jeeva et al. [17] defines a rule based approach for phishing url detection, employing association rule mining techniques viz a-priori and predictive a-priori.
The datasets used by Jeeva et al. [17] were from:

- https://www.phishtank.com for procuring confirmed phishing urls
- Legitimate urls dataset

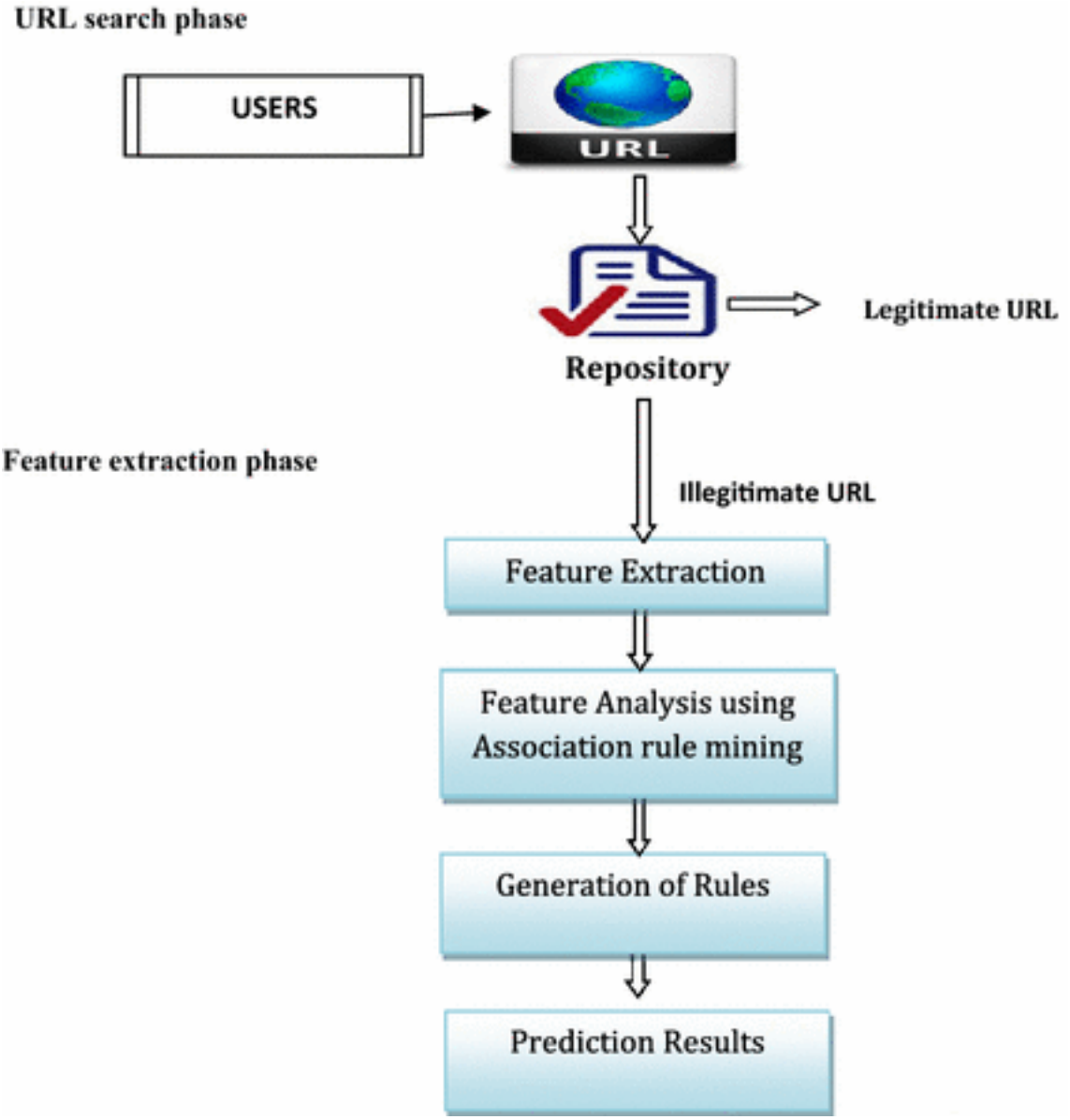

Fig. 2.3 Phishing URL detection

The system architecture for the method is defined in phases as follows:

1. URL Search Phase: This phase involves searching url repositories for checking legitimacy of the URL
2. Feature Extraction Phase: If the URL is not part of legitimate repositories, the URL is passed onto this phase which involves the following steps:
    1. Feature Extraction: This step extracts the following features from the URL
        1. Length of hostname in URL
        2. # of slashes in URL

3. # of dots in URL Hostname
4. # of terms in hostname
5. Presence of special characters in hostname
6. Presence of IP address in hostname
7. Presence of Unicode in hostname of URL
8. Use of TLS in url
9. Presence of subdomain in the URL
10. Check for keywords in URL
11. Use of TLD
12. # of dots in path portion of URL
13. # of hyphens in hostname of URL
14. Length of URL

2. Feature Analysis using Association rule mining
3. Rules Generation: The rules were generated using the following formulae:

$$Conf(X \rightarrow Y) = \frac{Supp(XY)}{Supp(X)} = \frac{P(X \cap Y)}{P(X)}$$

$$\{Support, Condifidence\} = Accuracy$$

$$Supp(X) = \frac{Number\ of\ times\ X\ appears}{N} = P(X)$$

Following were the rules extracted:

Algorithm: Apriori

Rules:

Phishing URLs

Rule 1: *if* {Transport Layer Security = http ∩ Keyword in the path portion of the URL = yes ∩ Top level domain = yes} => *class phishing*(*conf.*, 1).

Rule 2: *if* { Number of slash in URL ≥ 5 ∩ Transport Layer Security = http ∩ Keyword in the path portion of the URL = yes} => *class phishing* (*conf.*, 1).

Rule 3: *if* {Special characters = yes ∩ Transport Layer Security = http ∩ Number of terms in the host name of the URL > 4} => *class phishing* (conf., 1).

Rule 4: *if* {Dot in the host URL > 4 ∩ Transport Layer Security = http ∩ Number of terms in the host name of the URL > 4} => *class phishing* (conf., 1).

Rule 5: *if* {Number of slash in the URL ≥ 5 ∩ Dot in the host URL > 4 ∩ Length of the URL > 75} => *class phishing* (conf., 1).

Rule 6: *if* {*Special Characters = yes ∩ Transport Layer Security = http ∩ Top level domain = yes* } => *class phishing* (*conf.*, 1).

Rule 7: *if* {*Dot in the host URL > 4 ∩ Transport Layer Security = http ∩ Keyword in the path portion of the URL = yes* } => *class phishing* (*conf.*, 1).

Rule 8: *if* {*Special Characters = yes ∩ Transport Layer Security = http ∩ Keyword in the path portion of the URL = yes* } => *class phishing* (*conf.*, 1).

Rule 9: *if* { *Dot in the host URL > 4 ∩ Keyword in the path portion of the URL = yes ∩ Top level domain = yes*} => *class phishing* (*conf.*, 1).

Fig. 2.4 A-priori Rules Part 1

Algorithm: Predictive Apriori

Rules:

Rule 1: if {Number of slashes in URL $\geq 5 \cap$ *Special characters = yes* $\cap$ *Transport Layer Security = http* $\cap$ *Keyword in the path portion of the URL = yes* $\cap$ *Top Level Domain = yes* $\cap$ *Length of the URL* > 75 *} => class phishing (acc., 0.995).*

Rule 2: if {Number of slashes in URL $\geq 5 \cap$ *Unicode in URL = yes* $\cap$ *Transport Layer Security = yes* $\cap$ *Length of the URL* > 75*} => class phishing (acc., 0.995).*

Rule 3: if {Number of slashes in URL $\geq 5 \cap$ *Dots in host name of the URL* $> 4 \cap$ *Unicode in URL = yes* $\cap$ *Keyword in the path portion of the URL = yes* $\cap$ *Top Level Domain = yes} = > class phishing (acc., 0.995).*

Rule 4: if {Number of slashes in URL $\geq 5 \cap$ *Special characters = yes* $\cap$ *Unicode in URL = yes* $\cap$ *Top Level Domain = yes* $\cap$ *Length of the URL* > 75 *} => class phishing (acc., 0.995).*

Rule 5: if {Number of slashes in URL $\geq 5 \cap$ *Dots in the hostname of the URL* $> 4 \cap$ *Special characters = yes* $\cap$ *Unicode in URL = yes* $\cap$ *Top Level Domain = yes } = > class phishing (acc., 0.995).*

Rule 6: if {Number of slashes in URL $\geq 5 \cap$ *Dots in the hostname of the URL* $> 4 \cap$ *Special characters = yes* $\cap$ *Length of the URL* > 75 *} => class phishing (acc., 0.995).*

Rule 7: if {Special characters = yes $\cap$ *Subdomain = yes* $\cap$ *Number of terms in the hostname of the URL* > 4 *} = > class phishing (acc., 0.995).*

Rule 8: if {Number of slashes in URL $\geq 5 \cap$ *Dots in the hostname of the URL* $> 4 \cap$ *Transport Layer Security = yes* $\cap$ *Keyword in the path portion of the URL = yes* $\cap$ *Top Level Domain = yes* $\cap$ *Length of the URL* > 75 *} = > class phishing (acc., 0.995).*

Rule 9: if {Unicode in URL = yes $\cap$ *Keyword in the path portion of the URL = yes* $\cap$ *Number of terms in the hostname of the URL* $> 4 \cap$ *Length of the URL* > 75 *} => class phishing (acc., 0.995).*

Fig. 2.5 A-priori Rules Part 2

5. Prediction.

Based on the analysis of Jeeva et al. [17], it was found out that the following features mainly characterize a phishing url:

- Unavailability of TLS
- Absence of TLD in URL
- Presence of keyword in path portion of URL
- # of slashes in URL
- # of dots in URL
- Length of URL

CHAPTER 3

METHODOLOGY

3.1 Proposed Approach

Our approach on solving the problem includes different metrics, each of which try solving a sub-problem. Metrics are based on features of the content of a news articles, social media account, url features and images or combinations of them which can be analyzed to summarize or outline some peculiarities in them which might be indicative of their authenticity. We divide each of the metrics into modules as follows.

3.1.1 Module 1: Stance Detection

Fake news might have inherent discrepancies and contradictions present within it. It is frequently propagated by spreading flashy headlines which may not reflect the real truth of the information and is usually done to excite the reader. Since, it would be very exhausting and time consuming for a reader to consciously look for such discrepancies, the stance detection metric has been developed. It is a metric which identifies the degree of agreement or disagreement between the headline and content of an article or between two articles through the use of machine learning algorithms. Hence, automating the process will help save the readers time and therefore is a crucial step for fake news analysis.

3.1.2 Module 2: User characteristics and User Similarity to a Community

Social media is a predominant and influential source from where almost everyone learns about the ongoings in the world. Hence, it is a vulnerable area in which perpetrators can try spreading fake news. This metric is focused on identifying specific users belonging to such communities that might be spreading fake news.

To begin with, the metric will identify communities to which users on social media platforms like twitter belong to and then compute the probability of each one of them being involved in spreading fake news by looking at each account in the community. Also, it will try to identify whether an account on twitter is fake or not.

3.1.3 Module 3: Credible Sources

Fake news is tough to identify. Many 'facts' are highly complex and difficult to check, or exist on a 'continuum of truth' or are compound sentences with fact and fiction overlapping. The best way to attack this problem is not through fact checking, but by comparing how reputable sources feel about a claim. The aim of this module is first to identify important keywords in the input article text, then to conduct a thorough search and collect all articles that may be relevant to them followed by establishing the credibility of the claim by understanding the stance of the articles published by certified news outlets on the same topic. Based on the results of each article, we try to help the user to take an informed decision on whether the article is a hoax or not.

Fact checking is not always feasible to verify credibility of facts. Some of the reasons for this are: overhead of manpower to source and examine facts manually, possibility of bias and error from human lapse of judgement, difficult to parse the nuances of human language into true/false dichotomies. and the slow pace of verification per article.

This metric is better suited to help establish truthfulness of an article because it does not depend on a single database of what is true/false, can handle nuances of the human language and can parse through many articles in a short amount of time.

The process followed by this module is summarized in this flowchart:

Fig. 3.1 Credible Sources Metric Flowchart

3.1.4 Module 4: Fake Website Detection

Since developing a website is very easy in today's day and age, it becomes difficult to manually check the authenticity of any content that may be posted on a website. Often, on websites, especially which allow users to voice opinions, have greater chances of encouraging content that entice readers and also provide links to certain sites that are potentially malicious in terms of stealing sensitive/confidential data, mostly phishing.

