

# **FAKE NEWS ANALYSIS USING MACHINE LEARNING**

Submitted in partial fulfillment of the  
requirements for the award of  
Bachelor of Engineering degree in Computer Science and Engineering

by

Shubham Rai (Reg.No.37110711)  
Krish Kumar Hissaria (Reg.No.37110382)



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING  
SCHOOL OF COMPUTING**

## **SATHYABAMA**

**INSTITUTE OF SCIENCE AND TECHNOLOGY  
(DEEMED TO BE UNIVERSITY)  
Accredited with Grade “A” by NAAC**

**JEPPIAAR NAGAR, RAJIV GANDHI SALAI,  
CHENNAI - 600 119**

**APRIL – 2021**



# **SATHYABAMA**

INSTITUTE OF SCIENCE AND TECHNOLOGY  
(DEEMED TO BE UNIVERSITY)

Accredited with “A” grade by NAAC  
Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai – 600 119  
[www.sathyabama.ac.in](http://www.sathyabama.ac.in)



## **DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

### **BONAFIDE CERTIFICATE**

This is to certify that this Project Report is the bonafide work of **Shubham Rai (Reg.No.37110711)** and **Krish Kumar Hissaria (Reg.no.37110382)** who carried out the project entitled “**Fake News Analysis using Machine Learning**” under our supervision from JANUARY 2021 to JUNE 2021.

Internal Guide

**Dr.S Revathy,M.E.,Ph.D.**

Head of the Department

**Dr. S. Vigneswari, M.E., Ph.D.,**

---

Submitted for Viva voce Examination held on \_\_\_\_\_

**Internal Examiner**

**External Examiner**

## **DECLARATION**

I Shubham Rai (Reg.No.37110711) hereby declare that the Project Report entitled “**Fake News Analysis using Machine Learning**” done by me under the guidance of **Dr. S Revathy, M.E.,Ph.D.** is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Computer Science and Engineering.

**DATE:**

**PLACE: CHENNAI**

**SIGNATURE OF THE CANDIDATE**

## ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management of Sathyabama Institute of Science And Technology** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T.Sasikala M.E.,Ph.D., Dean, School of Computing** **Dr.S.Vigneshwari M.E., Ph.D.** and **Dr.L.Lakshmanan M.E., Ph.D. ,** Heads of the Department of Computer Science and Engineering for providing us necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and a deep sense of gratitude to my Project Guide **Dr. S. Revathy M.E., Ph.D.,** for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

## **ABSTRACT**

Fake news has been an issue since the time the web blast. The very network that permits us to realize what's going on worldwide is the ideal favourable place for malignant and fake news. Fighting this fake news is significant on the grounds that the perspective is moulded by data. Individuals settle on significant choices dependent on data as well as structure their own suppositions. On the off chance that this data is bogus, it can have decimating outcomes. Confirming every news individually by a person is totally impractical. This project endeavours to facilitate the cycle of recognizable proof of fake news by proposing a framework that can dependably order fake news. Machine Learning calculations, for example, Naive Bayes, Passive Aggressive Classifier, and Deep Neural Networks have been utilized on eight distinctive datasets obtained from different sources. The project additionally incorporates the investigation and aftereffects of each model. The difficult errand of detection of fake news can be made minor with the utilization of the correct models with the correct apparatuses. The aftereffects of this undertaking exhibit the capacity for machine learning to be valuable in this assignment. We have constructed a model that gets numerous natural signs of genuine and fake news just as an application that guides in the representation of the arrangement choice.

# TABLE OF CONTENTS

| CHAPTER NO.                               | TITLE                  | PAGE No |
|-------------------------------------------|------------------------|---------|
|                                           | <b>ABSTRACT</b>        |         |
|                                           | <b>LIST OF FIGURES</b> |         |
| <b>1. INTRODUCTION</b>                    |                        |         |
| Introduction                              | 1                      |         |
| Data Analysis                             | 3                      |         |
| <b>2. AIM AND SCOPE</b>                   |                        |         |
| 2.1 Aim of the Project                    |                        | 4       |
| 2.2 Scope                                 |                        | 4       |
| <b>3. SYSTEM DESIGN &amp; METHODOLOGY</b> |                        |         |
| 3.1 Existing System                       |                        | 5       |
| 3.2 Proposed System                       |                        | 5       |
| 3.3 Software Development Methodology      |                        | 5       |
| 3.4 Requirement Specification             |                        | 7       |
| 3.4.1 Software Requirements               |                        | 7       |
| 3.4.1.1 CSS                               |                        | 7       |
| 3.4.1.2 HTML                              |                        | 9       |
| 3.4.1.3 Java Script                       |                        | 10      |
| 3.4.1.4 Bootstrap                         |                        | 12      |
| 3.4.1.5 Django                            |                        | 12      |
| 3.4.1.6 Sklearn                           |                        | 13      |
| 3.4.1.7 Tensorflow                        |                        | 14      |
| 3.4.1.8 Web Scrapping                     |                        | 15      |
| 3.4.2 Hardware Requirements               |                        | 16      |

|                                           |    |
|-------------------------------------------|----|
| 3.5 Modules                               | 16 |
| 3.5.1 Admin Module                        | 17 |
| 3.5.2 User Module                         | 17 |
| 3.5.3 Post Module                         | 18 |
| Algorithms Used                           | 18 |
| Logistic Regression                       | 18 |
| 3.6.2 Linear SVM                          | 20 |
| 3.6.3 Convolution Neural Network          | 21 |
| 3.6.4 Bidirectional LSTM Networks         | 22 |
| 3.6.5 Ensemble Learners                   | 23 |
| <br>                                      |    |
| <b>4. SOTWARE DEVELOPMENT METHODOLOGY</b> |    |
| <br>                                      |    |
| 4.1 Architectural Design                  | 27 |
| <br>                                      |    |
| <b>5. RESULTS AND DISCUSSIONS</b>         |    |
| <br>                                      |    |
| <b>6. CONCLUSION</b>                      |    |
| Conclusion                                | 31 |
| <br>                                      |    |
| <b>REFERENCES</b>                         | 32 |
| <br>                                      |    |
| <b>APPENDIX</b>                           |    |
| <br>                                      |    |
| <b>A. SOURCE CODE</b>                     | 34 |
| <br>                                      |    |
| <b>B. JOURNAL PAPER</b>                   | 37 |

## LIST OF FIGURES

| FIGURE<br>NUMBER | TITLE                                                                    | PAGE<br>NUMBER |
|------------------|--------------------------------------------------------------------------|----------------|
| 3.1              | Waterfall Methodology                                                    | 06             |
| 3.2              | Iterative Methodology                                                    | 06             |
| 3.3              | MVT Diagram                                                              | 13             |
| 4.1              | Workflow for training algorithms and<br>classification of news articles. | 27             |
| 5.1              | Post the Link of the webpage to check                                    | 30             |
| 5.2              | Displayed Result                                                         | 30             |

# **CHAPTER 1**

## **INTRODUCTION**

### **INTRODUCTION**

With the rise of the internet, and the fast adoption of social media platforms, the dissemination of data has paved the way for the history of humankind. In addition to others, news agencies have benefitted from the extensive use of social media platforms by supplying their customers with updated news in almost real time. From newspapers, tab lists and journals, news media moved into a digital form such as online news websites, social media news, etc.[1]. Consumers have been made more easily accessible to the latest stories. 70 percent of news website traffic is represented in Facebook referrals[2]. In their current state, social media platforms are highly powerful and helpful to enable people to debate and discuss topics like democracy, education and health. However, some companies often use the platforms with a negative viewpoint to generate monetary income[3, 4] and, in other cases, to generate partial views, manipulate thoughts and spread satire or absurdity. False news is usually referred to as phenomenon.

In the last decade, the spread of false news has been growing rapidly, most noticed during the US elections in 2016[5]. The proliferation of on-line articles not in line with reality has created many issues not only in politics, but also in different fields such as sport, health and science [3]. The financial markets [6] are among those affected by false news, with rumours having catastrophic effects and stopping the market.

Our decision-making capacity depends primarily on the kind of data we use; our worldview is based on data we digest. Consumers have been increasingly reacting nonsensically to news which subsequently proved false [7, 8]. A recent example is the spread of new corona viruses, in which false reports about the advent, nature and behaviour of the virus have been distributed over the internet[9]. The situation deteriorated as more people read online about false content. It is a difficult task to identify such news online.

Fortunately, a range of computing methods will, in terms of text content, be used to identify

some articles as fake[10]. A number of repositories maintained by academics containing lists of

websites which are recognised as ambiguous and fake[11] are used by the majority of these methods on websites that are fact-checked. However, the issue is that human knowledge is necessary to classify articles as being counterfeits. More importantly, websites for checking contain articles from specific fields such as politics and are not widely used for identifying false news articles from various fields such as entertainment, sports and technology.

Data are available in different formats such as emails, videos and audios on the World Wide Web. It is comparatively difficult to identify and classify online news published in unstructures (such as news, articles, videos and audio), as this purely calls for human know-how. Calculation tools such as Natural Language Practice (NLP) should, also, be used to find anomalies that separate a dishonest, fact-based text article[12] from articles. Analyzes of fake news in comparison to real news[13] include other tools. In particular, the approach analyses how a false news storey differs from a true article on a network. The answer that an article receives can be theoretically differentiated in order to identify the article as real or false. A more hybrid approach may also be used to assess an article's social reaction as well as investigate the textual characteristics to check whether or not an article is disappointing.

A number of studies have primarily focused on detection and classification of fake news on social media platforms such as Facebook and Twitter [13, 14]. At conceptual level, fake news has been classified into different types; the knowledge is then expanded to generalize machine learning (ML) models for multiple domains [10, 15, 16]. The study by Ahmed et al. [17] included extracting linguistic features such as n-grams from textual articles and training multiple ML models including K-nearest neighbor (KNN), support vector machine (SVM), logistic regression (LR), linear support vector machine (LSVM), decision tree (DT), and stochastic gradient descent (SGD), achieving the highest accuracy (92%) with SVM and logistic regression. According to the research, as the number of increased in -grams calculated for a particular article, the overall accuracy decreased. The phenomenon has been observed for learning models that are used for classification. Shu et al. [12] achieved better accuracies with different models by combining textual features with auxiliary information such as user social engagements on social media. The authors also discussed the social and psychological theories and how they can be used to detect false information online. Further, the

authors discussed different data mining algorithms for model constructions and techniques shared for

features extraction. These models are based on knowledge such as writing style, and social context such as stance and propagation.

## DATA ANALYSIS

In a simplistic approach, we developed a Machine Learning Model using various Machine Learning and Deep Learning Algorithms to get an idea of how good Machine Learning is in detecting Fake News. The results of our analysis are stated below often:

Table 1.1: Accuracy result of different algorithms

| Algorithm Implemented                        | Accuracy |
|----------------------------------------------|----------|
| One-Dimensional Convolutional Neural Network | 93.0%    |
| Logistic Regression                          | 98.0%    |
| Decision Tree                                | 95.0%    |
| Random Forest                                | 83.0%    |
| ExtraTrees                                   | 85.0%    |

## **CHAPTER 2**

### **AIM AND**

### **SCOPE**

#### **AIM OF PROJECT**

The aim of the project is the users get notified about dubious news sources using Machine Learning, Natural Language Processing and Crowdsourcing. Fake News from unchecked sources has been ever-increasing, since the inception of Social Media through various channels like Facebook, Twitter and WhatsApp. This has led to modern news and journalistic temper going into serious questionability about the credibility of the news being reported. To combat this, we have developed a Machine Learning Application built atop Django and to bring the user a News Platform which allows the tracing of Fake News.

#### **SCOPE**

To help our Machine Learning Algorithms better detect and classify Fake News against Original News, we decided on a concept of Reputability Score, which is determined by the probability of news being fake/real and the user opinion (depending on Upvote-Downvote Ratio). The Reputability Score is determined by a Stance Algorithm developed by us, which takes into account the Machine Learning Analysis, Upvote/Downvote Ratio, similar news all around the world which determines the reputability of the news source. When a User uploads a News Article, a Scraper is kick-started which scrapes related news article from all around the world while a Machine Learning Model performs analysis on the News Article and returns a probability score on what is the possibility of the news being original/fake. At last, the Upvote/Downvote Ratio is helped to determine what is the reputability score of the news which determines whether the news is fake or not.

## **CHAPTER 3**

### **SYSTEM DESIGN & METHODOLOGY**

#### **EXISTING SYSTEM**

- Existing system consists of either one or two algorithms.
- Does not display result using crowdsourcing.
- Precision of the Algorithms used for the fake news dataset were not very high as are in ensemble method.

#### **PROPOSED SYSTEM**

- The learning algorithms are trained with different hyperparameters to attain a maximum precision with an optimum balance between variance and distortion of a given dataset.
- A grid search is computationally costly to find the best parameters.
- The measure is taken nevertheless to prevent the model from overfitting and underfitting the data.
- In order to examine performances over multiple datasets, different ensemble approaches, including packaging, boosting and the voting classification, are new to this research.
- Since all features extracted by the tool are numeric values, no categorical variables need to be encoded.