This metric is targeted to find websites created for propagating fake news, by means of determining the authenticity of the website using the URL. Another aspect being explored it the ability to check if the website content is spam. Potentially spam websites manipulate their code as well as content for search engine optimization. Since Google is the most used search engine that is based on the page rank algorithm, in order to spread spam, website developers have been exploiting the algorithm by manipulating their content in such a way that their websites show up as top Google's top results. This metric looks at three action items and the results for any website are provided to the user to better interpret and take knowledgeable decisions based on the instructions given on website. The metrics are as follows:

1. Anchor - to - text percentage.
2. Word Count in a web page.
3. Checking for presence of exploited TLD.
4. Count of keywords in title of a webpage.

3.1.5 Module 5: Fake Image Detection

Social media platforms such as twitter have a limit on the number of words that can be used for posting. In such situations, people often resort to using images for spreading information. In such cases, images become another potential source for spreading fake news as images can be tampered with, in order to spread misinformation. Thus this module detects whether a given image is potentially fake. To identify tampered images, the metric identifies if the image has been tampered with or not. The metric also highlights the regions that have been altered..

3.2 High Level System Architecture

3.2.1 Functional Requirements

The display of outcomes has two major methods. First, through a simple, easy-to-use web application. Second, through the creation of a RESTful API.

A REST API is a stateless, efficient, and fast interface provided to developers to use the operations supported by an application. In this project, our team has provided access to each of the metrics through different endpoints. It can then be accessed by third-party applications to use these metrics in their solutions and even enhance them.

To display the results of various metrics, our team has employed a three tier architecture for a website along with a RESTful API service for 3rd party users.

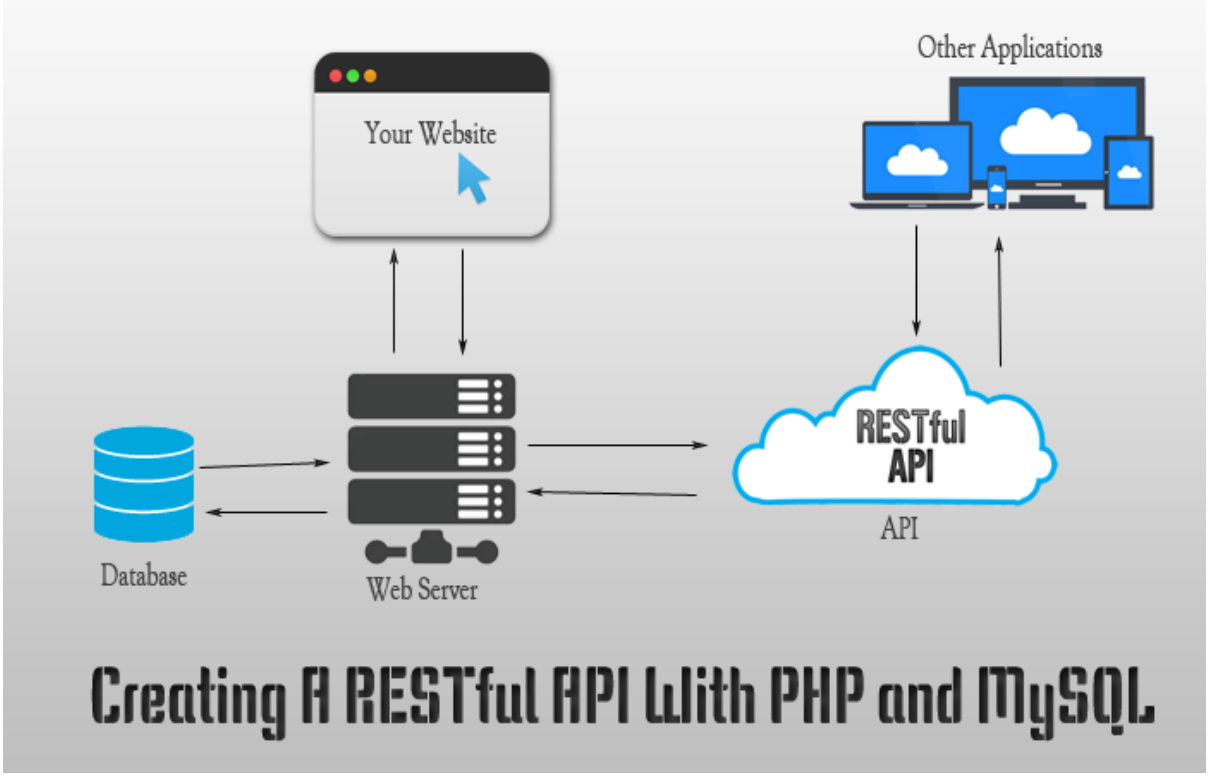

Fig. 3.2 Website Architecture

The System Architecture consists of the following components:

1. Website UI: This will be the interface which will allow the users to interact with our service on the browser
2. Web Server: This acts as a central controller to manage overall flow (handling requests/ response for both API and interactive use cases)
3. RESTful API: Our application offers access to our metrics through this API for third party usage
4. Database: used to store relevant data for the metrics for CRUD operations.

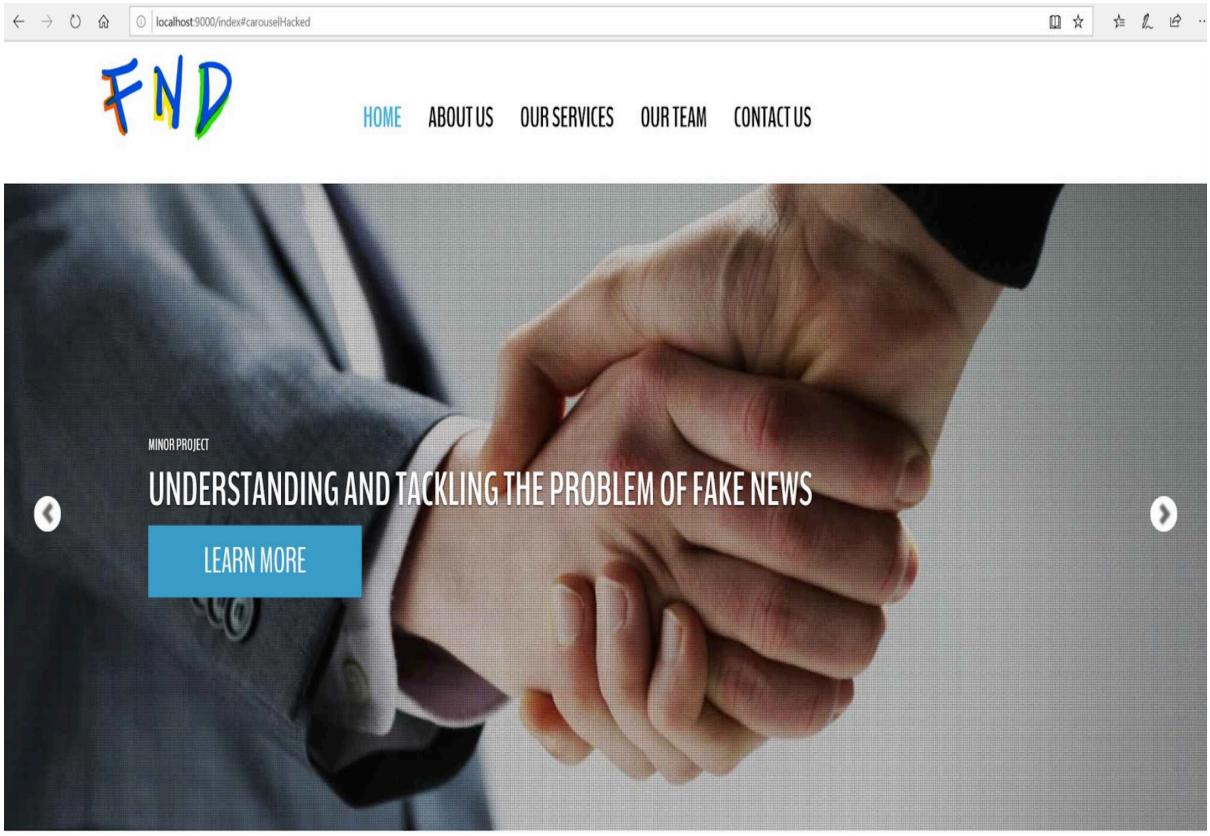

Fig. 3.3 Actual Website Home Page

3.2.2 Use Cases

Actor Name	Description
News Consumer	Uses the metrics and makes informed decisions about fake news
Developer	Creates and uses the REST API to access the implemented metrics

Table 3.1 Actor Descriptions

Use Case Name	Description
Supply input content	Enter claim in the form of text, URL, user ID or image
Interact with metrics	Access the models for metrics
Consume results	Interpret the metric results
Develop REST API	Create REST API endpoints

Table 3.2 Use Case Descriptions

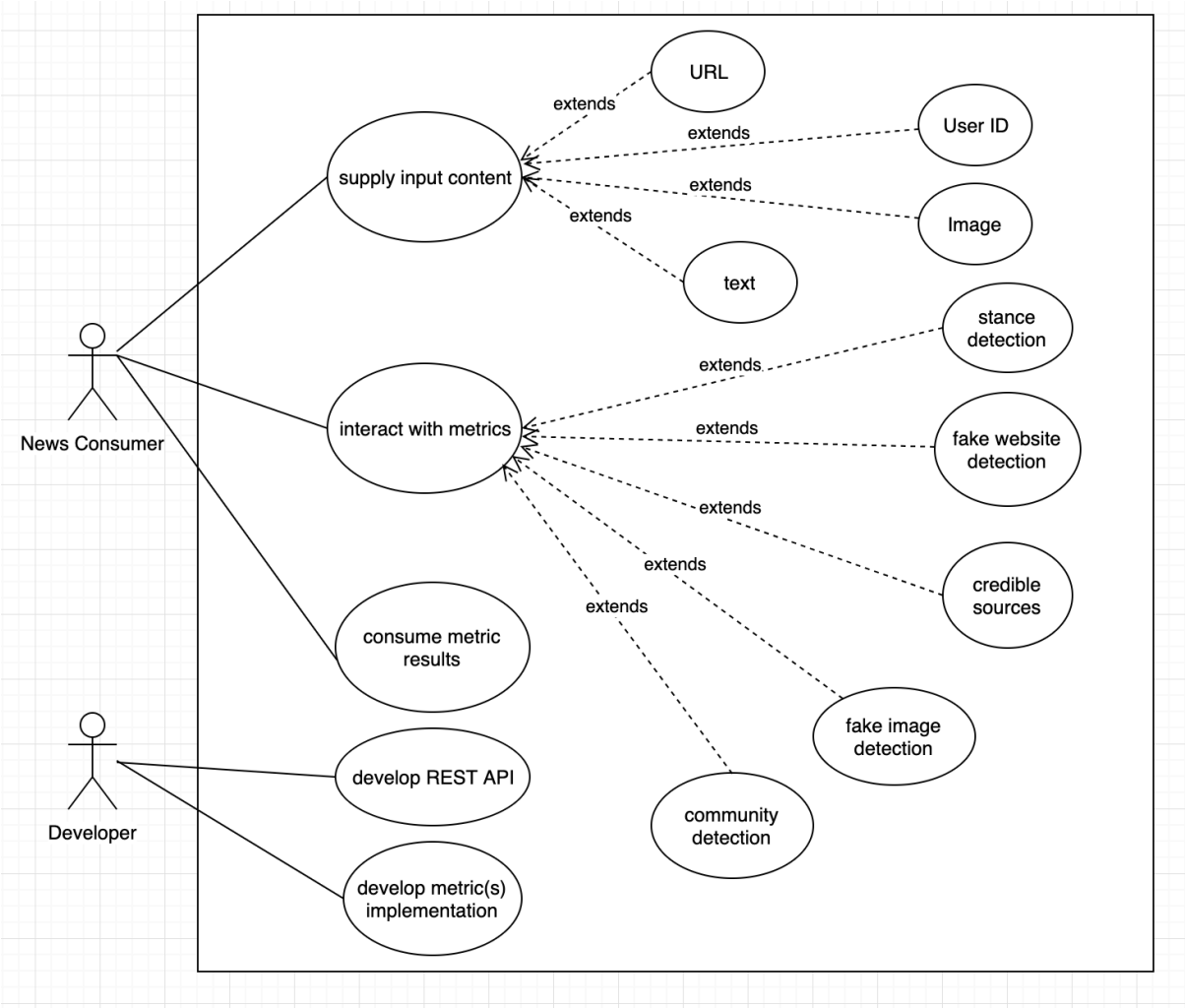

Fig. 3.4 Use Case Diagram

3.2.3 Non-Functional Requirements

1. This framework needs to support models which can provide information about fake news from a variety of input formats like text, URL, Image and User ID
2. The metrics should provide the general user with information which consists of quantifiable characteristics which aid them to decipher and decide on the credibility of the input

CHAPTER 4

ENVIRONMENT REQUIREMENTS

4.1 Hardware Requirements

The hardware requirements are as follows:

	CPU	RAM	Network	GPU	Display	Storage
Minimum	Intel i7 6700hq	12 GB DDR4	52 mbps	-	-	128GB
Recommended	Intel i7 7700k	16GB	512 mbps	Nvidia 1060Ti	Minimal, 32” LCD IPS	512 GB
Optimal	Intel Core i9-9940X	32GB	2 gbps	2 x Nvidia 1080Ti	Minimal, 32” LCD IPS	1 TB

Table 4.1 Hardware Requirements

4.2 Software Requirements

1. Operating System:
    1. Ubuntu:
        1. Ubuntu is a free and open-source Linux distribution based on Debian.
        2. It has a very rich developer community and receives latest updates often.
        3. Several python libraries used have been tested on ubuntu and seen to work fine.
    2. version: 18.04
        1. This is version is the most stable recent version of Ubuntu.
2. Flask:
    1. Description: A micro - web framework designed in python.
        1. Flasks forms the backbone of our application as it facilitates the communication between API’s and the front end.