#### **SOFTWARE DEVELOPMENT METHODOLOGY**

A software development methodology in software engineering is a framework that is used to structure, plan and control the process of developing an information system. A methodology defines a paradigm (i.e. approach) to be used and a precise lifecycle to be used.

In the development of this system, an Iterative and Waterfall Methodology was used to ensure systematic and sequential flow of the system stages and also considering error fixing during and after development.

## Software Development Model: Waterfall Model

- **Waterfall model**

- A classic life cycle, systematic and sequential

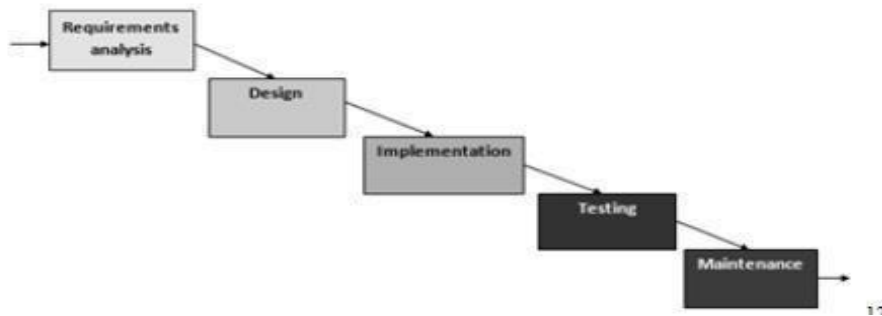


Figure 3.1 Waterfall Methodology

This methodology was chosen for its classic life cycle, systematic and sequential.

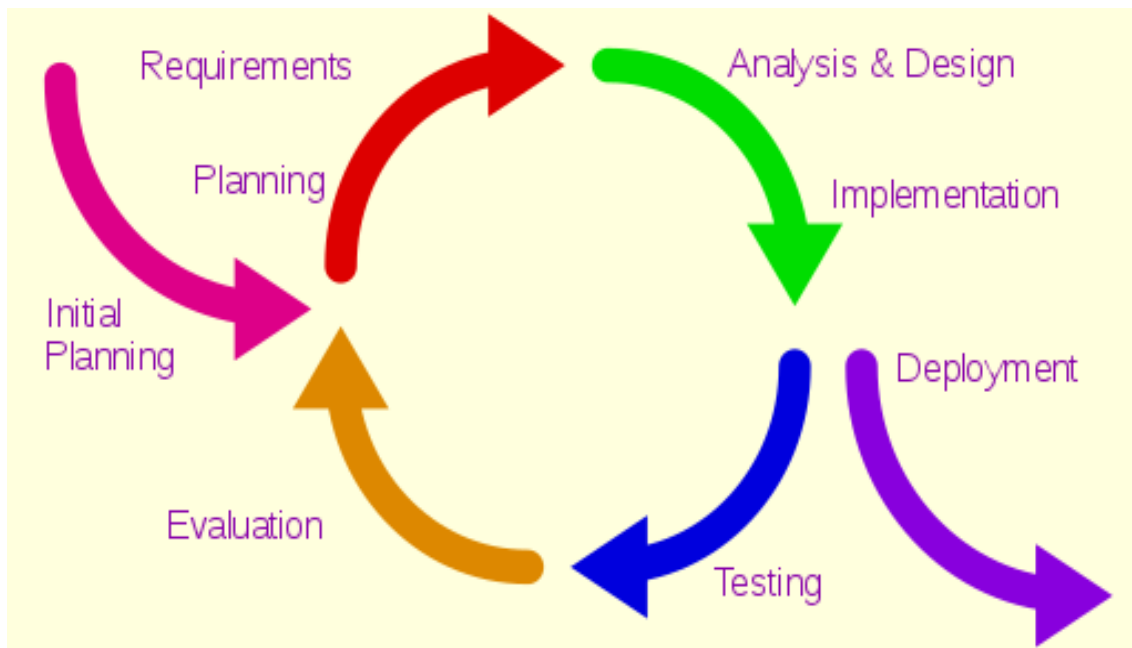


Figure 3.2 Iterative Methodology

## REQUIREMENT SPECIFICATION

This proposed software runs effectively on a computing system that has the minimum requirements. Undertaking all the equipment necessities are not satisfied but rather exist in their systems administration between the customer's machines already. So, the main need is to introduce appropriate equipment for the product.

## SOFTWARE REQUIREMENTS

1. **Django:** Django has been utilized to develop the Backend Part of our Web Application.
2. **Sklearn:** Used for implementing the Machine Learning Algorithms like Decision Tree, Random Forest and Logistic Regression.
3. **Tensorflow:** Used to implement a Single-Layered Convolutional Neural Network to perform analysis on Fake News Dataset.
4. **Chromium Extension:** Used to implement an Extension to adopt anonymous reportings from User on the state of the news and also display whether the news is dubious or not depending on the feedback and reputability score as per existing user through an API.
5. **Flutter:** For developing an Android/iOS Application to utilize our API and give out reputability score of various news articles.
6. **MySQL:** MySQL is utilized to store the news and their reputability score along with existing news sites to perform Machine Learning Analysis on a real-time basis.
7. **Front-End Frameworks:** HTML, CSS, JQuery and Bootstrap-4 has been utilized to develop a User-Interface.
8. **Web Scrappers:** Web Scrappers are utilized to scrap through the Web and give out news-articles for Machine Learning Analysis and Database storage.

## CSS

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language such as HTML. CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript.

CSS is designed to enable the separation of presentation and content, including layout, colors, and fonts. This separation can improve content accessibility, provide more flexibility and control in the

specification of presentation characteristics, enable multiple web pages to share formatting by specifying the relevant CSS in a separate .css file which reduces complexity and repetition in the structural content as well as enabling the .css file to be cached to improve the page load speed between the pages that share the file and its formatting.

Separation of formatting and content also makes it feasible to present the same markup page in different styles for different rendering methods, such as on-screen, in print, by voice (via speech-based browser or screen reader), and on Braille-based tactile devices. CSS also has rules for alternate formatting if the content is accessed on a mobile device.

The name cascading comes from the specified priority scheme to determine which style rule applies if more than one rule matches a particular element. This cascading priority scheme is predictable.

The CSS specifications are maintained by the World Wide Web Consortium (W3C). Internet media type (MIME type) text/css is registered for use with CSS by RFC 2318 (March 1998). The W3C operates a free CSS validation service for CSS documents.

In addition to HTML, other markup languages support the use of CSS including XHTML, plain XML, SVG, and XUL.

There are three ways of inserting a style sheet:

1. External CSS
2. Internal CSS
3. Inline CSS

**External CSS:** With an external style sheet, you can change the look of an entire website by changing just one file. Each HTML page must include a reference to the external style sheet file inside the <link> element, inside the head section.

**Internal CSS:** An internal style sheet may be used if one single HTML page has a unique style. The internal style is defined inside the <style> element, inside the head section.

**Inline CSS:** An inline style may be used to apply a unique style for a single element. To use

inline styles, add the style attribute to the relevant element. The style attribute can contain any CSS property.

# HTML

The HyperText Markup Language, or HTML is the standard markup language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets (CSS) and scripting languages such as JavaScript.

Web browsers receive HTML documents from a web server or from local storage and render the documents into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document.

HTML elements are the building blocks of HTML pages. With HTML constructs, images and other objects such as interactive forms may be embedded into the rendered page. HTML provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. HTML elements are delineated by tags, written using angle brackets. Tags such as `<img />` and `<input />` directly introduce content into the page. Other tags such as `<p>` surround and provide information about document text and may include other tags as sub-elements. Browsers do not display the HTML tags, but use them to interpret the content of the page.

HTML can embed programs written in a scripting language such as JavaScript, which affects the behavior and content of web pages. Inclusion of CSS defines the look and layout of content. The World Wide Web Consortium (W3C), former maintainer of the HTML and current maintainer of the CSS standards, has encouraged the use of CSS over explicit presentational HTML since 1997.

## HTML 2

November 24, 1995

HTML 2.0 was published as RFC 1866. Supplemental RFCs added capabilities:

November 25, 1995: RFC 1867 (form-based file upload)

May 1996: RFC 1942 (tables)

August 1996: RFC 1980 (client-side image maps)

January 1997: RFC 2070 (internationalization)

## **HTML 3**

January 14, 1997

HTML 3.2 was published as a W3C Recommendation. It was the first version developed and standardized exclusively by the W3C, as the IETF had closed its HTML Working Group on September 12, 1996.

Initially code-named "Wilbur", HTML 3.2 dropped math formulas entirely, reconciled overlap among various proprietary extensions and adopted most of Netscape's visual markup tags.

## **HTML 4**

December 18, 1997

HTML 4.0 was published as a W3C Recommendation. It offers three variations:

Strict, in which deprecated elements are forbidden.

Transitional, in which deprecated elements are allowed.

Frameset, in which mostly only frame related elements are allowed.

Initially code-named "Cougar", HTML 4.0 adopted many browser-specific element types and attributes, but at the same time sought to phase out Netscape's visual markup features by marking them as deprecated in favor of style sheets. HTML 4 is an SGML application conforming to ISO 8879 – SGML.

## **HTML 5**

October 28, 2014

HTML5 was published as a W3C Recommendation.

November 1, 2016

HTML 5.1 was published as a W3C Recommendation.

December 14, 2017

HTML 5.2 was published as a W3C Recommendation.

## **JAVA SCRIPT**

JavaScript often abbreviated as JS, is a programming language that conforms to the ECMAScript specification. JavaScript is high-level, often just-in-time compiled, and multi-

paradigm. It has curly-bracket syntax, dynamic typing, prototype-based object-orientation, and first-class functions.

Alongside HTML and CSS, JavaScript is one of the core technologies of the World Wide Web. Over 97% of websites use it client-side for web page behavior, often incorporating third-party libraries. All major web browsers have a dedicated JavaScript engine to execute the code on the user's device.

As a multi-paradigm language, JavaScript supports event-driven, functional, and imperative programming styles. It has application programming interfaces (APIs) for working with text, dates, regular expressions, standard data structures, and the Document Object Model (DOM).

The ECMAScript standard does not include any input/output (I/O), such as networking, storage, or graphics facilities. In practice, the web browser or other runtime system provides JavaScript APIs for I/O.

JavaScript engines were originally used only in web browsers, but they are now core components of other software systems, most notably servers and a variety of applications.

You can add JavaScript code in an HTML document by employing the dedicated HTML tag `<script>` that wraps around JavaScript code.

The `<script>` tag can be placed in the `<head>` section of your HTML, in the `<body>` section, or after the `</body>` close tag, depending on when you want the JavaScript to load. Generally, JavaScript code can go inside of the document `<head>` section in order to keep them contained and out of the main content of your HTML document.

However, if your script needs to run at a certain point within a page's layout — like when using `document.write` to generate content — you should put it at the point where it should be called, usually within the `<body>` section.

## **BOOTSTRAP**

Bootstrap is a free and open-source CSS framework directed at responsive, mobile-first front- end web development. It contains CSS- and (optionally) JavaScript-based design templates for typography, forms, buttons, navigation, and other interface components.

Bootstrap is among the most starred projects on GitHub, with more than 142,000 stars, behind freeCodeCamp (almost 312,000 stars) and marginally behind Vue.js framework.

Bootstrap is a framework to help you design websites faster and easier. It includes HTML and CSS based design templates for typography, forms, buttons, tables, navigation, modals, image carousels, etc. It also gives you support for JavaScript plugins.

- Bootstrap's responsive CSS adjusts to phones, tablets, and desktops
- Mobile-first styles are part of the framework
- Bootstrap is compatible with all modern browsers (Chrome, Firefox, Internet Explorer, Safari, and Opera).

## **DJANGO**

Django is a Python-based free and open-source web framework that follows the model-template-views (MTV) architectural pattern. It is maintained by the Django Software Foundation (DSF), an American independent organization established as a 501(c)(3) non-profit.

Django's primary goal is to ease the creation of complex, database-driven websites. The framework emphasizes reusability and "pluggability" of components, less code, low coupling, rapid development, and the principle of don't repeat yourself. Python is used throughout, even for settings, files, and data models. Django also provides an optional administrative create, read, update and delete interface that is generated dynamically through introspection and configured via admin models.

Some well known sites that use Django include PBS, Instagram, Mozilla, The Washington

Times, Disqus, Bitbucket, and Nextdoor.

Django is a high-level Python Web framework that encourages rapid development and clean, pragmatic design. Built by experienced developers, it takes care of much of the hassle of Web development, so you can focus on writing your app without needing to reinvent the wheel. It's free and open source.

### MVT Django Diagram

The MVT (Model View Template) is a software design pattern. It is a collection of three important components Model View and Template. The Model helps to handle database. It is a data access layer which handles the data.

The Template is a presentation layer which handles User Interface part completely. The View is used to execute the business logic and interact with a model to carry data and renders a template.

Although Django follows MVC pattern but maintains it's own conventions. So, control is handled by the framework itself.

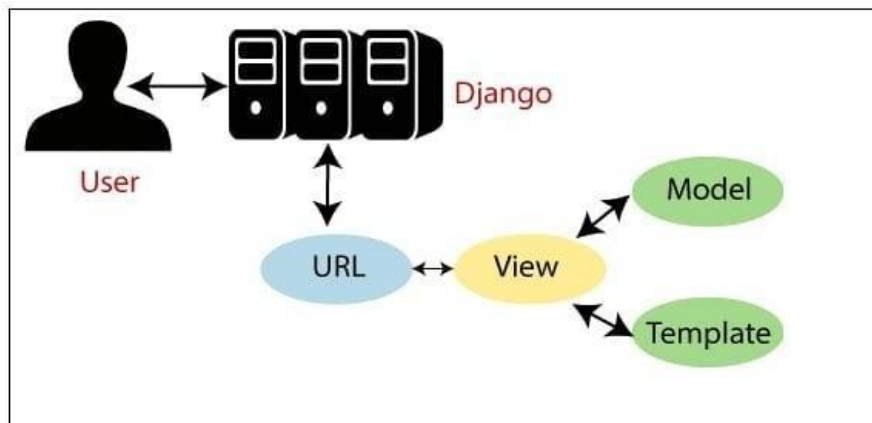


Figure 3.3: MVT Diagram

### SKLEARN

Scikit-learn (formerly scikits.learn and also known as sklearn) is a free software machine learning library for the Python programming language. It features various classification, regression and clustering algorithms including support vector machines, random forests, gradient boosting, k-

means and DBSCAN, and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy.

Scikit-learn (Sklearn) is the most useful and robust library for machine learning in Python. It provides a selection of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction via a consistent interface in Python. This library, which is largely written in Python, is built upon NumPy, SciPy and Matplotlib.

It was originally called scikits.learn and was initially developed by David Cournapeau as a Google summer of code project in 2007. Later, in 2010, Fabian Pedregosa, Gael Varoquaux, Alexandre Gramfort, and Vincent Michel, from FIRCA (French Institute for Research in Computer Science and Automation), took this project at another level and made the first public release (v0.1 beta) on 1st Feb. 2010.

## **TENSERFLOW**

TensorFlow is a free and open-source software library for machine learning. It can be used across a range of tasks but has a particular focus on training and inference of deep neural networks.

TensorFlow is a symbolic math library based on dataflow and differentiable programming. It is used for both research and production at Google.

TensorFlow was developed by the Google Brain team for internal Google use. It was released under the Apache License 2.0 in 2015.

TensorFlow is Google Brain's second-generation system. Version 1.0.0 was released on February 11, 2017. While the reference implementation runs on single devices, TensorFlow can run on multiple CPUs and GPUs (with optional CUDA and SYCL extensions for general-purpose computing on graphics processing units). TensorFlow is available on 64-bit Linux, macOS, Windows, and mobile computing platforms including Android and iOS.

Its flexible architecture allows for the easy deployment of computation across a variety of platforms (CPUs, GPUs, TPUs), and from desktops to clusters of servers to mobile and edge devices.

TensorFlow computations are expressed as stateful dataflow graphs. The name TensorFlow derives from the operations that such neural networks perform on multidimensional data arrays, which are referred to as tensors. During the Google I/O Conference in June 2016, Jeff Dean stated that 1,500 repositories on GitHub mentioned TensorFlow, of which only 5 were from Google.

In December 2017, developers from Google, Cisco, RedHat, CoreOS, and CaiCloud introduced Kubeflow at a conference. Kubeflow allows operation and deployment of TensorFlow on Kubernetes.

In March 2018, Google announced TensorFlow.js version 1.0 for machine learning in JavaScript.

In Jan 2019, Google announced TensorFlow 2.0. It became officially available in Sep 2019.

In May 2019, Google announced TensorFlow Graphics for deep learning in computer graphics.

## **WEB SCRAPING**

Web scraping, web harvesting, or web data extraction is data scraping used for extracting data from websites. The web scraping software may directly access the World Wide Web using the Hypertext Transfer Protocol or a web browser. While web scraping can be done manually by a software user, the term typically refers to automated processes implemented using a bot or web crawler. It is a form of copying in which specific data is gathered and copied from the web, typically into a central local database or spreadsheet, for later retrieval or analysis.

Web scraping a web page involves fetching it and extracting from it. Fetching is the downloading of a page (which a browser does when a user views a page). Therefore, web crawling is a main component of web scraping, to fetch pages for later processing. Once fetched, then extraction can take place. The content of a page may be parsed, searched, reformatted, its data

copied into a spreadsheet or loaded into a database. Web scrapers typically take something out of a page, to make use of it for another purpose somewhere else. An example would be to find and

copy names and telephone numbers, or companies and their URLs, or e-mail addresses to a list (contact scraping).

Web scraping is used for contact scraping, and as a component of applications used for web indexing, web mining and data mining, online price change monitoring and price comparison, product review scraping (to watch the competition), gathering real estate listings, weather data monitoring, website change detection, research, tracking online presence and reputation, web mashup and, web data integration.

Web pages are built using text-based mark-up languages (HTML and XHTML), and frequently contain a wealth of useful data in text form. However, most web pages are designed for human end-users and not for ease of automated use. As a result, specialized tools and software have been developed to facilitate the scraping of web pages.

Newer forms of web scraping involve monitoring data feeds from web servers. For example, JSON is commonly used as a transport storage mechanism between the client and the web server.

There are methods that some websites use to prevent web scraping, such as detecting and disallowing bots from crawling (viewing) their pages. In response, there are web scraping systems that rely on using techniques in DOM parsing, computer vision and natural language processing to simulate human browsing to enable gathering web page content for offline parsing.

## **HARDWARE REQUIREMENTS**

1. Hard disk of 20 GB
2. System memory (RAM) of 1 GB or higher
3. i3 processor-based computer or higher.

## **MODULES**

NewsFresh web application model consists of mainly three modules. They are Admin, User and Post module. The primary task of the admin is to maintain the database records and manage everything related to content in the system. The User module is used to save, search, delete user

related data in the database and also for to relate to different modules dependent on this one as a foreign-key relationship or many-to-many relationship or one-to-many. The post module is used to save, search, delete posts related data in the database, directly related to the User module by a Foreign key. Datapoints like post date and time, post likes and comments counts are stored and can be queried via this module. The Friends module handles user's friend lists, having many-to-many relations with the User module to maintain user-to-user connections, moreover having bunch of utilities which are used to form or delete connections between users.

There are three important modules. They are mentioned below:

## **ADMIN MODULE**

This module keeps the whole system secure and it doesn't allow any unauthorized users to access the system. The primary task of the admin is to maintain the database records and manage everything related to content in the system. Only those user accounts with superuser privileges can access this section of the system. The admin has the capabilities to use all the models to automatically build a site area that can be used to create, read, update and delete (CRUD) records, making it pretty easy to test out models and get a feel for whether the right data is been stored. For instance, in this application the admin can delete a user who is posting malicious content in the system, change a post's caption if they find it inappropriate etc., The admin panel is a default extension given by Django, the modules which needs to be maintained are to be manually registered in the admin panel by editing the admin.py file.

## **USER MODULE**

This module is used to save, search, access and delete user related data in the system and can be used to query different datapoints related to user like a particular user's post or friend list. This module has its own default authentication mechanism provided by Django, and is customizable to user's need. In this application, the User module stores user's info and settings, profile data and is related to Post module and Friends module. Datapoints like username, email, superuser permission, password hash, session key etc., are stored by this module. The session key stored by this module

can be used to access user's session data, the superuser permission is stored by the datapoint "is\_admin" which is used to distinguish normal users from admin users in the application. This

module is directly converted into User table in the database, with the datapoints acting as columns in the database, saving different data. As this module is directly connected to the database, it becomes easier to query, update and delete records from the database.

## **POST MODULES**

This module is used to save, search, delete posts related data in the database, directly related to the User module by a Foreign key. This module stores post of the link entered by the user in the input box. This module is registered to the admin panel for the admins to monitor posts being circulated across the application. This module is directly converted into User table in the database, with the datapoints acting as columns in the database, saving different data. As this module is directly connected to the database, it becomes easier to query, update and delete records from the database.

## **ALGORITHMS USED**

### **LOGISTIC REGRESSION.**

In statistics, the logistic model (or logit model) is used to model the probability of a certain class or event existing such as pass/fail, win/lose, alive/dead or healthy/sick. This can be extended to model several classes of events such as determining whether an image contains a cat, dog, lion, etc. Each object being detected in the image would be assigned a probability between 0 and 1, with a sum of one.

Logistic regression is a statistical model that in its basic form uses a logistic function to model a binary dependent variable, although many more complex extensions exist. In regression analysis, logistic regression (or logit regression) is estimating the parameters of a logistic model (a form of binary regression). Mathematically, a binary logistic model has a dependent variable with two possible values, such as pass/fail which is represented by an indicator variable, where the two values are labeled "0" and "1". In the logistic model, the log-odds (the logarithm of the odds) for the value labeled "1" is a linear combination of one or more independent variables ("predictors"); the independent variables can each be a binary variable (two classes, coded by an indicator

variable) or a continuous variable (any real value). The corresponding probability of the value labeled "1" can vary between 0 (certainly the value "0") and 1 (certainly the value "1"), hence the labeling; the function that converts log-odds to probability is the logistic function, hence the name. The unit of

measurement for the log-odds scale is called a logit, from logistic unit, hence the alternative names. Analogous models with a different sigmoid function instead of the logistic function can also be used, such as the probit model; the defining characteristic of the logistic model is that increasing one of the independent variables multiplicatively scales the odds of the given outcome at a constant rate, with each independent variable having its own parameter; for a binary dependent variable this generalizes the odds ratio.

In a binary logistic regression model, the dependent variable has two levels (categorical). Outputs with more than two values are modeled by multinomial logistic regression and, if the multiple categories are ordered, by ordinal logistic regression (for example the proportional odds ordinal logistic model). The logistic regression model itself simply models probability of output in terms of input and does not perform statistical classification (it is not a classifier), though it can be used to make a classifier, for instance by choosing a cutoff value and classifying inputs with probability greater than the cutoff as one class, below the cutoff as the other; this is a common way to make a binary classifier. The coefficients are generally not computed by a closed-form expression, unlike linear least squares; see § Model fitting. The logistic regression as a general statistical model was originally developed and popularized primarily by Joseph Berkson, beginning in Berkson (1944), where he coined "logit";

The logistic function, also called the sigmoid function was developed by statisticians to describe properties of population growth in ecology, rising quickly and maxing out at the carrying capacity of the environment. It's an S-shaped curve that can take any real-valued number and map it into a value between 0 and 1, but never exactly at those limits.

We classify text based on a wide range of features, a logistic regression (LR) model is used with binary output (true/false or true article/fake article), as it gives an intuitive equation to classify problems into binary or multiple classes[25]. We have tuned hyperparameters to get the best results for each dataset, while testing different parameters in order to achieve the highest accuracy of the LR model. The logistic hypothesis function can be defined mathematically as follows:

$$h_{\theta}(X) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X)}}.$$

Logistic regression uses a sigmoid function to convert the output into a probability, which is aimed at minimising the cost to reach an optimum likelihood. The cost function as shown is calculated

$$\text{Cost}(h_{\theta}(x), y) = \begin{cases} \log(h_{\theta}(x)), & y = 1, \\ -\log(1 - h_{\theta}(x)), & y = 0. \end{cases}$$

## LINEAR SVM

In machine learning, support-vector machines (SVMs, also support-vector networks) are supervised learning models with associated learning algorithms that analyze data for classification and regression analysis. Developed at AT&T Bell Laboratories by Vladimir Vapnik with colleagues (Boser et al., 1992, Guyon et al., 1993, Vapnik et al., 1997), SVMs are one of the most robust prediction methods, being based on statistical learning frameworks or VC theory proposed by Vapnik and Chervonenkis (1974) and Vapnik (1982, 1995). Given a set of training examples, each marked as belonging to one of two categories, an SVM training algorithm builds a model that assigns new examples to one category or the other, making it a non-probabilistic binary linear classifier (although methods such as Platt scaling exist to use SVM in a probabilistic classification setting). An SVM maps training examples to points in space so as to maximise the width of the gap between the two categories. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall.

In addition to performing linear classification, SVMs can efficiently perform a non-linear classification using what is called the kernel trick, implicitly mapping their inputs into high-dimensional feature spaces.

When data are unlabelled, supervised learning is not possible, and an unsupervised learning approach is required, which attempts to find natural clustering of the data to groups, and then map new data to these formed groups. The support-vector clustering algorithm, created by Hava Siegelmann and Vladimir Vapnik, applies the statistics of support vectors, developed in the support vector machines algorithm, to categorize unlabeled data, and is one of the most widely used clustering algorithms in industrial applications.[citation needed]

Linear SVM is used for linearly separable data, which means if a dataset can be classified into two classes by using a single straight line, then such data is termed as linearly separable data, and classifier is used called as Linear SVM classifier.