2. version: 1.0.2
3. source: https://github.com/pallets/flask

3. Python:
    1. Description: Python is an interpreted, high-level, general-purpose programming language.
    2. version: 3.7.2
    3. source: https://www.python.org/downloads/source/

4.3 Data Requirements

Data is being collected from different sources for different metrics. Here are some of the sources we wish to refer:

1. MIB dataset consists of information about fake accounts, spam accounts, genuine accounts on twitter annotated by CrowdFlower. This dataset is particularly useful for fake account detection and clustering.

group name	description	accounts	tweets	year
genuine accounts	verified accounts that are human-operated	3,474	8,377,522	2011
social spambots #1	retweeters of an Italian political candidate	991	1,610,176	2012
social spambots #2	spammers of paid apps for mobile devices	3,457	428,542	2014
social spambots #3	spammers of products on sale at Amazon.com	464	1,418,626	2011
traditional spambots #1	training set of spammers used by C. Yang, R. Harkreader, and G. Gu.	1,000	145,094	2009
traditional spambots #2	spammers of scam URLs	100	74,957	2014
traditional spambots #3	automated accounts spamming job offers	433	5,794,931	2013
traditional spambots #4	another group of automated accounts spamming job offers	1,128	133,311	2009
fake followers	simple accounts that inflate the number of followers of another account	3,351	196,027	2012

Fig. 4.1 MIB Dataset

2. The fake-news-detection dataset on Kaggle consists of many articles with headlines and a label depicting whether the article and headline agree or disagree. This dataset is used for training a stance detection classifier.

data.csv (11.99 MB) — 4 of 4 columns — Views

	URLs	Headline	Body	Label
	3352 unique values	2831 unique values	A Potato Battery ... 4% An Embattled Ph... 3% Other (2861) 93%	0 / 1
1	http://www.bbc.com/news/world-us-canada-41419190	Four ways Bob Corker skewered Donald Trump	Image copyright Getty Images On Sunday morning, Donald Trump went off on a Twitter tirade against a member of his own party. This, in itself, isn't exactly huge news. It's far from the first time the ...	1
2	https://www.reuters.com/article/us-filmfestival-london-lastflagflying/linklaters-war-veteran-comedy-speaks-to-modern-america-says-star-idUSKBN1CD0X2	Linklater's war veteran comedy speaks to modern America, says star	LONDON (Reuters) - "Last Flag Flying", a comedy-drama about Vietnam war veterans, will resonate with Trump's America, despite, or perhaps because of, its period setting, actor Bryan Cranston said on S...	1
3	https://www.nytimes.com/2017/10/09/us/politics/corkers-	Trump's Fight With Corker Jeopardizes His Legislative	The feud broke into public view last week when Mr. Corker	1

Fig. 4.2 Fake News Detection Dataset

3. The fake image detection model was evaluated using "Columbia Image Splicing Detection Evaluation Dataset"

- This dataset contains 1845 image blocks with a fixed size.
- Image size of 128 pixels x 128 pixels.
- Most image blocks are extracted from the CalPhotos collection, and a small number from digital cameras.
- The dataset includes about the same number of authentic and spliced image blocks.

- Also contains subcategories such as (smooth vs. textured, arbitrary object boundary vs. straight boundary).

4. The Fake Website Detection model was trained using dataset obtained from https://www.phishtank.com a large database of websites where users can submit urls and they're verified whether the site is fake or not. Data from this site was combined with genuine URLs and a dataset of 7030 records was prepared with following attributes:
    - URL: All the fake and genuine URLs listed under this column
    - Label: Indicates corresponding class of URL
        1. 0 for genuine url (3494 rows)
        2. 1 for phishing/fake url (3536 rows)

CHAPTER 5

DEMONSTRATION OF OUTCOME

Based on the system architecture in section 3.2, the outcome of each metric will be demonstrated using our Web Application UI, hosted on local machine. The UI of the application is loosely coupled with the backend where every interface related to a metric will be making an API call (to the backend).

The illustration for every metric will be as follows:

Metric	API	Input	Output
Community Detection	localhost:9000/community	Twitter username	Social Graph, Fake Probability of users in social graph
Fake Image Detection	localhost:9000/fakeimage	Image	Highlights tampered regions, if any
Fake Account Detection	localhost:9000/fakeaccount	Twitter username	Probability of account being fake
Credible Sources	localhost:9000/credible	News Article	Probability of credible news sources supporting the claim
Fake Website Detection	localhost:9000/fakeWebsite	URL	Inference whether url is phishing or not
Web Spam Detection	localhost:9000/WebSpamCheck	URL	Web page characteristics

Table 5.1 RESTful API Endpoints

CHAPTER 6

PROPOSED APPROACH

6.1 Algorithms

The below sub-sections explain the details of each algorithm for the metrics incorporated in this project

6.1.1Stance Detection

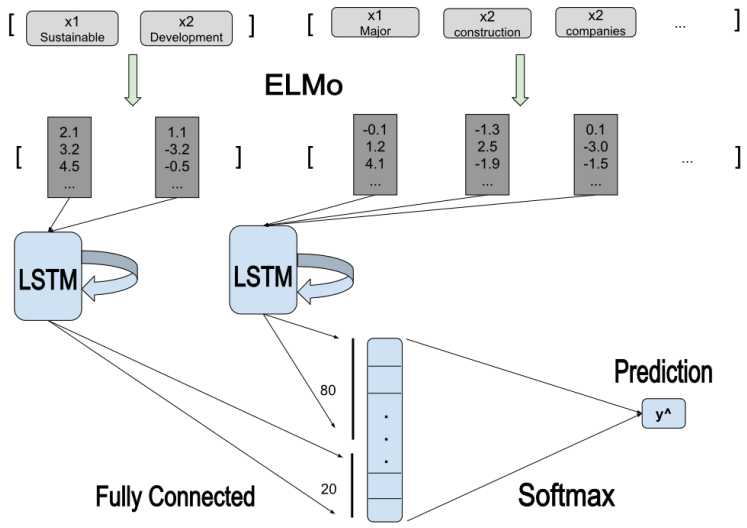

Fig. 6.1 Stance detection Model diagram

The stance detection algorithm involved the following steps:

1. Preprocessing:
    1. Removing stop words: Stop words are words that add no contextual meaning to the sentence they are present merely to make the sentence syntactically accurate.
    2. Converting letter to lowercase: the case of a word does not provide any extra semantic information and hence can be converted into lowercase to ease uniformity. However, this has a drawback, which is, proper nouns cannot be inferred accurately. But since this is of not much consequence for the problem of stance detection.

3. Converting words to sequences: words are mapped to numbers. These numbers are unique to a word and are used as an index their precomputed vector representation from a embedding matrix.
4. Padding: the sequences of words forming a sentence are converted into a list of numbers. This list of number is padded in the beginning to obtain a fixed size list. The fixed size an implementation restriction imposed by some of the machine learning libraries used. Whereas, the list is pre-padded because of the problem of vanishing gradients.

2. Training:
    1. The preprocessed sentences are input to the Machine Learning model. Here the sequence is first converted into a vector representation. The vectors represent the words in an ‘n’ dimensional space, where words with similar meanings are closer than words that are different in meaning.
    2. The word vectors are then fed into a recurrent neural network. The RNN is essentially a temporal deep neural network which takes in input sequentially from the beginning and maintains information about previous words. After all the word vectors are processed the model generates another vector which sums up the information of all the input word vectors.
    3. The previous stage is done independently until this point for the 2 input sentences that are to be analyzed.
    4. Finally the LSTM output of the 2 sentences are concatenated and passed through a fully connected layer and then squashed using the softmax function to obtain a single prediction.
    5. This prediction is contrasted with the ground truth value for the 2 input sentences.
    6. If the outputs do not match the weights of the Fully connected and LSTM layers are adjusted to make the predicted outputs closer to the ground truth.

3. Testing
    1. The saved train model is loaded and the sentences are taken as input.
    2. The sentences are cleaned is a manner similar to that done before training.
    3. The sentences are passed to the model and an output is generated. But only this time the weights of the neural networks are not adjusted.

6.1.2 Fake Account Detection

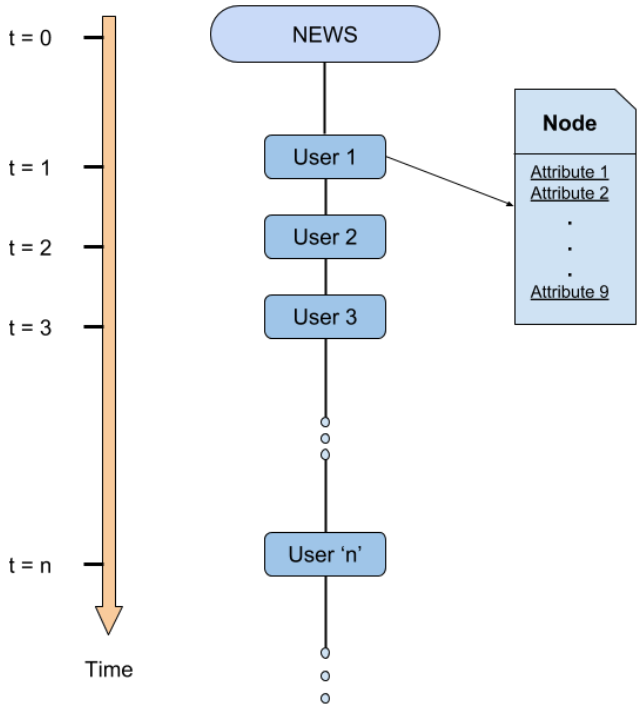

Fig. 6.2 Fake Account detection model flow chart

Algorithm used for Fake account detection on twitter :

1. Selecting relevant features:
    1. First a dataset was selected which had twitter accounts details with a ground truth label indicating whether the account was genuine, a fake follower account, spam account. A dataset called the MIB (my internet bubble) was found suitable for this task.

2. By performing data analysis and data visualization it was determined that certain features had a higher correlation and were more useful in predicting the which of the aforementioned categories the account belonged to.

2. Selecting a good ML model
    1. Using the sklearn library in python we were able to quickly and accurately choose an ML classifier for the task of Fake Account Detection.
    2. Using cross validation and testing sets a good ML model was chosen and further hyperparameter tuning was done to reduce overfitting on the dataset.

3. Testing the model
    1. After the model is trained as described in step 2 we select a tweet from a popular account on twitter, i.e. one which is followed by many users on twitter and exerts a lot of influence in the social sphere.
    2. Now we look at the latest tweet made by the user and use the twitter API to determine the individual accounts that retweeted the same.
    3. After the individual users are identified their characteristics are extracted and the trained model is used to predict a label indicating whether the account is genuine or not.
    4. Step 3c is repeated for all retweeters and all the fake accounts are identified.

6.1.3 Fake Image Detection

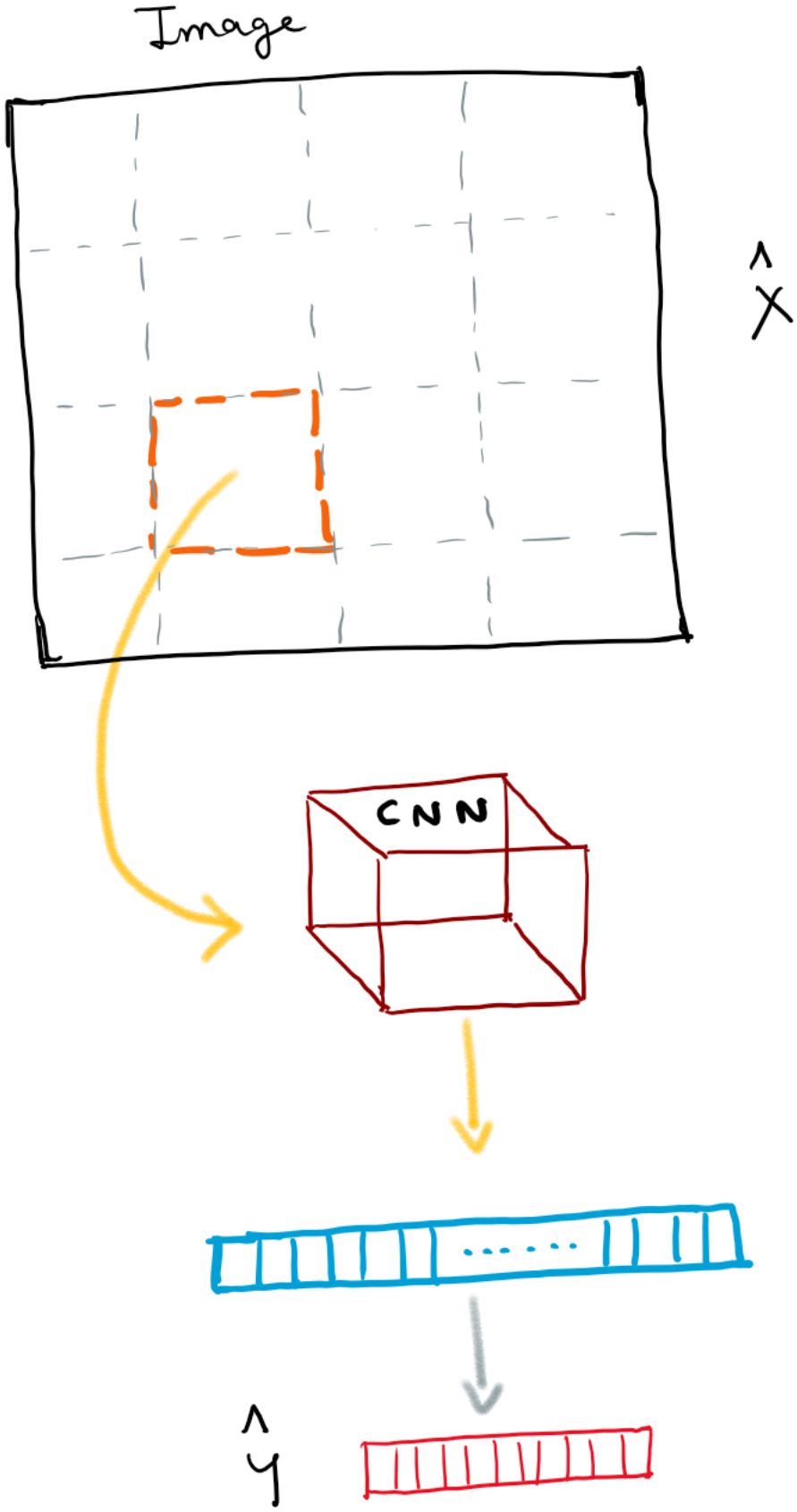

Fig. 6.3 Fake Image Detection

Algorithm for Fake Image Detection :

1. An image which is to be detected for anomalies is taken as input. These anomalies are a result of human manipulation.
2. The image is then divided into subregions where each region is of fixed size.
3. Each subregion is considered as an independent image and is passed to a pre-trained model which computes its EXIF metadata attributes.
4. Step 3 is repeated for all subregions and the computed metadata attributes are analyzed.
5. Since all the subregions belong to an image and taken by one camera only they must all have the same value of EXIF metadata. If not, the region(s) which are the most divergent will be the ones which have been manipulated.
6. The divergence is determined by using Mean Shift algorithm

6.1.4 Community Detection

1. Building a connections graph
    1. To form communities a graph is built first. Like any graph the graph comprises of nodes and edges.
    2. The nodes of the graph are accounts on twitter and an edge exists between 2 nodes if the 2 nodes are friends on twitter.
    3. The graph is an undirected graph.

2. Detecting Communities
    1. To detect communities we need a way to represent the account characteristics mathematically.
    2. The user description present on twitter is taken as input to a document 2 vector model. The doc2vec model generates a 100 dimensional feature vector.
    3. Vectors are generated for all account descriptions and then spectral clustering is performed to group them into communities.

3. Fake Community Detection
    1. After detecting communities as described in step 2 the account characteristics of every account in community is analyzed using the fake account detection model.
    2. The probabilities of the accounts in a community are used to determine whether a community is fake or not.

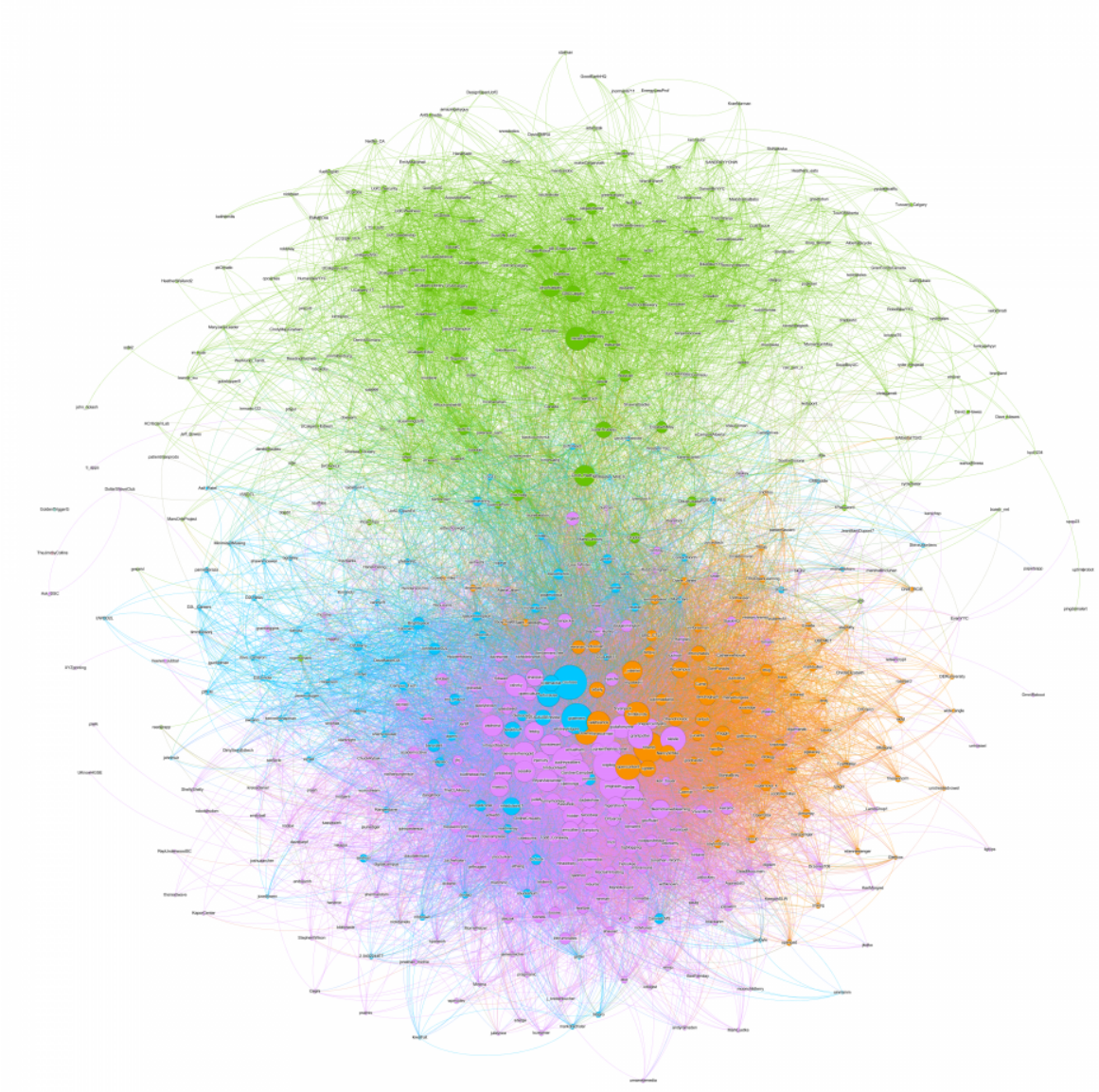

Fig. 6.4 Community Detection Algorithm

6.1.5 Credible Sources

This metric is based on the stance of articles published by reputable news sources and fact checking organizations.

The credible sources algorithm is as follows:

1. Extract Keywords: The first step is to extract the important keywords from the input article. For this, the usual bag of words model was found to be inefficient as it identifies only single-word terms and is based on the tf-idf concept. In order to consider word sequence information and also identify multi-word phrases, a graph-based ranking algorithm is used. This algorithm is seen in other places like Google's Pagerank algorithm.
    1. Words are tokenized and annotated with parts-of-speech tags. This is done only for individual words, not for n-grams
    2. Words are added to the graph as vertices, but first filtered on all lexical units using a syntactic filter
    3. An edge is added between lexical units that co-occur within a window of N words of each other to obtain an unweighted undirected graph
    4. The vertices ranking algorithm is run until convergence
    5. The vertices are ranked by their score and the top T words are selected as keywords
    6. If vertices in the top T keywords appear as adjacent vertices in the graph, they are collapsed to form a multi-word expression/phrase.

2. Search the internet for relevant articles: This is done so that an efficient search can be conducted to identify the most relevant published articles through a database of collected news articles
    1. Based on the results of step 1, the extracted keywords (single or multi-word phrases) are passed on to the Event Registry API. This is a freemium service using which relevant articles can be collected

2. There is flexibility to collect articles based on their relevance, website importance ranking and popularity score on social media. Based on the requirement, it is possible to target the data collection process

3. Identify the stance of each article
    1. Now that we have a list of articles collected from reputable sources which are relevant and related to the input article, we need to examine the stance of each article against the input. In other words, we need to check if each collected article supports or rejects the claim proposed in the input article
    2. For this, we supply both texts to our stance detection algorithm optimized to compare to articles, as opposed to comparing an article's headline with its body. The methodology followed for the stance detection algorithm is mentioned in section 1.7.1

4. Semantic Similarity: Another metric to compare the combinations of input claim and each collected article is to check their semantic similarity
    1. To compare the 2 articles, we first create a "semantic fingerprint" of each article. A Semantic Fingerprint is a sparse distributed representation of the word. Similar words and synonyms have similar fingerprints.
    2. When the fingerprints of each word in the article are placed one over another, and a threshold is applied, we obtain the semantic fingerprint of the article
    3. These fingerprints are then compared using matrix operations to obtain the similarity score.
    4. For convenience, we use the Jaccard distance measure. The closer this score is to 1, the more dissimilar are the 2 articles

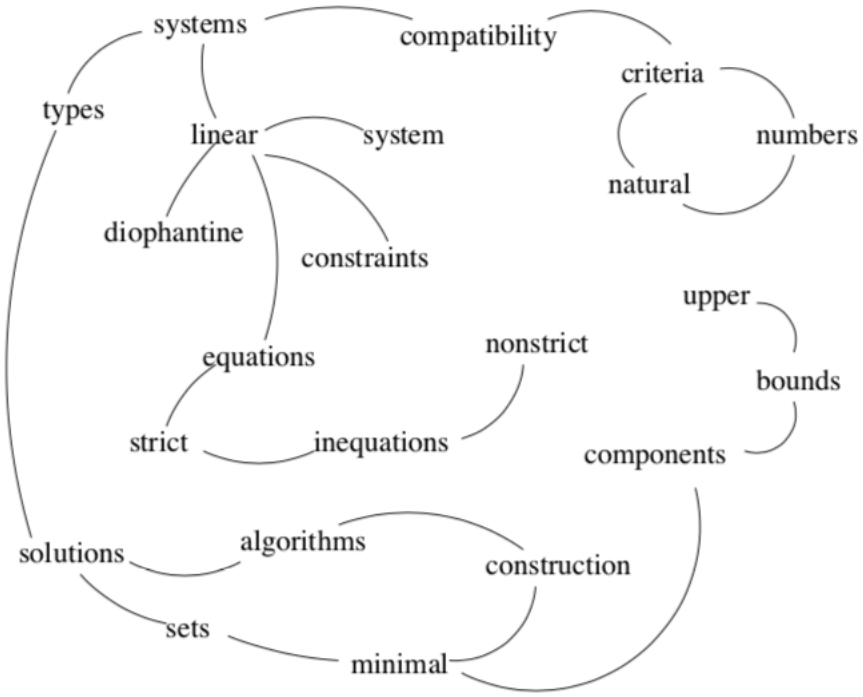

Fig. 6.5 Sample Text Graph

6.1.6 Phishing URL Detection

This metric makes use of features of URL to detect whether a given URL is potentially phishing or not. Any given URL was passed through a Rule Based Classifier as proposed in (paper name) and a Machine Learning based Classifier.