We use the linear SVM method. We trained the linear SVM on the function set with 5-fold cross validation to ensure meaningful comparison. Please note that in the text the methodology is called Perez-LSVM.

## **CONVOLUTION NEURAL NETWORK**

In deep learning, a convolutional neural network (CNN, or ConvNet) is a class of deep neural networks, most commonly applied to analyzing visual imagery. They are also known as shift invariant or space invariant artificial neural networks (SIANN), based on the shared-weight architecture of the convolution kernels that shift over input features and provide translation equivariant responses. Counter-intuitively, most convolutional neural networks are only equivariant, as opposed to invariant, to translation. They have applications in image and video recognition, recommender systems, image classification, image segmentation, medical image analysis, natural language processing, brain- computer interfaces, and financial time series.

CNNs are regularized versions of multilayer perceptrons. Multilayer perceptrons usually mean fully connected networks, that is, each neuron in one layer is connected to all neurons in the next layer. The "full connectivity" of these networks make them prone to overfitting data. Typical ways of regularization, or preventing overfitting, include: penalizing parameters during training (such as weight decay) or trimming connectivity (skipped connections, dropout, etc.) CNNs take a different approach towards regularization: they take advantage of the hierarchical pattern in data and assemble patterns of increasing complexity using smaller and simpler patterns embossed in their filters. Therefore, on a scale of connectivity and complexity, CNNs are on the lower extreme.

Convolutional networks were inspired by biological processes in that the connectivity pattern between neurons resembles the organization of the animal visual cortex. Individual cortical neurons

respond to stimuli only in a restricted region of the visual field known as the receptive field. The receptive fields of different neurons partially overlap such that they cover the entire visual field.

CNNs use relatively little pre-processing compared to other image classification algorithms. This means that the network learns to optimize the filters (or kernels) through automated learning, whereas in traditional algorithms these filters are hand-engineered. This independence from prior knowledge and human intervention in feature extraction is a major advantage.

CNN is a deep neural network originally designed for image analysis. Recently, it was discovered that the CNN also has an excellent capacity in sequent data analysis such as natural language processing (Zhang, 2015). CNN always contains two basic operations, namely convolution and pooling. The convolution operation using multiple filters is able to extract features (feature map) from the data set, through which their corresponding spatial information can be preserved. The pooling operation, also called subsampling, is used to reduce the dimensionality of feature maps from the convolution operation. Max pooling and average pooling are the most common pooling operations used in the CNN. Due to the complicity of CNN, relu is the common choice for the activation function to transfer gradient in training by backpropagation.

For automatically detecting fake news, Wang [18] used the convolutionary neural network (CNN). We used our sample to follow the same strategy. However, we were unable to use the Wang [18] feature set as the dataset includes only short statements. The Wang-CNN in the text is the tactic.

## **BIDIRECTIONAL LONG SHORT-TERM MEMORY NETWORKS**

Long short-term memory (LSTM) is an artificial recurrent neural network (RNN) architecture used in the field of deep learning. Unlike standard feedforward neural networks, LSTM has feedback connections. It can not only process single data points (such as images), but also entire sequences of data (such as speech or video). For example, LSTM is applicable to tasks such as unsegmented, connected handwriting recognition, speech recognition and anomaly detection in network traffic or IDSs (intrusion detection systems).

A common LSTM unit is composed of a cell, an input gate, an output gate and a forget gate. The cell remembers values over arbitrary time intervals and the three gates regulate the flow of

information into and out of the cell.

LSTM networks are well-suited to classifying, processing and making predictions based on time series data, since there can be lags of unknown duration between important events in a time series. LSTMs were developed to deal with the vanishing gradient problem that can be encountered when training traditional RNNs. Relative insensitivity to gap length is an advantage of LSTM over RNNs, hidden Markov models and other sequence learning methods in numerous applications.

An algorithm called bidirectional long short-term memory networks (BLSTM) for processing sequential data is introduced. This supervised learning method trains a special recurrent neural network to use very long-range symmetric sequence context using a combination of nonlinear processing elements and linear feedback loops for storing long-range context. The algorithm is applied to the sequence-based prediction of protein localization and predicts 93.3 percent novel non plant proteins and 88.4 percent novel plant proteins correctly, which is an improvement over feedforward and standard recurrent networks solving the same problem.

## **ENSEMBLE LEARNERS**

In statistics and machine learning, ensemble methods use multiple learning algorithms to obtain better predictive performance than could be obtained from any of the constituent learning algorithms alone. Unlike a statistical ensemble in statistical mechanics, which is usually infinite, a machine learning ensemble consists of only a concrete finite set of alternative models, but typically allows for much more flexible structure to exist among those alternatives.

An ensemble is itself a supervised learning algorithm, because it can be trained and then used to make predictions. The trained ensemble, therefore, represents a single hypothesis. This hypothesis, however, is not necessarily contained within the hypothesis space of the models from which it is built. Thus, ensembles can be shown to have more flexibility in the functions they can represent. This flexibility can, in theory, enable them to over-fit the training data more than a single model would, but in practice, some ensemble techniques (especially bagging) tend to reduce problems related to over-fitting of the training data.

Empirically, ensembles tend to yield better results when there is a significant diversity

among the models. Many ensemble methods, therefore, seek to promote diversity among the models they combine. Although perhaps non-intuitive, more random algorithms (like random decision trees) can

be used to produce a stronger ensemble than very deliberate algorithms (like entropy-reducing decision trees). Using a variety of strong learning algorithms, however, has been shown to be more effective than using techniques that attempt to dumb-down the models in order to promote diversity.

In order to increase the overall accuracy of the classification between a true article and a false one, we suggested that existing assemble procedures along with textual features be used. Together students tend to have better accuracy, as all of the models are trained to minimise the overall error rate with a specific approach and increase model performance. The intuition behind ensemble modelling is identical with the one already used to in our everyday lives such as seeking the views of different experts in order to mitigate an improper choice or unwanted outcomes before taking a particular decision. For example, a grade algorithm with a special set of parameters can be trained on a specific dataset which can generate a decision limit that is in some extent suitable for the data. Not only the parameters used to train the model, but also the type of training data is the outcome of this particular algorithm. If less variance or uniform data are found in the training data, the model will overflow and generate skewed results over unseen information. To mitigate the risk of overfitting, methods such as cross validation are used. A variety of models can be trained on various parameters in order to establish multiple decision borders on randomly selected data points. Therefore these problems can be solved and mitigated by training different algorithms using ensemble learning methods, and their results can also be combined to achieve almost optimal results. This approach involves using voting classifiers, which rely on the biggest votes of all algorithms for the final classification.

However, in different situations such as the following, there are also other ensemble methods.

1. Random Forest (RF): The random forest (RF) is a state-of-the-art decision-making (DT) model that is now a controlled learning model. RF includes a considerable number of decision- making treasures that work independently to predict an outcome in a class in which the ultimate forecast is based on a majority voting class. Random forest error is low in comparison to other models because of low tree[27] correlation. In our random forest model, the parameters used to generate the best model that can predict the result with high precision

were trained, i.e. in a grid search different number of estimators. are many algorithms to decide on the regression or classification problem in a decision tree. We used the Gini index as a cost

function to quantify a fraction of the data set for the classification problem. The index Gini is computed by subtracting from one the total of each class' squared probabilities. The Gini Index ( $G_{ind}$ ) mathematical formula is as follows:

$$G_{ind} = 1 - \sum_{i=1}^c (P_i)^2,$$

2. **Bagging Ensemble Classifiers:** Bootstraps are an early ensemble technique used primarily to minimise the variation (overfitting) over a training set in a short bagging classification. One of the most widely used random forest model as a baggage classification variant. Intuitively, the bagging model chooses the class for the classification problem based on the main votes estimated by the M number of trees to decrease overall variation while random sampling of each tree with substitution from the total dataset would be chosen. However, the bagging model averages many estimates for regression problems.
3. **Boosting Ensemble Classifiers:** Boosting is an additional technique commonly used for training weak models to become strong students. A forest of random trees is created for this purpose, and the final forecast is based on the results from the majority voting in each tree. Method encourages poor students in a commonly misclassified approach to classify data points correctly. Initially the same weighted coefficients are used to classify a given problem for all data points. In the following rounds, the weighted coefficients are lowered for properly classified data points and improved for misclassified data points[29]. Each subsequent tree created each round reduces the errors of the previous round by properly classifying data points misclassified in the previous rounds to improve the overall precision. One of the the problems with boosting ensemble is that training data may overflow, leading to inaccurate prognoses for unseen cases [30]. Available are some boosting algorithms that can be used for both classification and regression purposes. In our trials we used classification algorithms XGBoost[31] and AdaBoost[32].
4. **Voting Ensemble Classifiers:** Voting ensemble is usually used for classification problems since this enables two or more learning models to be combined with the training of the entire dataset[33]. Every model forecasts the result of a sample data point which is regarded as a "vote" for the class expected by the model. As discussed, package algorithms construct a variety of data sub-sets by randomly sampling the data set and replacing it with the entire

data set. The final outcomes of each model are aggregated by each dataset, since each model is

used to create a model. When boosted, several models are trained sequentially, whereby each model learns from the previous models by increasing weights for the wrong stitches to create a generic model that can classify the problem correctly. On the other hand, however, voting ensemble is a combination of several independent models that generates classification results which contribute by majority vote to the overall prediction.

## CHAPTER 4

### SOFTWARE DEVELOPMENT METHODOLOGY

#### ARCHITECTURAL DESIGN

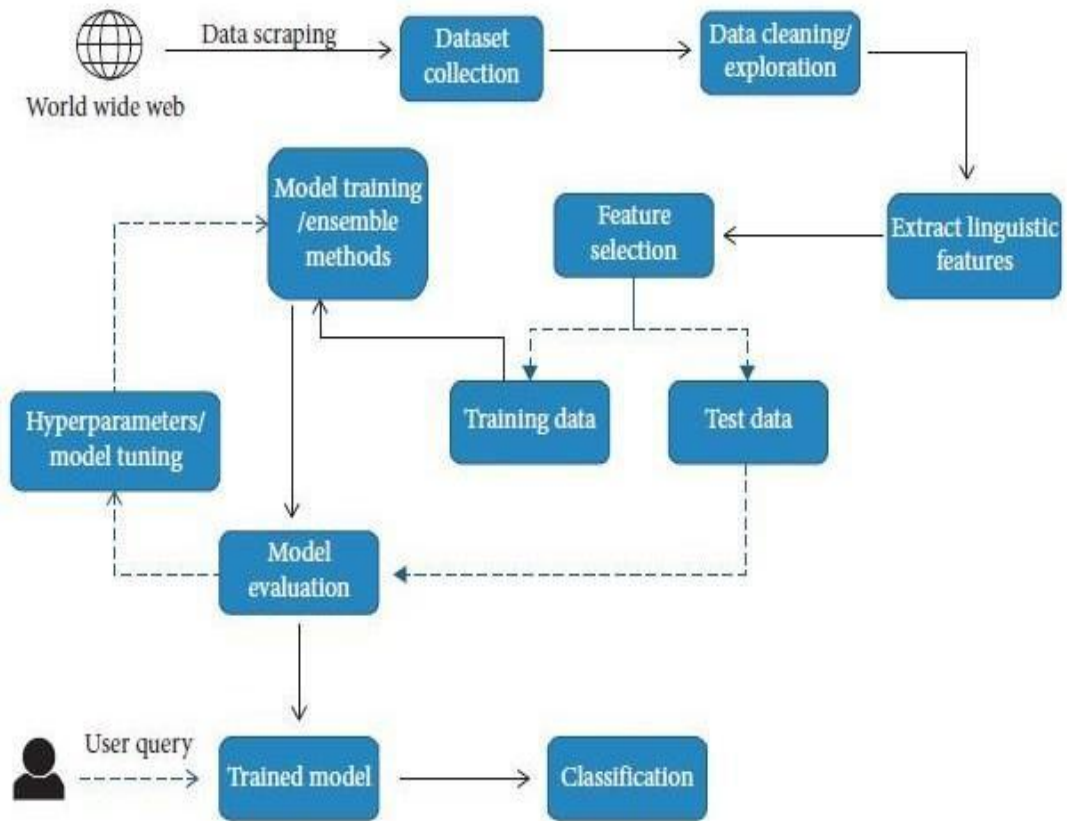


Figure 4.1: Workflow for training algorithms and classification of news articles.