The Output of this metric is:

- 1 if the URL is potentially a phishing URL, and
- 0 if the URL is not phishing URL.

The algorithm is as follows:

1. Machine Learning Based Classifier:
    1. Data Collection:
        1. Data was collected from website called https://www.phishtank.com (a large database of websites where users can submit urls and they're verified whether the site is fake or not) by making API calls to the website.

2. A Dataset of 7030 phishing + genuine URLs was made with the columns:
    1. URL: All the fake and genuine URLs listed under this column
    2. Label: Indicates the corresponding class of URL - 0 for genuine url (3494 rows) and 1 for fake URL (3536 rows)
2. Once the dataset was obtained a feature set was prepared (function written for each feature) consisting of the following features. Each features takes the URL as an input and returns an integer as the feature value

Sl. No.	Feature Name	Method
1	No. Of dots	Extract frequency of dots using count()
2	Presence of hyphen	Extract frequency of hyphen using count()
3	URL length	Calculate length using len()
4	Presence of @	Extract frequency of @ using count()
5	Presence of //	Extract frequency of //using count()
6	No. Of sub-directories	Extract frequency of sub-directories using count()
7	No. Of sub-domains	Extract frequency of sub-domains using count()
8	Domain length	Calculate length using len()
9	No. of queries	Extract frequency of queries using count()
10	Presence of IP address in hostname	Use library ipaddress
11	Presence of Suspicious TLD from symantec.com	Check the following TLDs: Zip, Cricket, Link, Work, Party, Gq, Kim, Country, Science, Tk
12	Presence of Suspicious Domains extracted from trend micro's top suspicious domains	Check the following domains: 1. luckytime.co.kr 2. mattfoll.eu.interia.pl 3. tafficholder.com 4. dl.baixaki.com.br 5. bembed.redtube.comr 6. tags.expo9.exponential.com 7. deepspacer.com 8. funad.co.kr 9. trafficconverter.biz

Table 6.1 Fake Website Feature Extraction Methods

3. The following Machine Learning algorithms were tried out: Decision Trees, SVM, Random Forest Classifier, XGBoost. XGBoost was the one with highest accuracy thus it was used as the final classifier for the project.

2. Rule Based Classifier:

This classifier also employs a feature based approach and was built by referring to Jeeva et al. [17]. This paper defines a rule based approach for phishing url detection, employing association rule mining techniques viz a-priori and predictive a-priori. The paper, after analyzing various features extracted from the URLs comes up with a set of important URL features as rules to determining whether URL is fake or not. The algorithm is as follows:

A sub part of this metric is Web Spam Detection:

This method evaluates a website based on the following factors:

1. Number of Words in a webpage: This was calculated as follows:
    1. Check whether website is active or not, if the website is inactive display appropriate message.
    2. A library called “Beautiful Soup” was used to extract HTML content from webpage.
    3. Once HTML content is extracted, it is cleaned to extract the important text.
    4. Regex was used to count number of words in the text.

2. Count of Keywords in title: This was extracted as follows:
    1. Check whether website is active or not, if the website is inactive display appropriate message.
    2. A library called “Beautiful Soup” was used to extract HTML content from webpage.
    3. Use the library’s parser to get text of title tag of the webpage.
    4. Use nltk’s tokenizer to tokenize the title text.
    5. Remove stop words from the tokenized text.
    6. Return the count of keywords in title.

3. Safety of TLD: The website was checked whether it uses safe TLD or not. This was matched against spamhaus' list of most exploited TLDs:
    1. Extract TLD from the URL.
    2. Check against the list of exploited TLDs
    3. Return the inference.

4. Text to anchor ratio: this was extracted as follows
    1. Check whether website is active or not, if the website is inactive display appropriate message.
    2. A library called "Beautiful Soup" was used to extract HTML content from webpage.
    3. Count number of words in text using regex.
    4. HTML content is cleaned to get the text from webpage.
    5. The library's parser was used to check whether anchor tags had text in them and keep a count for same.
    6. Calculate text to anchor text ratio and return the result.

CHAPTER 7

RESULTS

The end goal of the project is to give the users the ability to make wiser decisions when drawing conclusions from an online news source.

7.1 Community Detection

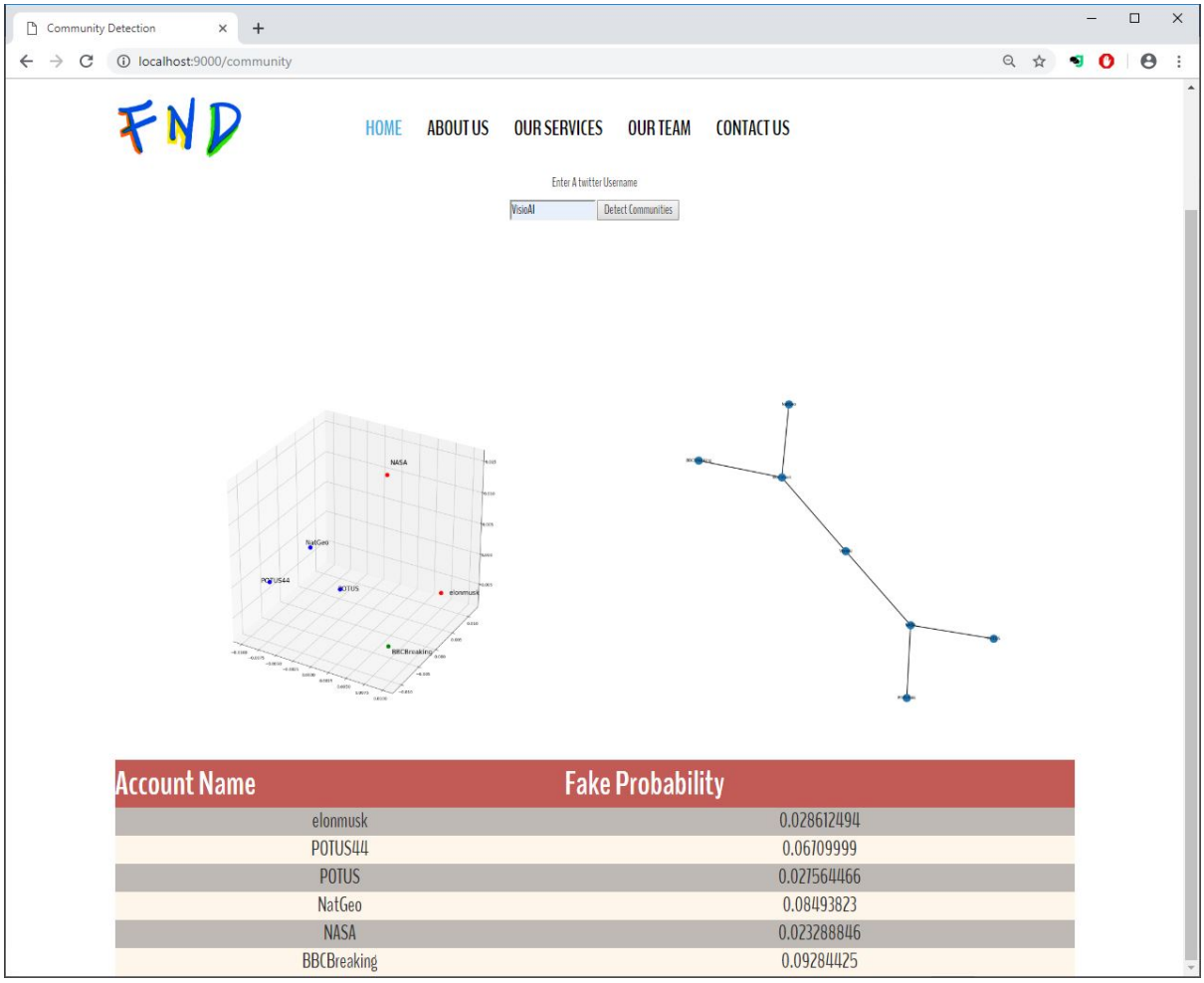

Fig. 7.1 Community Detection Website Result

Runtime : 5-10 seconds depending on internet connection and the size of the communities.

Performance: The Community detection algorithm implicitly uses 3 Machine Learning models:

Model	ML Classifier	Task	Accuracy	Run Time
Fake Account Detection Model	XGBoost	Supervised Learning	92%	~ ms per account
Clustering	Spectral Clustering	Unsupervised Learning	72%	~ms
Document 2 Vector	Paragraph Vectors	Unsupervised Learning	-	~ms

Table 8.1 Community Detection ML Results

7.2 Fake Account Detection

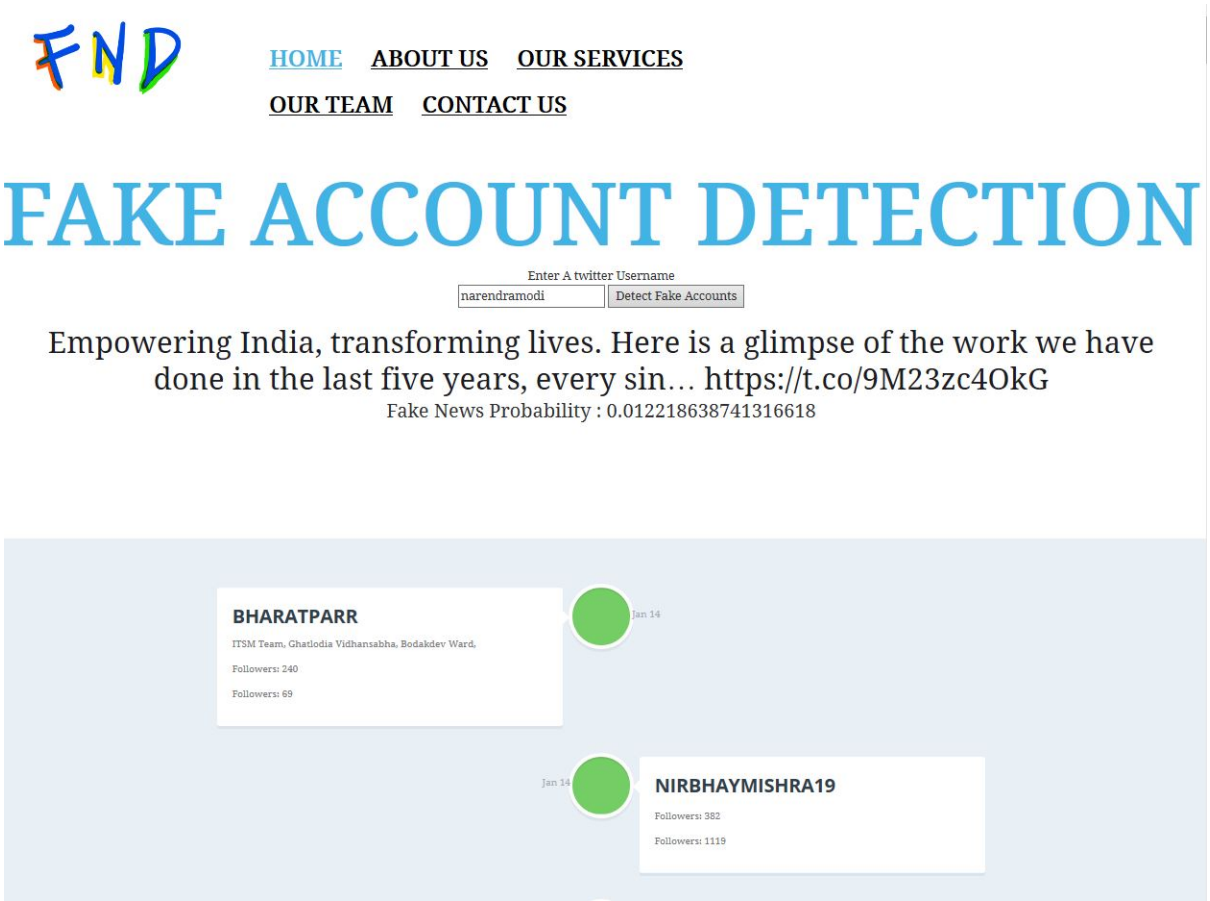

Fig. 7.2 Fake Account Detection Website Result - 1

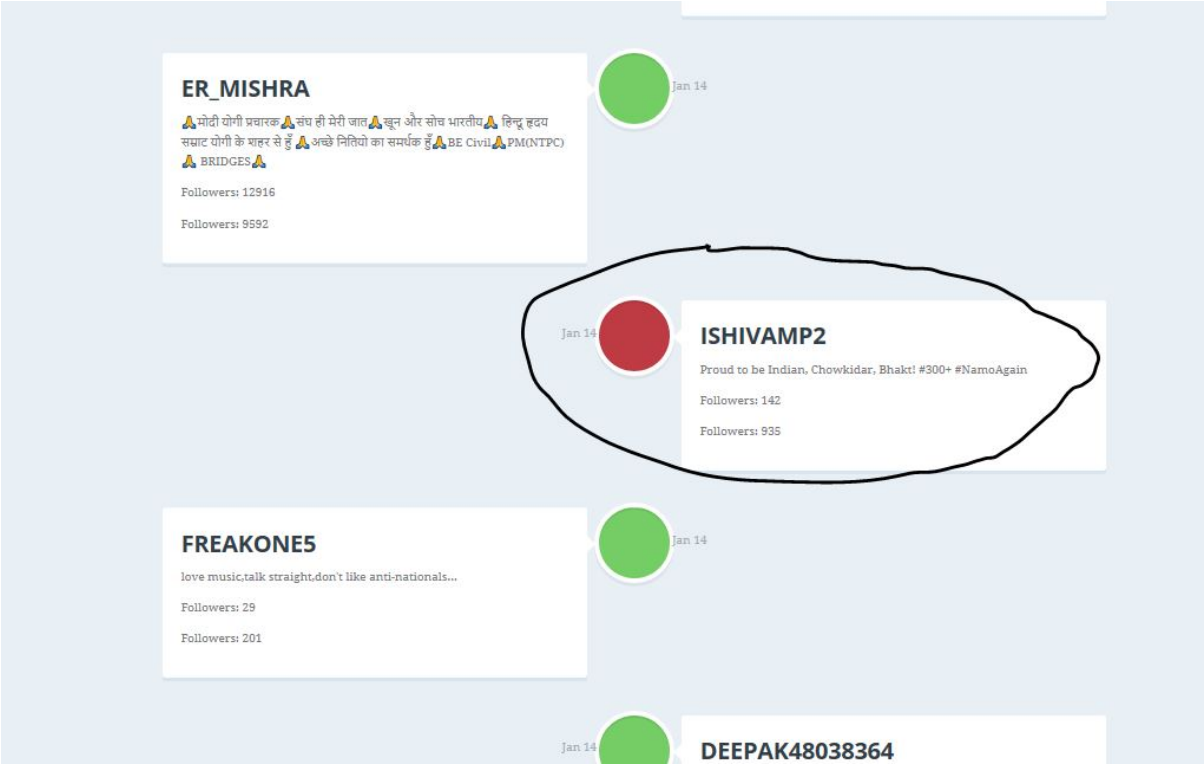

Fig. 7.3 Fake Account Detection Website Result - 2

Runtime: 5-20 seconds depending on internet connection and the number of tweets to be analyzed.

Performance: Fake Account detection was trained using the MIB dataset.

The Dataset consisted of details of twitter with a label indicating whether the account was fake or not.

Genuine Accounts	3,474
Fake Accounts	3,351

Table 7.2 Fake Account Detection Dataset Counts

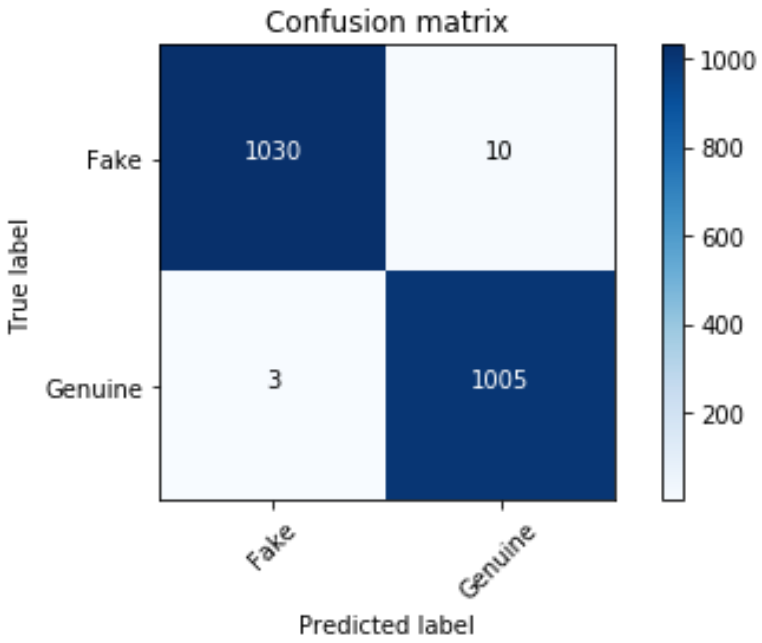

Fig. 7.4 Confusion Matrix of XGBoost Model

Accuracies of the different ML models trained on the dataset are listed below:

Model	Accuracy
XGBoost	99.5%
Random Forest	99.1%
K Nearest Neighbors	84%
Support Vector Machines	95%
Decision Trees	93%

Table 7.3 Fake Account Detection ML Results

7.3 Fake Website Detection

The Machine Learning algorithms that were used had the following accuracies:

	Cross Validation Accuracy	Unseen Data Accuracy
Decision Trees	85.16%	66.7%
Support Vector Machines	84.88%	68.8%
Random Forest	89.88%	77.2%
XGBoost	91.35%	79%