## CHAPTER 5

### RESULTS AND DISCUSSION

The precision of each algorithm on the four datasets considered is summarised in Table-1. It is clear that DS1 (ISOT Fake news data set) achieves a maximum precision of 99 per cent by using random forest algorithms and Perez-LSVM. Linear SVM, multi-layer, bagger and boosting classifications have reached 98 percent accuracy. The average precision of ensemble learners is 97.67 percent for DS1, compared to 95.25 percent for individual students. The absolute difference between individual students and ensemble students is 2.42%, which does not matter. Wang-CNN and Wang- Bi-LSTM benchmark algorithms have been poorer than any other algorithm. On DS 2, the best performing algorithms are bagging classifiers (decision trees) and increasing classifiers (XGBoost) with a 94% accuracy. Interestingly, DS2 was not well performed by linear SVM, random forest and Perez-LSVM. Individual students found 47.75 percent accuracy, compared to 81.5 percent accuracy among ensemble learners. A comparable pattern is noted with DS3, in which the accuracy of individual students is 80% while the accuracy of ensemble students is 93.5%. Unlike DS2, Perez- LSVM, which attained 96 percent accuracy, is the best performing DS3 algorithm. The best performing algorithm for DS4 (DS1, DS2, and DS3 combined) is random forest (91 percent accuracy). On average, individual students have attained 85% accuracy and ensemble students have attained 88,16% accuracy. Wang-Bi-LS, which achieves a precision of 62%, is the best performing algorithm.

TABLE 5.1: Overall accuracy score for each dataset.

|                                     | DS1  | DS2  | DS3  | DS4  |
|-------------------------------------|------|------|------|------|
| <b>Algorithms</b>                   |      |      |      |      |
| Logistic regression (LR)            | 0.97 | 0.91 | 0.91 | 0.87 |
| Perez-LSVM                          | 0.99 | 0.79 | 0.96 | 0.9  |
| Wang-CNN                            | 0.87 | 0.66 | 0.58 | 0.73 |
| Wang-Bi-LSTM                        | 0.86 | 0.52 | 0.57 | 0.62 |
| <b>Ensemble learners</b>            |      |      |      |      |
| Random forest (RF)                  | 0.99 | 0.35 | 0.95 | 0.91 |
| Voting classifier (RF, LR, KNN)     | 0.97 | 0.88 | 0.94 | 0.88 |
| Voting classifier (LR, LSVM, CART)  | 0.96 | 0.86 | 0.92 | 0.85 |
| Bagging classifier (decision trees) | 0.98 | 0.94 | 0.94 | 0.9  |
| Boosting classifier (AdaBoost)      | 0.98 | 0.92 | 0.92 | 0.86 |
| Boosting classifier (XGBoost)       | 0.98 | 0.94 | 0.94 | 0.89 |



TABLE 5.2: Precision on the 4 datasets.

|                                     | DS1  | DS2  | DS3  | DS4  |
|-------------------------------------|------|------|------|------|
| Algorithms                          |      |      |      |      |
| Logistic regression (LR)            | 0.98 | 0.92 | 0.93 | 0.88 |
| Perez-LSVM                          | 0.99 | 0.79 | 0.96 | 0.9  |
| Wang-CNN                            | 0.84 | 0.65 | 0.48 | 0.72 |
| Wang-Bi-LSTM                        | 0.92 | 0.43 | 0.5  | 0.65 |
| Ensemble learners                   |      |      |      |      |
| Random forest (RF)                  | 0.99 | 0.3  | 0.98 | 0.92 |
| Voting classifier (RF, LR, KNN)     | 0.96 | 0.88 | 0.92 | 0.86 |
| Voting classifier (LR, LSVM, CART)  | 0.94 | 0.86 | 0.88 | 0.83 |
| Bagging classifier (decision trees) | 0.98 | 0.94 | 0.93 | 0.9  |
| Boosting classifier (AdaBoost)      | 0.98 | 0.92 | 0.92 | 0.86 |
| Boosting classifier (XGBoost)       | 0.99 | 0.94 | 0.96 | 0.92 |

TABLE 5.3: Recall on the 4 datasets.

|                                     | DS1  | DS2  | DS3  | DS4  |
|-------------------------------------|------|------|------|------|
| Algorithms                          |      |      |      |      |
| Logistic regression (LR)            | 0.98 | 0.9  | 0.92 | 0.86 |
| Perez-LSVM                          | 0.99 | 0.81 | 0.97 | 0.91 |
| Wang-CNN                            | 0.9  | 0.71 | 0.29 | 0.75 |
| Wang-Bi-LSTM                        | 0.78 | 0.59 | 0.35 | 0.61 |
| Ensemble learners                   |      |      |      |      |
| Random forest (RF)                  | 1    | 0.34 | 0.93 | 0.91 |
| Voting classifier (RF, LR, KNN)     | 0.97 | 0.89 | 0.96 | 0.9  |
| Voting classifier (LR, LSVM, CART)  | 0.97 | 0.87 | 0.96 | 0.89 |
| Bagging classifier (decision trees) | 0.97 | 0.95 | 0.94 | 0.91 |
| Boosting classifier (AdaBoost)      | 0.98 | 0.93 | 0.92 | 0.86 |
| Boosting classifier (XGBoost)       | 0.99 | 0.94 | 0.94 | 0.89 |

TABLE 5.4: F1-score on the 4 datasets.

|                                     | DS1  | DS2  | DS3  | DS4  |
|-------------------------------------|------|------|------|------|
| Algorithms                          |      |      |      |      |
| Logistic regression (LR)            | 0.98 | 0.91 | 0.92 | 0.87 |
| Perez-LSVM                          | 0.99 | 0.8  | 0.96 | 0.9  |
| Wang-CNN                            | 0.87 | 0.67 | 0.31 | 0.73 |
| Wang-Bi-LSTM                        | 0.84 | 0.44 | 0.35 | 0.57 |
| Ensemble learners                   |      |      |      |      |
| Random forest (RF)                  | 0.99 | 0.32 | 0.95 | 0.91 |
| Voting classifier (RF, LR, KNN)     | 0.97 | 0.88 | 0.94 | 0.88 |
| Voting classifier (LR, LSVM, CART)  | 0.96 | 0.86 | 0.92 | 0.86 |
| Bagging classifier (decision trees) | 0.98 | 0.94 | 0.94 | 0.9  |
| Boosting classifier (AdaBoost)      | 0.98 | 0.92 | 0.92 | 0.86 |
| Boosting classifier (XGBoost)       | 0.99 | 0.94 | 0.95 | 0.9  |

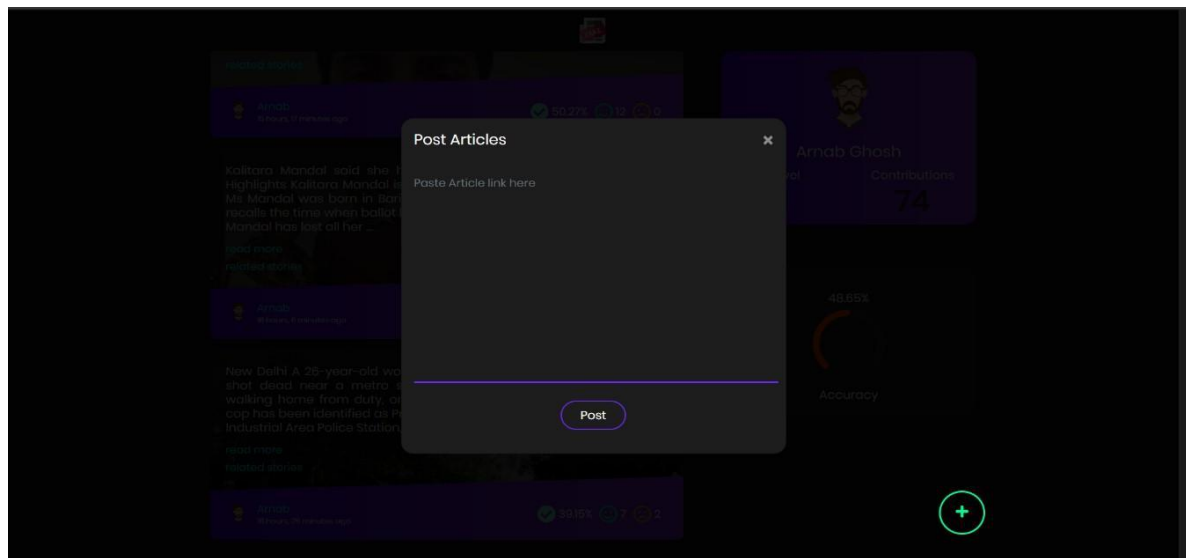


Figure 5.1: Post the Link of the webpage to check

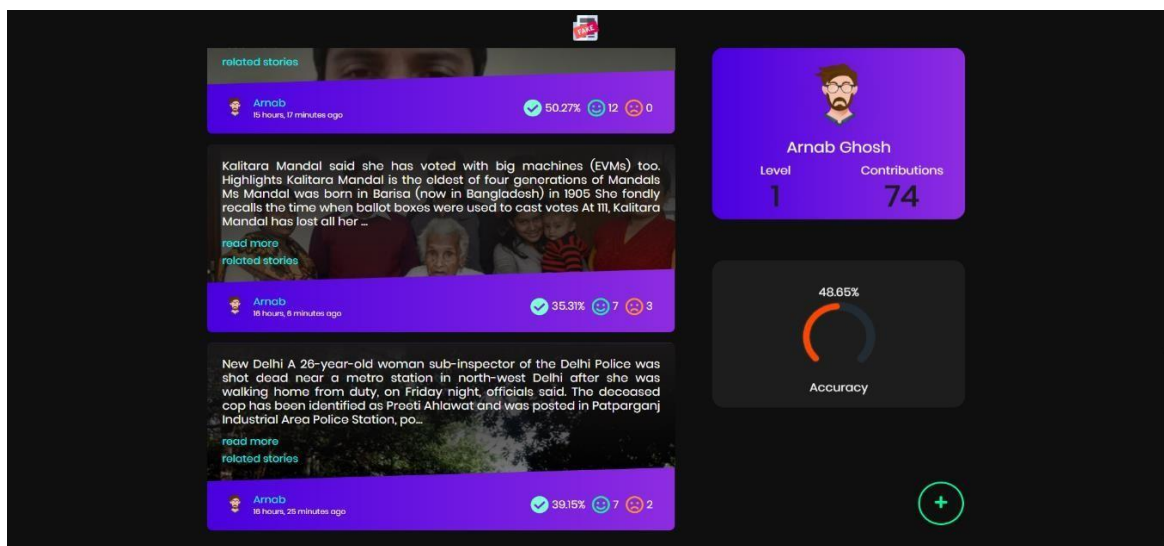


Figure 5.2: Displayed Result

## **CHAPTER 6**

### **CONCLUSION**

#### **CONCLUSION**

In order to manually classify messages, a thorough understanding of the field and expertise are required to detect text anomalies. We tackled the issue of classifying fake news items using models and ensemble methods. In this study we examined The data we used in our work is gathered from the World Wide Web and includes news items from different domains to cover most of the news instead of classifying political news in particular. The research's primary purpose is to detect text patterns that distinguish fake articles from true news. We have collected various text characteristics from the articles using a LIWC tool and used the set as an input to the models. The learning models were trained to get optimum accuracy and set parameters. Some models have a greater precision than others. In comparison with each algorithm we use different performance metrics. In comparison to the individual students, the ensemble students have demonstrated a higher score on all the performance criteria.

Fake news detection has many open problems which need researchers' attention. For example, the identification of significant components involved in the distribution of news is an important step to minimise the spread of fake news. Graph theory and machine learning approaches can be used to detect the main sources of false news distribution. Also, a possible future direction may be in real time false news identification in videos.