Table 7.4 Fake Website Detection ML Results

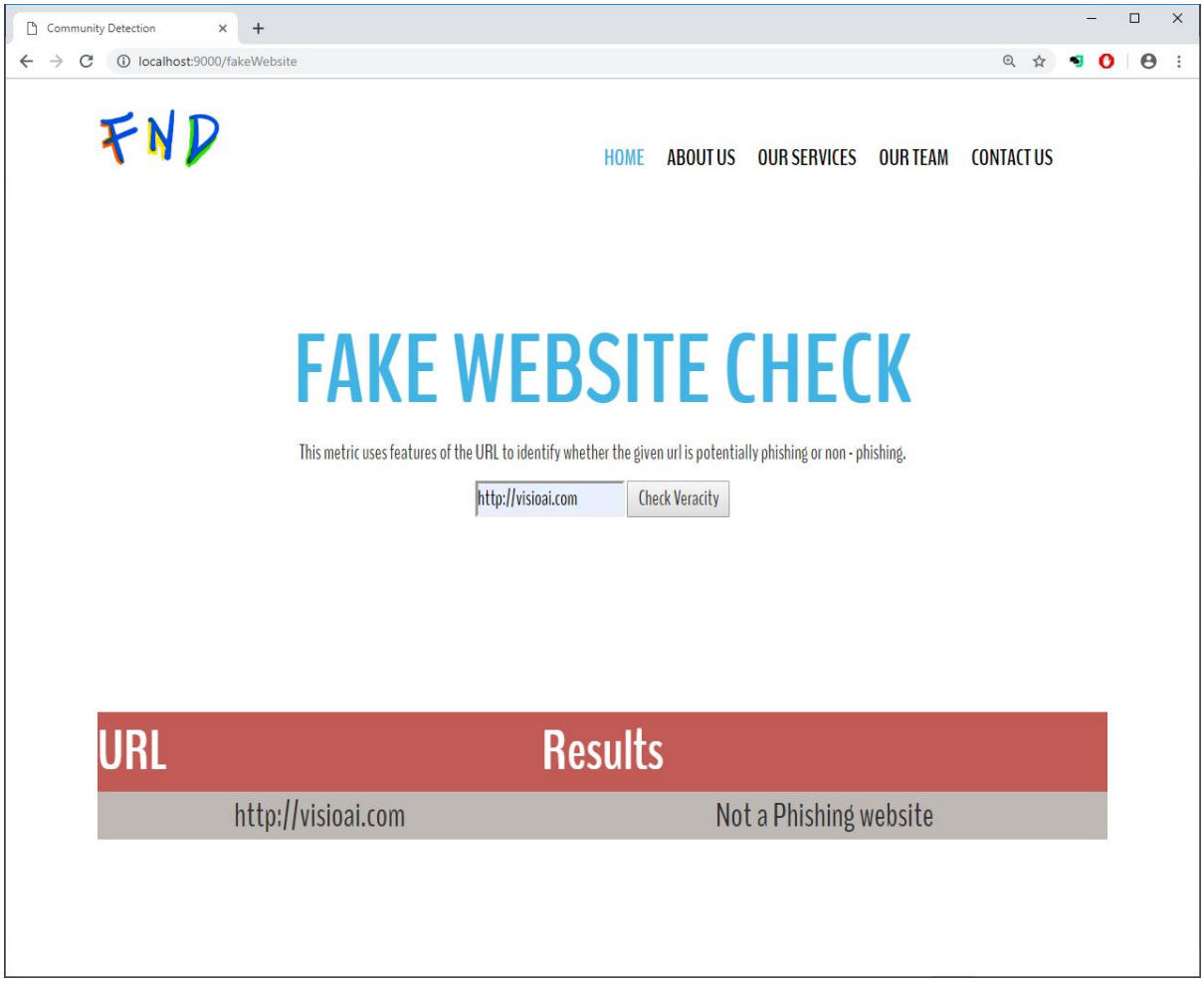

Fig. 7.5 Fake Website Check Result

7.4 Spam Website Detection

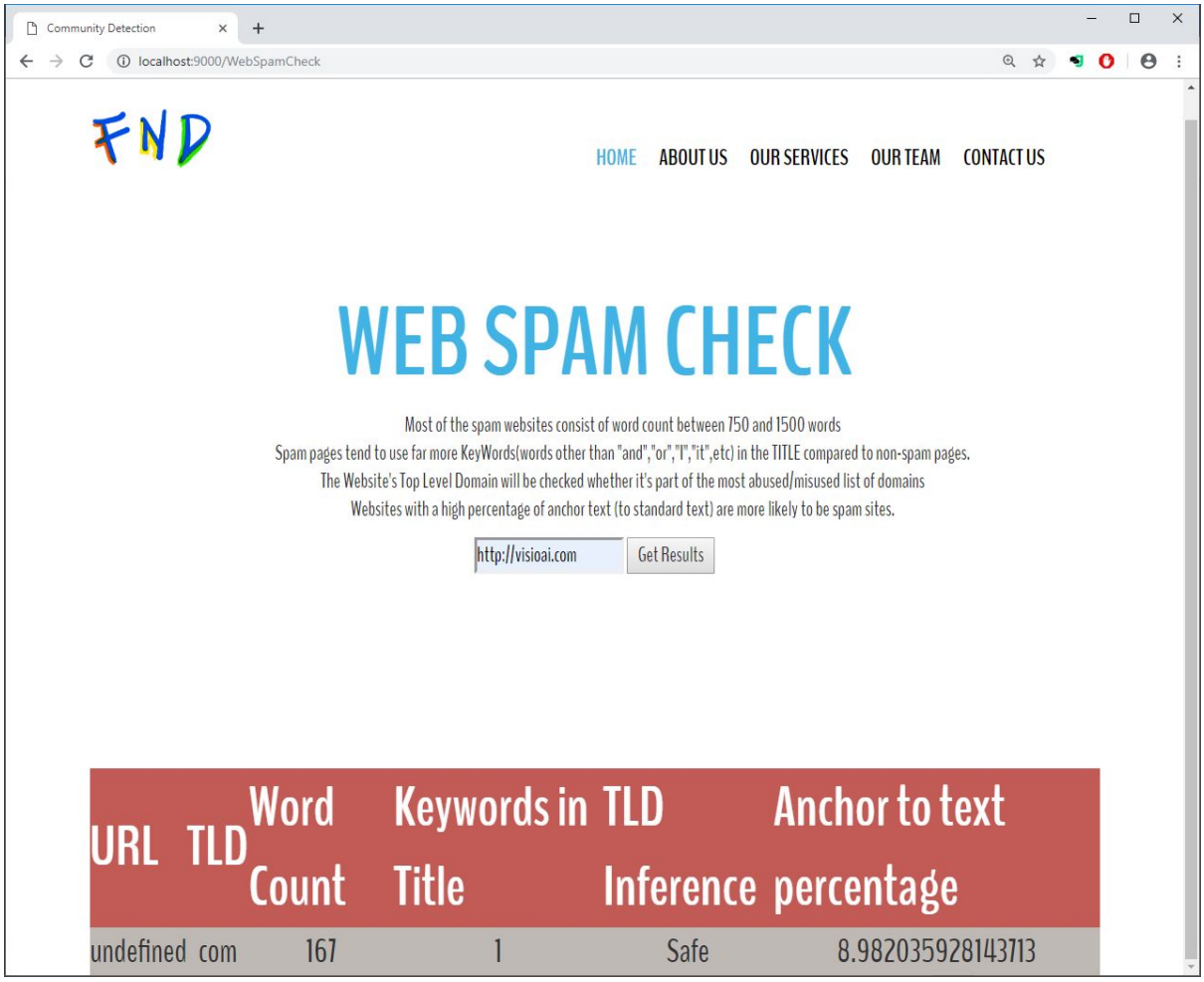

Fig. 7.6 Web Spam Detection Result

7.5 Fake Image Detection

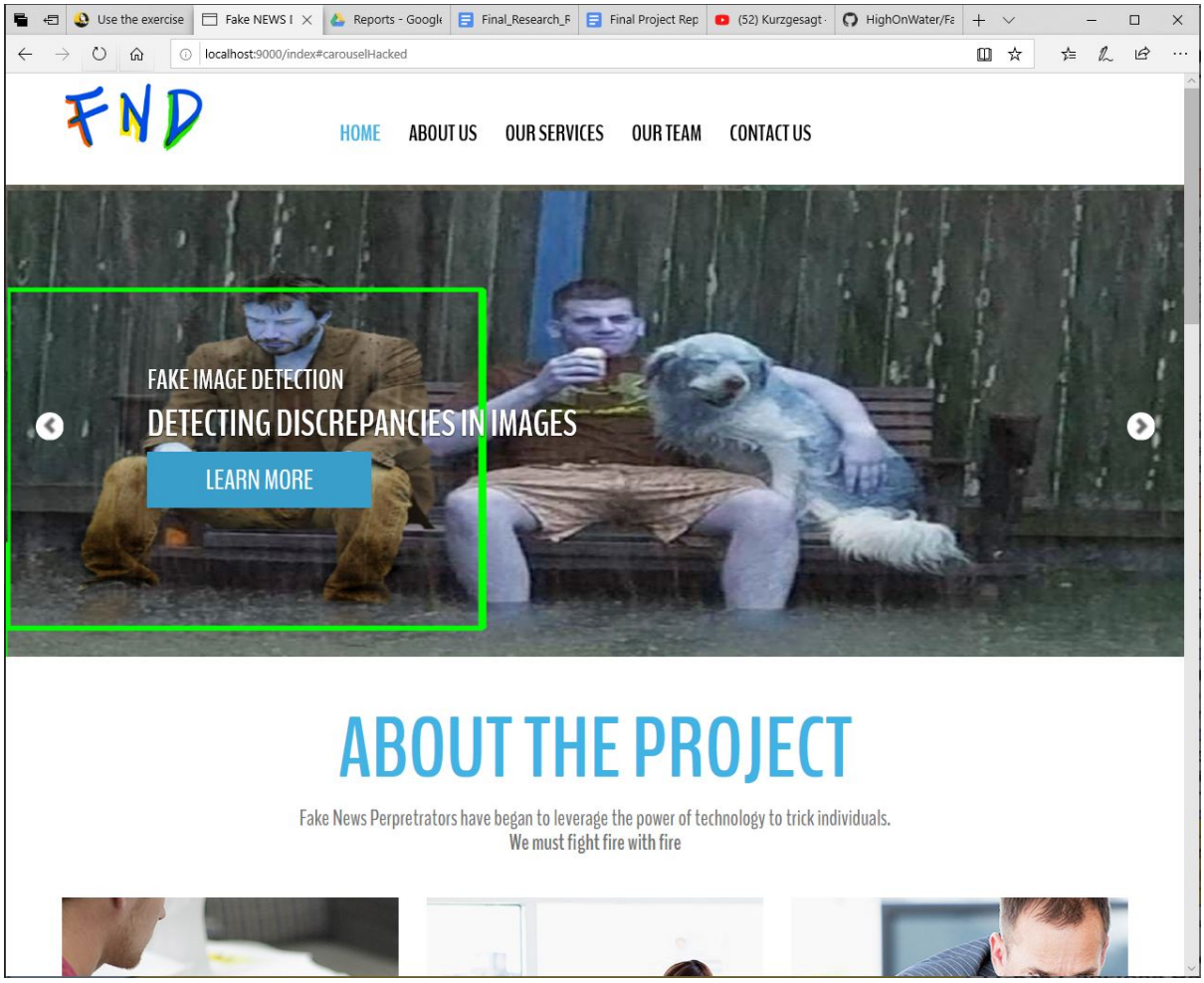

Fig. 7.7 Fake Image Detection Result

The fake image detection model looks for inconsistencies in an images based of the EXIF predicted metadata values for subregions in an Image.

Timing: 1 - 5 mins per image based on the image size and the GPU used.

Performance:

Dataset	mAP
Carvalho	0.75
Columbia	0.97

Table 7.5 Fake Image Detection Model Performance

7.6 Stance Detection

Performance of the stance detection model on the kaggle fake news detection dataset.

The dataset contains articles and headlines for the article along with their labels indicating whether the headline agrees with the article or not.

Agreement	1,872
Disagreement	2,137

Table 7.6 Stance Detection Dataset Details

Model	Accuracy	Run Time
ELMo	96%	> 10 ms
LSTM + Word2Vec	94%	> 1ms

Table 7.7 Stance Detection ML Results

Graphs for ELMO Model

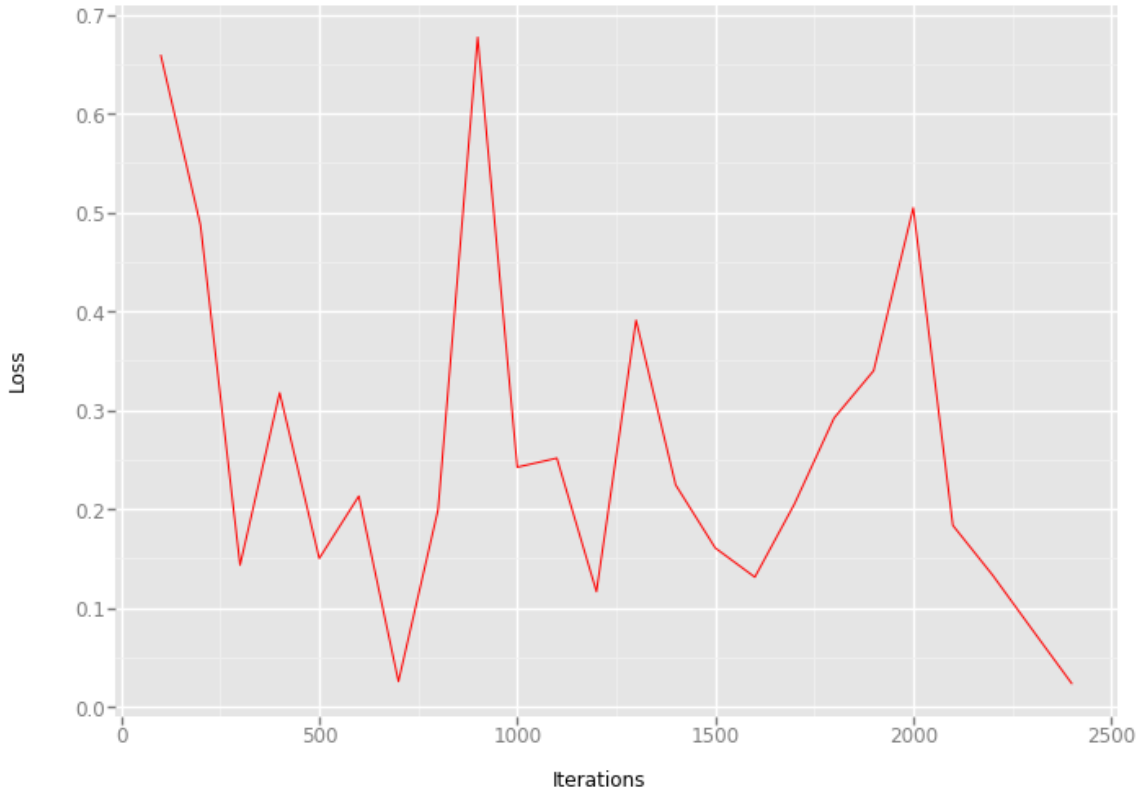

Fig. 7.8 Stance Detection - Loss vs Iterations Graph

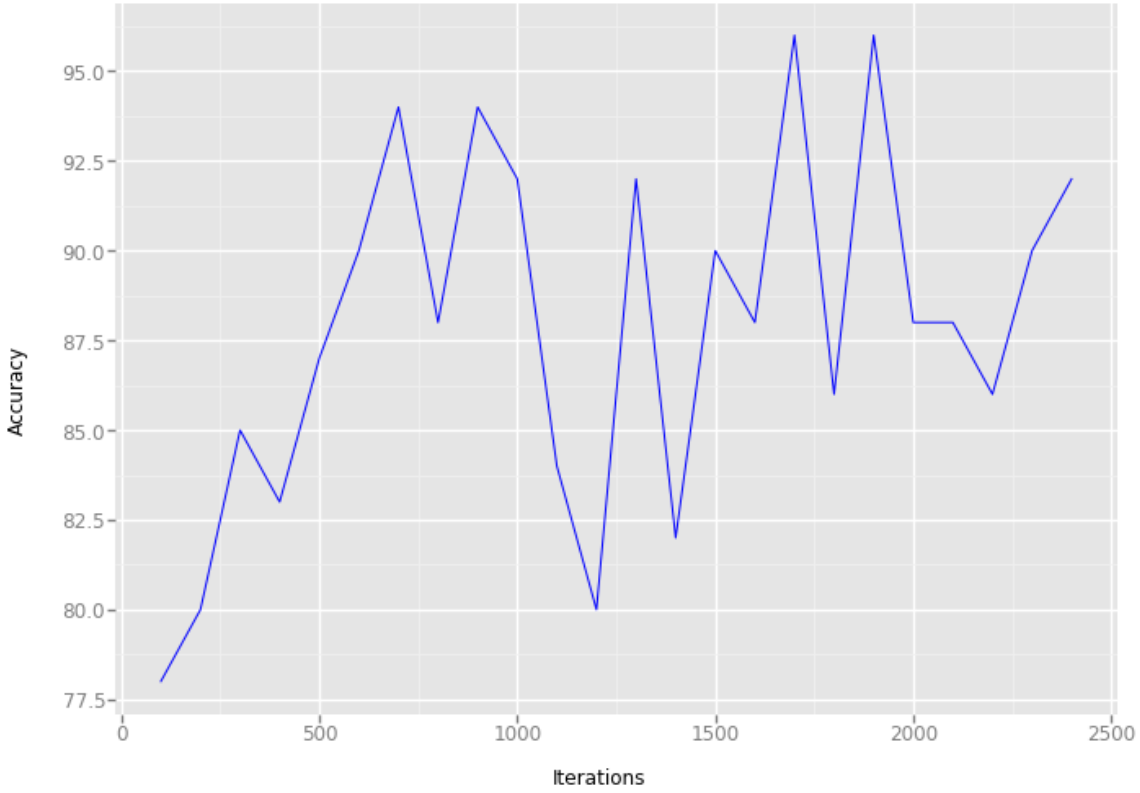

Fig. 7.9 Stance Detection - Accuracy vs Iterations Graph

CHAPTER 8

CONCLUSION

The project introduces a new and comprehensive approach to tackling the problem of fake news. It builds on a strong foundation which is backed by extensive research. This report investigates fake news right from its inception to the reasons for its rapid spread and the psychology of its perpetrators and victims. Furthermore, the project illustrates some important examples and discusses the consequences of fake news in depth. After building a solid background the project lists certain metrics that can be used to scrutinize a news article and make a judgement on its reliability.