## REFERENCES

- [1] Douglas, “News consumption and the new electronic media,” *International Journal of Press/Politics*, vol. 11, no. 1, pp. 29–52, 2006.
- [2] J. Wong, “Almost all the traffic to fake news sites is from facebook, new data show,” 2016.
- [3] D. M. J. Lazer, M. A. Baum, Y. Benkler et al., “science of fake news,” *Science*, vol. 359, no. 6380, pp. 1094–1096, 2018.
- [4] S. A. Garc’ia, G. G. Garc’ia, M. S. Prieto, A. J. M. Guerrero, and C. R. Jimenez, “impact of term fake news on the scientific community scientific performance and mapping in web of science,” *Social Sciences*, vol. 9, no. 5, 2020.
- [5] D. Holan, *2016 Lie of the Year: Fake News*, Politifact, Washington, DC, USA, 2016.
- [6] S. Kogan, T. J. Moskowitz, and M. Niessner, “Fake News: Evidence from Financial Markets,” 2019, <https://ssrn.com/abstract=3237763>.
- [7] Robb, “Anatomy of a fake news scandal,” *Rolling Stone*, vol. 1301, pp. 28–33, 2017.
- [8] J. Soll, “long and brutal history of fake news,” *Politico Magazine*, vol. 18, no. 12, 2016.
- [9] J. Hua and R. Shaw, “Corona virus (covid-19) “infodemic” and emerging issues through a data lens: the case of China,” *International Journal of Environmental Research and Public Health*, vol. 17, no. 7, p. 2309, 2020.
- [10] N. K. Conroy, V. L. Rubin, and Y. Chen, “Automatic deception detection: methods for finding fake news,” *Proceedings of the Association for Information Science and Technology*, vol. 52, no. 1, pp. 1–4, 2015.
- [11] T. Asr and M. Taboada, “Misinfotext: a collection of news articles, with false and true labels,” 2019.
- [12] K. Shu, A. Sliva, S. Wang, J. Tang, and H. Liu, “Fake news detection on social media,” *ACM SIGKDD Explorations Newsletter*, vol. 19, no. 1, pp. 22–36, 2017.
- [13] S. Vosoughi, D. Roy, and S. Aral, “spread of true and false news online,” *Science*, vol. 359, no. 6380, pp. 1146–1151, 2018.
- [14] Allcott and M. Gentzkow, “Social media and fake news in the 2016 election,” *Journal of Economic Perspectives*, vol. 31, no. 2, pp. 211–236, 2017.
- [15] V. L. Rubin, N. Conroy, Y. Chen, and S. Cornwell, “Fake news or truth? using satirical cues to detect potentially misleading news,” in *Proceedings of the Second Workshop on Computational Approaches to Deception Detection*, pp. 7–17, San Diego, CA, USA, 2016.
- [16] H. Jwa, D. Oh, K. Park, J. M. Kang, and H. Lim, “exBAKE: automatic fake news detection model based on bidirectional encoder representations from transformers (bert),” *Applied Sciences*, vol. 9, no. 19, 2019.
- [17] Ahmed, I. Traore, and S. Saad, “Detection of online fake news using n-gram analysis and machine learning techniques,” in *Proceedings of the International Conference on Intelligent, Secure, and Dependable Systems in Distributed and Cloud Environments*, pp. 127–138, Springer, Vancouver, Canada, 2017.
- [18] W. Y. Wang, *Liar, Liar Pants on Fire: A New Benchmark Dataset for Fake News Detection*, Association for Computational Linguistics, Stroudsburg, PA, USA, 2017.
- [19] Riedel, I. Augenstein, G. P. Spithourakis, and S. Riedel, “A simple but tough-to-beat baseline for the fake news challenge stance detection task,” 2017, <https://arxiv.org/abs/1707.03264>.
- [20] V. Perez’-Rosas, B. Kleinberg, A. Lefevre, and R. Mihalcea, “Automatic detection of fake news,” 2017, <https://arxiv.org/abs/1708.07104>.
- [21] H. Ahmed, I. Traore, and S. Saad, “Detecting opinion spams and fake news using text classification,” *Security and Privacy*, vol. 1, no. 1, 2018.
- [22] Kaggle, *Fake News*, Kaggle, San Francisco, CA, USA, 2018, <https://www.kaggle.com/c/fake-news>.
- [23] Kaggle, *Fake News Detection*, Kaggle, San Francisco, CA, USA, 2018, <https://www.kaggle.com/jruvika/fake-news-detection>.
- [24] J. Bergstra and Y. Bengio, “Random search for hyper-parameter optimization,” *Journal of Machine Learning Research*, vol. 13, pp. 281–305, 2012.
- [25] T. M. Mitchell, *Discipline of Machine Learning*, Carnegie Mellon University, Pittsburgh, PA, USA, 2006.
- [26] D. Ruta and B. Gabrys, “Classifier selection for majority voting,” *Information Fusion*, vol. 6, no. 1, pp. 63–81, 2005.
- [27] B. Gregorutti, B. Michel, and P. Saint-Pierre, “Correlation and variable importance in random forests,”

*Statistics and Computing*, vol. 27, no. 3, pp. 659–678, 2017.

- [28] L. Breiman, J. Friedman, R. Olshen, and C. Stone, *Classification and Regression Trees*, Springer, Berlin, Germany, 1984.
- [29] R. E. Schapire, "A brief introduction to boosting," *IJCAI*, vol. 99, pp. 1401–1406, 1999
- [30] E. M. Dos Santos, R. Sabourin, and P. Maupin, "Overfitting cautious selection of classifier ensembles with genetic algorithms," *Information Fusion*, vol. 10, no. 2, pp. 150–162, 2009
- [31] T. Chen and C. Guestrin, "Xgboost: a scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 785–794, San Francisco, CA, USA, 2016
- [32] T. Hastie, S. Rosset, J. Zhu, and H. Zou, "Multi-class ada-boost," *Statistics and its Interface*, vol. 2, no. 3, pp. 349–360, 2009
- [33] L. Lam and S. Y. Suen, "Application of majority voting to pattern recognition: an analysis of its behavior and performance," *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 27, no. 5, pp. 553–568, 1997.

## APPENDIX

### Source Code –

```
import pickle
import re
from urllib.parse import urlparse

import joblib
import numpy as np
import pandas as pd
from googlesearch import search
from newspaper import Article
from sklearn.feature_extraction.text import CountVectorizer, TfidfVectorizer
from sklearn.metrics.pairwise import cosine_similarity
from sklearn.model_selection import
train_test_split from sklearn.naive_bayes import
MultinomialNB

#extractor function that gets the article body from the url
def extractor(url):
    article = Article(url)
    try:
        article.download()
        article.parse()
    except:
        pass

    #gets some of the article features like title
    article_title = article.title
    article = article.text.lower()
    article = [article]
    return (article, article_title)

#function for the textbox extractor
#it doesn't need an API
def textAreaExtractor(text):
    text = text.lower()
    text = re.sub(r'[^a-zA-Z0-9\s]', ' ', text)
    text = re.sub("(\r|\r\n)\n$", " ", text)
    text = [text]
    return text

#performs a google search using the article title
```



```

search_title = []
search_urls = []
source_sites = []
#stores the sitenames and the urls into a list
for i in search(title, tld = "com", num = 10, start = 1, stop = 6):
    if "youtube" not in i and domain not in i:
        source_sites.append(urlparse(i).hostname)
        search_urls.append(i)
        """article = Article(i)
        try:
            article.download()
            article.parse()
        except:
            pass
        title = article.title"""

```

```

return search_urls, source_sites

```

```

def similarity(url_list, article):
    article = article
    sim_tfv = TfidfVectorizer(stop_words = "english")
    #article needs to be vectorized first
    sim_transform1 = sim_tfv.fit_transform(article)
    cosine = []
    cosineCleaned = []
    cosineAverage = 0
    count = 0
    #loop to calculate each article from the google search
    #against the original article
    for i in url_list:
        test_article, test_title = extractor(i)
        test_article = [test_article]
        sim_transform2 = sim_tfv.transform(test_article[0])
        score = cosine_similarity(sim_transform1, sim_transform2)
        cosine.append(score*100)
        print("Article " + str(count) + " similarity calculated")
        count+=1
    for i in cosine:
        x = str(i).replace('[', '').replace(']', '')
        cosineCleaned.append(x)

    #averages the similarity score
    averageScore = cosineAverage/count

```



```
print(averageScore)
return cosineCleaned, averageScore

#classification function
def handlelink(article_link):

    #loads the models
    job_cv =
    joblib.load("Dashboard/static/models/cv.pkl") job_pac
    = joblib.load("Dashboard/static/models/pac.pkl")
    job_vec =
    joblib.load("Dashboard/static/models/tfv.pkl") url =
    (article_link)

    #extracts the article and title from the
    url article, article_title =
    extractor(article_link)

    #prediction is made
    pred =
    job_pac.predict(job_vec.transform(article))
    print("Target article has been classified")
```

## **Journal Paper**

# Identification of Fake News using Machine Learning

Shubham Rai  
U.G Scholar  
School of Computing  
Sathyabama Institute of Science and  
Technology Chennai, India  
shubham15031998@gmail.com

Krish Kumar Hissaria  
U.G Scholar  
School of Computing  
Sathyabama Institute of Science and  
Technology Chennai, India  
hissariakrish3@gmail.com

Dr. S. Revathy  
Associate Professor  
School of Computing  
Sathyabama Institute of Science and  
Technology Chennai, India  
revathy.it@sathyabama.ac.in

with reality has created many issues not only in politics,

## Abstract

*Since the Internet boomed, fake news has been an issue. The very network that lets us know what is going on worldwide is the perfect source of malice and counterfeit news. It is vital to combat this false news because information shapes the world's perspective. People make critical informational decisions as well as their own views. It will have disastrous consequences if this information is incorrect. It is totally impossible to verify every news one at a time by a human being. This paper tries to speed up the process of identifying false news by suggesting a system to classify fake news accurately. In eight different datasets, derived from a variety of sources, Machine Learning algorithms like Naive Bayes, passive aggressive classification and deep neural networks have been used. In addition the analysis and results of each model are included. By using the appropriate model with the right tools, the arduous task of detecting fake news can be trivialised. The findings of this project show the ability to teach machines to help in this task. We have developed a model that captures many intuitive indicators of real and false news and an application that enables the classification decisions to be displayed.*

## I. Introduction

With the rise of the internet, and the fast adoption of social media platforms, the dissemination of data has paved the way for the history of humankind. In addition, news outlets have also benefited from the widespread usage of social media sites in almost real time by providing their subscribers with updated news. From newspapers, tab lists and journals, news media moved into a digital platform such as news websites, social- media news, etc.[1]. Consumers have been made more easily accessible to the latest stories. 70 percent of news website traffic is represented in Facebook referrals[2]. In their current state, social media platforms are highly powerful and helpful to enable people to debate and discuss topics like democracy, education and health. However, some companies often use the platforms with a negative viewpoint to generate monetary income[3, 4] and, in other cases, to generate partial views, manipulate thoughts and spread satire or absurdity. False news is usually referred to as phenomenon.

In the last decade, the spread of false news has been growing rapidly, most noticed during the US elections in 2016[5]. The proliferation of on-line articles not in line

but also in different fields such as sport, health and science [3]. The financial markets [6] are among those affected by false news, with rumours having catastrophic effects and stopping the market.

Our decision-making capacity depends primarily on the kind of data we use; our worldview is based on data we digest. Consumers have been increasingly reacting nonsensically to news which subsequently proved false. The latest example is the spreading of new corona viruses, in which false reports about the advent, nature and behaviour of the virus have been distributed over the internet[9]. When more people read online about fake content, the condition worsened. Identifying such news on line is a challenging challenge.

Fortunately, a range of computing methods will, in terms of text content, be used to identify some articles as false[10]. Any of these tools are used in websites that are checked by a variety of archives managed by scholars containing lists of websites that are recognised as ambiguous or fake[11]. The question is, however, that human intelligence has to be classified as counterfeit articles. Furthermore, the checking pages include posts about particular areas like politics and are not commonly used for the identification of fake news items from different fields like culture, sport and technology.

Data are available in different formats as emails, videos, and audios on the World Wide Web. It is comparatively difficult to identify and classify online news published in unstructures (such as news, videos and audio), as this purely calls for human know-how. Calculation tools such as Natural Language Processing (NLP) should, also, used to find keywords that separate a dishonest, fact-based text article [12] from articles. Analyzes of fake news in comparison to real news [13] include other tools. In particular, the approach analyses how a false news storey differs from a true article on a network. The answer that an article receives can be theoretically differentiated in order to identify the article as real or false. A more hybrid approach may also be used to assess an article's social reaction as well as investigate the textual characteristics to check whether or not an article is disappointing.

## II.Literature Survey

Wang[18] follows a different strategy. For training various Machine Language models, the Author has used textual characteristics and metadata. The author primarily concentrated on the use of the neural network convolutionary (CNN). The dependence between the metadata vectors is measured with a convolutionary layer, followed by a two-way LSTM layer. A bidirectional LSTM representation of metadata, which was fed into a completely linked layer with a softmax activation function, was concatenated with the max-pool text representations to produce the final prediction. A re- search is carried out on a political domain dataset of statements from two parties. The precision of 27,7% was obtained by a combination of functions like text and speaker, and 27,4% by the combination of the different components of metadata with text was achieved. Riedel et al. [19] provides a competitive solution, which is a position sensor which assigns an article to one of four labels, depending on whether the headline of the article conforms to the text of an article, "agrees," "discrees," "discusses," or "unrelated." The author has the function set of linguistic properties such as term frequency (TF) and term frequency inverted document frequency (TF-IDF), and a multi-layer perceptron (MLP) classifier is used on the output of the final layer, with a secret layer and a softmax function. Data set included headline, body and label articles. Data set The accuracy of the system in test examples on the "disagree" label was weak, while the label "agree" shows the best performance. Authors have used a basic, finely tuned MLP with some hyperparameters to obtain a total precision of 88.46%. Shu et al.[12] have also reviewed a range of approaches for veracity evaluation to identify online fake news. There are two main assessment categories: linguistic and network analytics approaches. Two major evaluation techniques are discussed. A combination of the two offers a stronger hybrid approach to fake online news detection. In linguistic approaches are included deep grammar, rhetorical form, and discussion analyses. Linguistic methods are used in order to train classifications such as SVM or naive Bayes. The study and processing of social networks and associated data behaviour is included in network technologies. Vosoughi et al. [13] took a singular approach to exploring the characteristics of news on social media. This means that writers explored how news is disseminated through social media (rumours) such as Twitter and the circulation on Twitter by fictional news. The authors addressed the importance of this strategy. The article addresses various methods of analysing the dissemination of misleading online news such as depth, height, maximum duration, systemic virality, the mean width of authentic rumours at different depths, the number of Twitter users and the time it takes to get to the true and wrong rumours.

### III. Proposed System

#### A. Proposed Framework

As outlined in Figure 1, we will draw on current literature with a range of linguistic composition approaches to classify news articles in various fields as true or false. The uniqueness of the approach proposed for this research is the combination of approaches used, along with the language inquiry and the Word Count (LIWC).

There are many renowned websites posting legitimate information and a few other sites, for example PolitiFact and Snopes, that are used to verify the facts. In order to retain an accurate list of current data sets and hyperlinks to potential fact check sites which will contribute to avoid fake news being more frequent, researchers[11] are still keeping open repositories. As always, for our experiments we chose three datasets containing news in various fields (e.g. politics, entertainment, technology, sports) and a combination of true and false articles. The World Wide Web offers online data sets and downloads them. ISOT Fake News Dataset [21] is the first dataset; Kaggle [22, 23] has the second and third datasets publicly available.

Before it is used for training the models, the corpus compiled from the Internet is pre-processed. Articles such as publishers, date postings, URLs and categories of unwanted variables have been filtered away. Articles which have no text in the body of the article or less than 20 words are also omitted. Multicolumn products are turned for consistency of formats and structure into single column items. This is done with all datasets so that the format and the structure are consistent.

The next step is the extraction of language properties if during data cleaning and exploration the correct features are selected. Linguistic features have incorporated numerical numerical textual features that can be used with training models. The functional characteristics include the percentage of the words to stop; the word to punctuate; the phrases to work; the casual language. In order to extract corpus characteristics, we have used the LIWC 2015 method, that classifies text into several distinct, continuous variables. The LIWC tool extracts 91 separate functions in any given document. Since all functions obtained from the tool are numerical quantities, categorical variables must not be encoded. Scaling is therefore used to ensure that different feature values are within the range (0, 1). This is needed since certain values range from 0 to

100 (e.g., percentages) (such as word counts). The input function is then used to train various study models on a CPU. Per dataset is divided into a training & test collection of 70/30 split. In testing and test cases, the papers are combined to ensure equal dissemination of false and true products. The study algorithms are equipped to reach maximal accuracy in an optimal equilibrium between variance

and distortion of a given data set using various hyperparameters. Each model is trained several times using a ring quest to optimise the model with a number of different parameters for improved results. The best parameters are found in computationally expensive grid searches;[24] however, measures are taken to prevent the pattern from overfitting and malfunctioning.

In order to examine performances over multiple datasets, different ensemble approaches, including packaging, boosting and the voting classification, are new to this research. A logistic regression, random woods and KNN

group form the first voting type. The second category of voting consists of logistical regression, linear SSM and arbours of grouping and regression. We have used two voting classen consisting of three learning modells (CART). The training includes that all three models train voting classifiers on the basis of large votes based on a collection of results labels and the best parameters of each model. A group of 100 decision making bodies have been prepared, while XGBoost and AdaBoost used two strengthening algorithms.

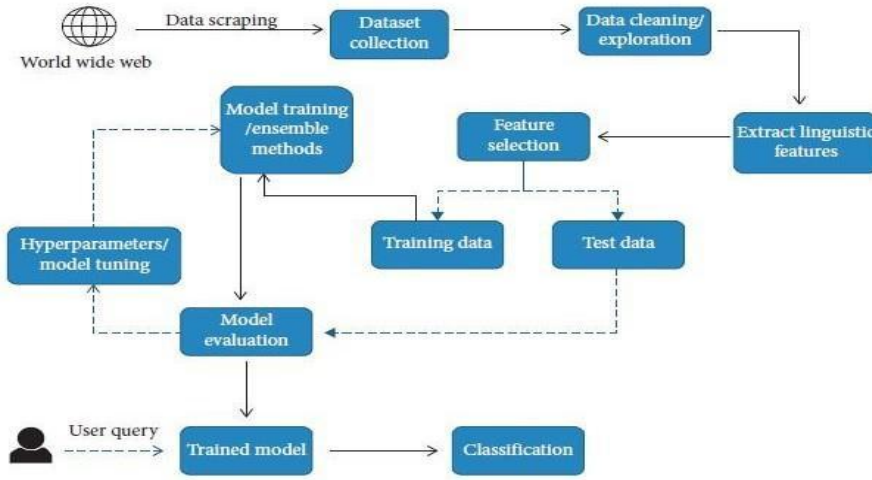


Fig 1: Architecture for algorithms training and news articles classification.

## B. Algorithm

Our technique is used to analyse the performance of fake news detection classifications through the following learning algorithms.

1. Logistic Regression. Because we classify text based on a wide range of features, a logistic regression (LR) model is used with binary output (true/false), as it gives an intuitive equation to classify problems into binary or multiple classes[25]. We have tuned hyperparameters to get the best results for each dataset, while testing different parameters in order to achieve the highest accuracy of the LR model. The logistic hypothesis function can be defined mathematically as follows[25].

$$h_{\theta}(X) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X)}} \quad (1)$$

LR uses a sigmoid function to convert the output into a probability, which is aimed at minimising the cost to reach an optimum likelihood. The cost function as shown is calculated

(2)

$$\text{Cost}(h_{\theta}(x), y) = \begin{cases} \log(h_{\theta}(x)), & y = 1, \\ -\log(1 - h_{\theta}(x)), & y = 0. \end{cases}$$

2. Linear SVM. In [20] we use the linear SVM method. We trained the linear SVM on the function set as outlined in [20] with 5-fold cross validation to ensure meaningful comparison. Please note that in the text the methodology is called Perez-LSVM.
3. Convolutional Neural Network For automatically detecting fake news, Wang [18] used the convolutionary neural network (CNN). We used our sample to follow the same strategy. However, we were unable to use the Wang [18] feature set as the dataset includes

only short statements. The Wang-CNN in the text is the tactic.

4. Bidirectional Long Short-Term Memory Networks. Wang[18] used long-run bidirectional short-run memory networks, with different functional sets. Wang also used the same technique. The Wang-Bi-LSTM in the text is called the methodology.

correlation. In our random forest model, the parameters used to generate the best model that

### C. Ensemble Learners

In order to increase the overall accuracy of the classification between a true article and a false one, we suggested that existing assemble procedures along with textual features be used. Together students tend to have better accuracy, as all of the models are trained to minimise the overall error rate with a specific approach and increase model performance. The intuition behind ensemble modelling is identical with the one already used to in our everyday lives such as seeking the views of different experts in order to mitigate an improper choice or unwanted outcomes before taking a particular decision. For example, a grade algorithm with a special set of parameters can be trained on a specific dataset which can generate a decision limit that is in some extent suitable for the data. The result of this particular algorithm is not only the parameters used to form the model, but also the type of training data. If the training data contains fewer variation or uniform data, the model overruns and produces distorted results over unseen information. To mitigate the risk of overfitting, methods such as cross validation are used. Various models may be conditioned on different parameters to set several judgement limits on randomly chosen data points. Thus, by training various algorithms with ensemble learning techniques, these problems can be solved and mitigated and their results can be combined to produce almost optimal results. This approach involves using voting classifiers, which rely on the biggest votes of all algorithms for the final classification.

However, in different situations such as the following, there are also other ensemble methods.

1. *Random Forest (RF)*: The random forest (RF) is a state-of-the-art decision-making (DT) model that is now a controlled learning model. RF includes a considerable number of decision-making treasures that work independently to predict an outcome in a class in which the ultimate forecast is based on a majority voting class. Random forest error is low in comparison to other models because of low tree[27]

can predict the result with high precision were trained, i.e. in a grid search different number of estimators. are many algorithms to decide on the regression or classification problem in a decision tree. In order to calculate a fraction of the data collection, we used the Gini index as a costs function. The Gini index is calculated by excluding the cumulative probabilities of each class. The Gini Index (Gind) mathematical formula is as follows[28]:

(3)

$$G_{ind} = 1 - \sum_{i=1}^c (P_i)^2,$$

## 2. *Bagging*

*Ensemble Classifiers*: Bootstraps is an early ensemble technique used mostly to minimise (overfit) variance over an exercise in a short classification of bags. One of the most often used model of random forest for bagging. The bagging model intuitively selects a class for the issue of classification based on major voting estimates by the M amount of trees to reduce the total variance while selecting a random sample of each trees to replace the total data collection. However, for regression problems the bagging model averages multiple predictions.

3. *Boosting Ensemble Classifiers*: It is an additional technique commonly used for training weak models to become strong students. To this end, a forest of random trees is generated and the final prediction is based on plurality voting outcomes in each tree. The method helps weak students to interpret data points accurately in a widely misclassified technique. The same weighted coefficients are first used to classify all data points into a given problem. For correctly categorised data points and for misclassified data points, the weighted coefficients are reduced in the following rounds[29]. Each subsequent tree created each round reduces the errors of the previous round by properly classifying data points misclassified in the previous rounds to improve the overall precision. One of the the problems with boosting ensemble is that training data may overflow, leading to inaccurate prognoses for unseen cases [30]. Some boosting algorithms are available that can be used for both grading and regression. We have used

XGBoost[31] and AdaBoost[32] classification algorithms in our trials.

4. *Voting Ensemble Classifiers*: Voting ensemble is generally used for classification issues since this makes it possible to integrate two or more learning models with the preparation of the whole dataset[33]. Each model predicts the outcome of a sample data point that is considered a 'poll' for the model's anticipated class. When each model predicts the outcome, the final prediction is dependent on a plurality voting in a certain class[26]. Ensemble voting is faster when it comes to implementing than bagging and enhancing algorithms. As discussed, package algorithms construct a variety of data sub-sets by randomly sampling the data set and replacing it with the entire data set. The final outcomes of each model are aggregated by each dataset, since each model is used to create a model. Several models are sequentially trained as stimulated, whereby any model learns from the previous models by increasing weight for the incorrect stitches, creating a generalised model that can properly identify the problem. On the other hand, however, voting ensemble is a combination of several independent models which generates results of classifications that contribute to the overall prediction by majority vote.

#### D. Datasets

The data sets we used in this research are available online and open source. Fake and true news articles from several areas are included in the data. Truthful published news articles contain real-world events, while fake news sites contain claims not in line with facts. Compliance by blogs such as politifact.com and snopes.com with the conformity of claims for many of such articles in the political domain is manually verifiable. In this study, we used three different datasets, which are described in a short way.

The first set of data is known as the "ISOT False News Dataset" [21], which includes fake and true articles taken from the Internet. Real articles are collected from a famous website, reuters.com, while fake articles have been obtained from several sources, mostly from websites flagged by politifact.com. Datasets include 45,098 articles, of which 22,487 are real articles and 22,611 are fake articles. Total corporations contain articles from various fields that target most political news.

A second data-set of 21,386 articles for training and 6,106 articles for testation is available from the Kaggle[22] (hereinafter referred to as DS2). The dataset is constructed from a variety of Internet sources. Article

not including false and true elements in separate fields are only limited to one area. It's not just a single field. The third dataset, which contains a total of 4,322 fake and true articles, is also available at Kaggle [23] (hereafter DS3). Real articles have been collected from reputable sources such as the CNN, the newspaper Reuters, The New York Times and other sources. Sports, entertainment and politics are domains covered.

An essay compilation of the three datasets is a fusion of data (hereafter referred to as DS4). The fourth dataset is intended to evaluate the output of algorithms in a unique dataset spanning a wide variety of realms, due to the existence of the papers in each data set.

#### E. Performance Metrics

We also used various metrics to evaluate algorithms efficiency. Much of them are founded on the instability matrix. The confusion matrix is a table depicting a test set classification model composed of the four parameters: real positive, false positive and real negative (see Table 1).

1. *Accuracy*: It is the most commonly used metric, which represents the proportion of true or false predictions of observations. The following equation can be used to compute the accuracy of a model performance:

(4)

High-precision values are always fine, but since we are trained on a classification model in the case we are trained in, an article which was predicted to be true when false (false positive) has negative effects which would lead to trust issues if an article is predicted to be false. Three additional metrics were used which consider incorrect classification observations, i.e. precision, reminder and F1 score.

2. *Recall*: It shows the total number of good classifications from the true class. In our case, the number of articles expected to be true is the total number of true articles.