The metrics are chosen carefully and reasons for their inclusion are explained in great detail in this report. Finally, the results of each metric are displayed concisely on a web platform which a user can look at and make his own judgements on the reliability of the news. An API access to all the developed metrics is also provided to make our work accessible for use by 3rd party application developers or individuals who seek access to our work for research purposes. The goal of our project was to provide conclusive, insight based results and let user decide whether to trust a piece of news content based on the same.

In our research project, we first analyzed the challenges involved in detecting fake news, through an extensive literature survey. We then came up with 6 metrics that are relevant for the task of providing the reader with quantitative information regarding fake news and implemented them using the Python programming language. Our aim was to make our research work accessible to all and hence to demonstrate them we developed a dashboard based, RESTful API and a single / unified web platform that takes input from user in various formats and evaluates the given input against the metrics that we defined:

1. Stance Detection
2. Fake Image Detection
3. Community Detection
4. Credible Resources Check
5. Fake Website Detection
6. Fake Account Detection

The metric for identifying disagreement between an article's headline and body was implemented using an LSTM recurrent neural network, which was also modified to compare two article bodies. Although there are no formal datasets that are directly relevant to this task, we have manually tested this model by comparing articles that we know are fake to articles collected from the web. The dissimilarity score was another metric used to identify discrepancies in the stance. The Jaccard distance was used for this and provided accurate results.

The metric for evaluating the veracity of a website was built using the XGBoost algorithm, which focused on the features of a website URL, with an accuracy of 91.35%, the highest among other classifiers viz. Random Forest Classifier, Decision Trees, SVM. A sub-part of this metric was Web Spam Detection which evaluated web content based on metrics that represented the characteristics of spam-based websites. Finally, we incorporated the work of Jeeva et al. [17], for fake website detection with our XGBoost algorithm that provides a stricter version for evaluating URL, increasing the accuracy.

The fake image detection metric is able to detect spliced regions in an image with good precision. Furthemore, the algorithm also displays a bounding box around the spliced regions making it easier for the users to spot the anomaly. One downside of this algorithm is its running time. Generally takes over 1 minute to process a single image.

Community detection metric is able to successfully identify accounts on Twitter which share commonalities and group them appropriately. This helps in recognizing the communities that might be responsible for spreading fake news.

Fake Account detection analyzes the flow of a tweet on Twitter and looks at each account individually to determine the probability of the news being fake. The algorithm is accurately able to determine spam and fake fake accounts on Twitter.

CHAPTER 9

FUTURE WORK

1. The metric Phishing URL detection defined in section 6.1.6 can be improved by experimenting with ensemble methods for improving accuracies. New URL features and dimensionality reduction techniques could be experimented with, for increasing accuracy. Also Link based techniques can be used in Web Spam Detection.

2. The fake image detection model needs to be made more efficient. The run time for this metric needs to be reduced to under 10 seconds if it has to be used for commercial applications. A possible solution to this problem is to cache the intermediate outputs.

3. The model for the Stance detection metric was trained on a relatively small dataset. The dataset had only political articles which makes the model overfit to this task. Hence to improve the accuracy of stance detection a more comprehensive dataset must be used for training it.

4. The Community detection metric currently uses only Twitter descriptions to generate a 100 dimensional vector which is then used for clustering the accounts into communities. This can improved by using the contents of accounts past tweets as well.

CHAPTER 10

BIBLIOGRAPHY

[1] Zhou, Xinyi, and Reza Zafarani. "Fake News: A Survey of Research, Detection Methods, and Opportunities." arXiv preprint arXiv:1812.00315(2018).

[2] Riedel, B., Augenstein, I., Spithourakis, G.P. and Riedel, S., 2017. A simple but tough-to-beat baseline for the Fake News Challenge stance detection task. arXiv preprint arXiv: 1707.03264.

[3] Ruder, S., Glover, J., Mehrabani, A. and Ghaffari, P., 2018, June. 360 stance detection. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations.

[4] Ajao, O., Bhowmik, D. and Zargari, S., 2018, July. Fake news identification on twitter with hybrid cnn and rnn models. In Proceedings of the 9th International Conference on Social Media and Society ACM.

[5] Huh, M., Liu, A., Owens, A. and Efros, A.A., 2018. Fighting fake news: Image splice detection via learned self-consistency. In Proceedings of the European Conference on Computer Vision (ECCV)

[6] Zhang, J., Cui, L., Fu, Y. and Gouza, F.B., 2018. Fake news detection with deep diffusive network model. arXiv preprint arXiv:1805.08751.

[7] Peters, M.E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K. and Zettlemoyer, L., 2018. Deep contextualized word representations. arXiv preprint arXiv:1802.05365.

[8] Liu, Y. and Wu, Y.F.B., 2018, April. Early detection of fake news on social media through propagation path classification with recurrent and convolutional networks. In Thirty-Second AAAI Conference on Artificial Intelligence.

[9] Ranjan, V., Kwon, H., Balasubramanian, N. and Hoai, M., 2018. Fake Sentence Detection as a Training Task for Sentence Encoding. arXiv preprint arXiv:1808.03840.

[10] Gragnaniello, D., Marra, F., Poggi, G. and Verdoliva, L., 2018, September. Analysis of adversarial attacks against CNN-based image forgery detectors. In 2018 26th European Signal Processing Conference (EUSIPCO) IEEE.

[11] Tosik, M., Mallia, A. and Gangopadhyay, K., 2018. Debunking Fake News One Feature at a Time. arXiv preprint arXiv:1808.02831.

[12] Shu, K., Mahudeswaran, D., Wang, S., Lee, D. and Liu, H., 2018. Fakenewsnet: A data repository with news content, social context and dynamic information for studying fake news on social media. arXiv preprint arXiv:1809.01286.

[13] Choy, M. and Chong, M., 2018. Seeing Through Misinformation: A Framework for Identifying Fake Online News. arXiv preprint arXiv:1804.03508.

[14] Vicario, M.D., Quattrociocchi, W., Scala, A. and Zollo, F., 2019. Polarization and fake news: Early warning of potential misinformation targets. ACM Transactions on the Web (TWEB).

[15] Von Luxburg, U., 2007. A tutorial on spectral clustering. Statistics and computing.

[16] Havemann, Frank, Jochen Gläser, and Michael Heinz. "Communities as Well Separated Subgraphs With Cohesive Cores: Identification of Core-Periphery Structures in Link Communities." In International Workshop on Complex Networks and their Applications, pp. 219-230. Springer, Cham, 2018.

[17] Jeeva, S.C. and Rajsingh, E.B., 2016. Intelligent phishing url detection using association rule mining. Human-centric Computing and Information Sciences

CHAPTER 12

APPENDICES

1. Social Media: Interactive computer-mediated technologies that facilitate the creation and sharing of information, ideas, career interests and other forms of expression via virtual communities and networks. The variety of stand-alone and built-in social media services currently available introduces challenges of definition; however, there are some common features:

- Social media are interactive Web 2.0 Internet-based applications.
- User-generated content, such as text posts or comments, digital photos or videos, and data generated through all online interactions, is the lifeblood of social media.
- Users create service-specific profiles for the website or app that are designed and maintained by the social media organization.
- Social media facilitate the development of online social networks by connecting a user's profile with those of other individuals or groups.

Users usually access social media services via web-based technologies on desktops and laptops, or download services that offer social media functionality to their mobile devices (e.g., smartphones and tablets). As users engage with these electronic services, they create highly interactive platforms through which individuals, communities, and organizations can share, co-create, discuss, and modify user-generated content or pre-made content posted online.

2. Fake news or junk news or pseudo-news is a type of yellow journalism or propaganda that consists of deliberate disinformation or hoaxes spread via traditional print and broadcast news media or online social media. The false information is often caused by reporters paying sources for stories, an unethical practice called checkbook journalism. Digital news has brought back and increased the usage of fake news, or yellow journalism. The news is then often reverberated as misinformation in social media but occasionally finds its way to the mainstream media as well.Fake news is written and published usually with the intent to mislead in order to damage an agency, entity, or person, and/or gain financially or politically,

often using sensationalist, dishonest, or outright fabricated headlines to increase readership. Similarly, clickbait stories and headlines earn advertising revenue from this activity.

3. Machine learning is an application of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed. Machine learning focuses on the development of computer programs that can access data and use it learn for themselves.

4. Deep learning (also known as deep structured learning or hierarchical learning) is part of a broader family of machine learning methods based on the layers used in artificial neural networks. Learning can be supervised, semi-supervised or unsupervised.

5. Diffusion is the movement of a substance from an area of high concentration to an area of low concentration. Diffusion happens in liquids and gases because their particles move randomly from place to place. Diffusion is an important process for living things; it is how substances move in and out of cells. The propagation of fake news can be captured using the process of diffusion

6. tf–idf or TFIDF, short for term frequency–inverse document frequency, is a numerical statistic that is intended to reflect how important a word is to a document in a collection or corpus. It is often used as a weighting factor in searches of information retrieval, text mining, and user modeling. The tf–idf value increases proportionally to the number of times a word appears in the document and is offset by the number of documents in the corpus that contain the word, which helps to adjust for the fact that some words appear more frequently in general. Tf–idf is one of the most popular term-weighting schemes today; 83% of text-based recommender systems in digital libraries use tf–idf.

7. XGBoost is an open-source software library which provides a gradient boosting framework for C++, Java, Python,R, and Julia.It works on Linux, Windows, and macOS. From the project description, it aims to provide a "Scalable, Portable and Distributed Gradient Boosting (GBM, GBRT, GBDT) Library". Other than running on a single machine, it also

supports the distributed processing frameworks Apache Hadoop, Apache Spark, and Apache Flink. It has gained much popularity and attention recently as the algorithm of choice for many winning teams of machine learning competitions.

8. ELMo is a deep contextualized word representation that models both (1) complex characteristics of word use (e.g., syntax and semantics), and (2) how these uses vary across linguistic contexts (i.e., to model polysemy). These word vectors are learned functions of the internal states of a deep bidirectional language model (biLM), which is pre-trained on a large text corpus. They can be easily added to existing models and significantly improve the state of the art across a broad range of challenging NLP problems, including question answering, textual entailment and sentiment analysis.

9. In the field of machine learning and specifically the problem of statistical classification, a confusion matrix, also known as an error matrix, is a specific table layout that allows visualization of the performance of an algorithm, typically a supervised learning one (in unsupervised learning it is usually called a matching matrix). Each row of the matrix represents the instances in a predicted class while each column represents the instances in an actual class (or vice versa).

10. A data set (or dataset) is a collection of data. Most commonly a data set corresponds to the contents of a single database table, or a single statistical data matrix, where every column of the table represents a particular variable, and each row corresponds to a given member of the data set in question. The data set lists values for each of the variables, such as height and weight of an object, for each member of the data set. Each value is known as a datum. The data set may comprise data for one or more members, corresponding to the number of rows.

11. Keras is an open-source neural-network library written in Python. It is capable of running on top of TensorFlow, Microsoft Cognitive Toolkit, Theano, or PlaidML. Designed to enable fast experimentation with deep neural networks, it focuses on being user-friendly, modular, and extensible. It was developed as part of the research effort of project ONEIROS (Open-ended Neuro-Electronic Intelligent Robot Operating System), and its primary author and

maintainer is François Chollet, a Google engineer. Chollet also is the author of the XCeption deep neural network model.

12. Representational State Transfer (REST) is a software architectural style that defines a set of constraints to be used for creating Web services. Web services that conform to the REST architectural style, termed RESTful Web services (RWS), provide interoperability between computer systems on the Internet. RESTful Web services allow the requesting systems to access and manipulate textual representations of Web resources by using a uniform and predefined set of stateless operations. Other kinds of Web services, such as SOAP Web services, expose their own arbitrary sets of operations.

13. In computer programming, an application programming interface (API) is a set of subroutine definitions, communication protocols, and tools for building software. In general terms, it is a set of clearly defined methods of communication among various components. A good API makes it easier to develop a computer program by providing all the building blocks, which are then put together by the programmer. An API may be for a web-based system, operating system, database system, computer hardware, or software library.An API specification can take many forms, but often includes specifications for routines, data structures, object classes, variables, or remote calls. POSIX, Windows API and ASPI are examples of different forms of APIs. Documentation for the API usually is provided to facilitate usage and implementation.