Table 1: Confusion Matrix

precision of 99 per cent by using random forest

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

|              | Predicted True      | Predicted False     |
|--------------|---------------------|---------------------|
| Actual true  | True positive (TP)  | False negative (FN) |
| Actual false | False positive (FP) | True negative (TN)  |

(5)

3. Precision:

Conversely, the accuracy score reflects the true positive ratio to all forecast events. In our case, accuracy indicates that of all positive (true) articles, the number of articles marked as true:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

(6)

4. F1 Score: F1-score

is the compromise between accuracy and reminder. The harmonic mean between the two is calculated. The false positive observations are both taken into account and the false negative ones. The following formula can be used to determine F1-score:

$$\text{F1 - score} = 2 \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

(7)

#### IV. Results

The precision of each algorithm on the four datasets considered is summarised in Table 2. It is clear that DS1 (ISOT Fake news data set) achieves a maximum

algorithms and LSVM. Linear SVM, multi-layer, bagger and boosting classifications have reached 98 percent accuracy. The average precision of ensemble learners is

97.57 percent for Data Set 1, compared to 95.25 percent for individual students. The total disparity between person and group students is 2.24%, which does not matter. CNN and Bi-LSTM benchmark algorithms have been poorer than any other algorithm. On DS 2, the best performing algorithms are bagging classifiers (decision trees) and increasing classifiers (XGBoost) with a 94% accuracy. Interestingly, DS2 was not well performed by linear SVM, random forest and Perez- LSVM. Individual students found 47.75 percent accuracy, compared to 81.5 percent accuracy among ensemble learners. A comparable pattern is noted with DS3, in which the accuracy of individual students is 80% while the accuracy of ensemble students is 93.5%. Unlike DS2, LSVM, which attained 96 percent accuracy, is the best performing DS3 algorithm. The best performing algorithm for Data Set 4 (Data Set 1, Data Set 2, and Data Set 3 combined) is random forest (92 percent accuracy). On average, individual students have attained 85% accuracy and ensemble students have attained 88,16% accuracy. Wang-Bi-LS, which achieves a precision of 62%, is the best performing algorithm.

TABLE 2: Accuracy score of each dataset.

|                                   | Data-Set 1 | Data-Set 2 | Data-Set 3 | Data-Set 4 |
|-----------------------------------|------------|------------|------------|------------|
| Algorithms                        |            |            |            |            |
| Logistic regression               | 0.97       | 0.91       | 0.91       | 0.87       |
| Lagrangian Support Vector Machine | 0.99       | 0.79       | 0.96       | 0.9        |
| CNN                               | 0.87       | 0.66       | 0.58       | 0.73       |
| Bi-LSTM                           | 0.86       | 0.52       | 0.57       | 0.62       |
| Ensemble learners                 |            |            |            |            |
| Random forest (RF)                | 0.99       | 0.35       | 0.95       | 0.91       |
| Voting classifier (RF, LR, KNN)   | 0.97       | 0.88       | 0.94       | 0.88       |

|                                     |      |      |      |      |
|-------------------------------------|------|------|------|------|
| Voting classifier (LR, LSVM, CART)  | 0.96 | 0.86 | 0.92 | 0.85 |
| Bagging classifier (decision trees) | 0.98 | 0.94 | 0.94 | 0.9  |
| Boosting classifier (AdaBoost)      | 0.98 | 0.92 | 0.92 | 0.86 |
| Boosting classifier (XGBoost)       | 0.98 | 0.94 | 0.94 | 0.89 |

TABLE 3: Precision on all the datasets.

|                                   | Data-Set 1 | Data-Set 2 | Data-Set 3 | Data-Set 4 |
|-----------------------------------|------------|------------|------------|------------|
| Algorithms                        |            |            |            |            |
| Logistic regression               | 0.98       | 0.92       | 0.93       | 0.88       |
| Lagrangian Support Vector Machine | 0.99       | 0.79       | 0.96       | 0.9        |
| CNN                               | 0.84       | 0.65       | 0.48       | 0.72       |

|                                     |      |      |      |      |
|-------------------------------------|------|------|------|------|
| Bi-LSTM                             | 0.92 | 0.43 | 0.5  | 0.65 |
| Ensemble learners                   |      |      |      |      |
| Random forest (RF)                  | 0.99 | 0.3  | 0.98 | 0.92 |
| Voting classifier (RF, LR, KNN)     | 0.96 | 0.88 | 0.92 | 0.86 |
| Voting classifier (LR, LSVM, CART)  | 0.94 | 0.86 | 0.88 | 0.83 |
| Bagging classifier (decision trees) | 0.98 | 0.94 | 0.93 | 0.9  |
| Boosting classifier (AdaBoost)      | 0.98 | 0.92 | 0.92 | 0.86 |
| Boosting classifier (XGBoost)       | 0.99 | 0.94 | 0.96 | 0.92 |

TABLE 4: Recall on all the datasets.

|                                     | Data-Set 1 | Data-Set 2 | Data-Set 3 | Data-Set 4 |
|-------------------------------------|------------|------------|------------|------------|
| Algorithms                          |            |            |            |            |
| Logistic regression                 | 0.98       | 0.9        | 0.92       | 0.86       |
| Lagrangian Support Vector Machine   | 0.99       | 0.81       | 0.97       | 0.91       |
| CNN                                 | 0.9        | 0.71       | 0.29       | 0.75       |
| Bi-LSTM                             | 0.78       | 0.59       | 0.35       | 0.61       |
| Ensemble learners                   |            |            |            |            |
| Random forest (RF)                  | 1          | 0.34       | 0.93       | 0.91       |
| Voting classifier (RF, LR, KNN)     | 0.97       | 0.89       | 0.96       | 0.9        |
| Voting classifier (LR, LSVM, CART)  | 0.97       | 0.87       | 0.96       | 0.89       |
| Bagging classifier (decision trees) | 0.97       | 0.95       | 0.94       | 0.91       |
| Boosting classifier (AdaBoost)      | 0.98       | 0.93       | 0.92       | 0.86       |
| Boosting classifier (XGBoost)       | 0.99       | 0.94       | 0.94       | 0.89       |

TABLE 5: F1-score on all the datasets.

|                                     | Data-Set 1 | Data-Set 2 | Data-Set 3 | Data-Set 4 |
|-------------------------------------|------------|------------|------------|------------|
| Algorithms                          |            |            |            |            |
| Logistic regression                 | 0.98       | 0.91       | 0.92       | 0.87       |
| Lagrangian Support Vector Machine   | 0.99       | 0.8        | 0.96       | 0.9        |
| CNN                                 | 0.87       | 0.67       | 0.31       | 0.73       |
| Bi-LSTM                             | 0.84       | 0.44       | 0.35       | 0.57       |
| Ensemble learners                   |            |            |            |            |
| Random forest (RF)                  | 0.99       | 0.32       | 0.95       | 0.91       |
| Voting classifier (RF, LR, KNN)     | 0.97       | 0.88       | 0.94       | 0.88       |
| Voting classifier (LR, LSVM, CART)  | 0.96       | 0.86       | 0.92       | 0.86       |
| Bagging classifier (decision trees) | 0.98       | 0.94       | 0.94       | 0.9        |
| Boosting classifier (AdaBoost)      | 0.98       | 0.92       | 0.92       | 0.86       |
| Boosting classifier (XGBoost)       | 0.99       | 0.94       | 0.95       | 0.9        |

## V. Conclusion

In order to manually classify messages, a thorough understanding of the field and expertise are required to detect text anomalies. We dealt with the classification of counterfeit news products using templates and ensemble approaches. We examined this study and gathered the information that we use in our work from the Internet and including news reports from different sectors. Identifying text characteristics that distinguish fake reports from true news is the key goal of the study. Different text characteristics were collected from LIWC publications and the array was used as a template input. The learning models have been conditioned to achieve optimal precision and set parameters. Certain models are more accurate than

others. In comparison with each

Algorithm we use different performance metrics. In comparison to the individual students, the ensemble students have demonstrated a higher score on all the performance criteria.

There are several open issues of fake news identification that require attention from researchers. The key inspiration for the dissemination of fake news can be used in graph theory and machine learning approaches. Fake news discovery of videos could also be a potential future path in real time.

## References

- [1] Douglas, "News consumption and the new electronic media," *International Journal of Press/Politics*, vol. 11, no. 1, pp. 29–52, 2006.
- [2] J. Wong, "Almost all the traffic to fake news sites is from facebook, new data show," 2016.
- [3] D. M. J. Lazer, M. A. Baum, Y. Benkler et al., "science of fake news," *Science*, vol. 359, no. 6380, pp. 1094–1096, 2018.
- [4] S. A. Garc'ia, G. G. Garc'ia, M. S. Prieto, A. J. M. Guerrero, and C. R. Jimenez, "impact of term fake news on the scientific community scientific performance and mapping in web of science," *Social Sciences*, vol. 9, no. 5, 2020.
- [5] D. Holan, 2016 Lie of the Year: Fake News, Politifact, Washington, DC, USA, 2016.
- [6] S. Kogan, T. J. Moskowitz, and M. Niessner, "Fake News: Evidence from Financial Markets," 2019, <https://ssrn.com/abstract=3237763>.
- [7] Robb, "Anatomy of a fake news scandal," *Rolling Stone*, vol. 1301, pp. 28–33, 2017.
- [8] J. Soll, "long and brutal history of fake news," *Politico Magazine*, vol. 18, no. 12, 2016.
- [9] J. Hua and R. Shaw, "Corona virus (covid-19) "infodemic" and emerging issues through a data lens: the case of China," *International Journal of Environmental Research and Public Health*, vol. 17, no. 7, p. 2309, 2020.
- [10] N. K. Conroy, V. L. Rubin, and Y. Chen, "Automatic deception detection: methods for finding fake news," *Proceedings of the Association for Information Science and Technology*, vol. 52, no. 1, pp. 1–4, 2015.
- [11] T. Asr and M. Taboada, "Misinfotext: a collection of news articles, with false and true labels," 2019.
- [12] K. Shu, A. Sliva, S. Wang, J. Tang, and H. Liu, "Fake news detection on social media," *ACM SIGKDD Explorations Newsletter*, vol. 19, no. 1, pp. 22–36, 2017.
- [13] S. Vosoughi, D. Roy, and S. Aral, "spread of true and false news online," *Science*, vol. 359, no. 6380, pp. 1146–1151, 2018.
- [14] Allcott and M. Gentzkow, "Social media and fake news in the 2016 election," *Journal of Economic Perspectives*, vol. 31, no. 2, pp. 211–236, 2017.
- [15] V. L. Rubin, N. Conroy, Y. Chen, and S. Cornwell, "Fake news or truth? using satirical cues to detect potentially misleading news," in *Proceedings of the Second Workshop on Computational Approaches to Deception Detection*, pp. 7–17, San Diego, CA, USA, 2016.
- [16] H. Jwa, D. Oh, K. Park, J. M. Kang, and H. Lim, "exBAKE: automatic fake news detection model based on bidirectional encoder representations from transformers (bert)," *Applied Sciences*, vol. 9, no. 19, 2019.
- [17] Ahmed, I. Traore, and S. Saad, "Detection of online fake news using n-gram analysis and machine learning techniques," in *Proceedings of the International Conference on Intelligent, Secure, and Dependable Systems in Distributed and Cloud Environments*, pp. 127–138, Springer, Vancouver, Canada, 2017.
- [18] W. Y. Wang, *Liar, Liar Pants on Fire: A New Benchmark Dataset for Fake News Detection*, Association for Computational Linguistics, Stroudsburg, PA, USA, 2017.
- [19] Riedel, I. Augenstein, G. P. Spithourakis, and S. Riedel, "A simple but tough-to-beat baseline for the fake news challenge stance detection task," 2017, <https://arxiv.org/abs/1707.03264>.
- [20] V. Perez'-Rosas, B. Kleinberg, A. Lefevre, and R. Mihalcea, "Automatic detection of fake news," 2017, <https://arxiv.org/abs/1708.07104>.
- [21] H. Ahmed, I. Traore, and S. Saad, "Detecting opinion spams and fake news using text classification," *Security and Privacy*, vol. 1, no. 1, 2018.
- [22] Kaggle, Fake News, Kaggle, San Francisco, CA, USA, 2018, <https://www.kaggle.com/c/fake-news>.
- [23] Kaggle, Fake News Detection, Kaggle, San Francisco, CA, USA, 2018, <https://www.kaggle.com/jruvika/fake-news-detection>.
- [24] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," *Journal of Machine Learning Research*, vol. 13, pp. 281–305, 2012.
- [25] T. M. Mitchell, *Discipline of Machine Learning*, Carnegie Mellon University, Pittsburgh, PA, USA, 2006.
- [26] D. Ruta and B. Gabrys, "Classifier selection for majority voting," *Information Fusion*, vol. 6, no. 1, pp. 63–81, 2005.
- [27] B. Gregorutti, B. Michel, and P. Saint-Pierre, "Correlation and variable importance in random forests," *Statistics and Computing*, vol. 27, no. 3, pp. 659–678, 2017.
- [28] L. Breiman, J. Friedman, R. Olshen, and C. Stone, *Classification and Regression Trees*, Springer, Berlin, Germany, 1984.
- [29] R. E. Schapire, "A brief introduction to boosting," *IJCAI*, vol. 99, pp. 1401–1406, 1999.
- [30] E. M. Dos Santos, R. Sabourin, and P. Maupin, "Overfitting cautious selection of classifier ensembles with genetic algorithms," *Information Fusion*, vol. 10, no. 2, pp. 150–162, 2009.
- [31] T. Chen and C. Guestrin, "Xgboost: a scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 785–794, San Francisco, CA, USA, 2016.

[32] T. Hastie, S. Rosset, J. Zhu, and H. Zou, “Multi-class ada-boost,” *Statistics and its Interface*, vol. 2, no. 3, pp. 349–360, 2009.



