**Synopsis Report**

**Source Adaptive Disinformation Detection**

**Group ID**: 19 **Project mentor**: Dr. Irfan Siddavatam

**Team members** – Heet Sakaria (Team leader), Vedang Parasnis, Hritik Jaiswal

# Abstract

In the recent years, the topic of fake news has experienced a resurgence of interest in society. The increased attention stems largely from growing concerns about the widespread impact of fake news on public opinion and events. Based on the research, it is noted that with massive growth of online communication the potential of people to deceive through computer mediated communication has also grown. Such deception can have a deleterious impact and far-reaching results, also causing a political upheaval in world politics. The topic of fake news has not only received public attention but has also drawn increasing attention from the academic community. We present a novel solution to detect disinformation, which is a subcategory of fake information, from variegated sources. Focusing on developing a strong computational model, we strive to capture the general features of the text embodying disinformation. Using a hybrid of content-based and feedback-based approaches, the model predicts the degree of disinformation concentrated in a text.

# Background

The usage and meaning of the term fake news has evolved over time. Google Trends Analysis of the term by Alcott and Gentzkow revealed a sudden burst in popularity around the time of the 2016 US Presidential Election. A study in 2017 revealed that fake news was widely shared during the three months prior to the election with 30 million total Facebook shares of 115 known pro-Trump fake stories and 7.6 million shares of 41 known pro-Clinton fake stories. Another example is major rise in tumult in 2011 which noted the false report of bankrupt of United Airlines s parent company in 2009 caused the stock price to drop by as much as 76% in a matter of minutes. The Fake news term has evolved and become synonymous with the spread of false information. Fake news has generally been defined as “a news article that is intentionally and verifiably false” or “information presented as a news story that is factually incorrect and designed to deceive the consumer into believing it is true".

The intent of deception, and do not capture the broader scope of the term based on its current usage. This definition allows us to capture the different types of fake news identified in which can be differentiated by the means employed to falsify information, such as fabricated content (completely false), misleading content (misleading use of information to frame an issue), imposter content (genuine sources impersonated with false sources), manipulated content (genuine information or imagery manipulated to deceive), false connection (headlines, visuals or captions that do not support the content), and false context (genuine content shared with false contextual information). The definition also allows us to include different types of fake news identified by their motive or intent, such as malicious intent (to hurt or disrepute), profit (for financial gain by increasing views), influence (to manipulate public opinion), sow discord (to create disorder and confusion), passion (to promote ideological biases), amusement (individual entertainment) [126]. We can also subdivide false information by intent as misinformation and disinformation. Misinformation refers to unintentionally spread false information which can be a result of misrepresentation or misunderstanding stemming from cognitive biases or lack of understanding or attention; and disinformation refers to false information created and spread specifically with the intention to deceive [54]. Another type of information that might be closely connected to fake news is satire - satire presents stories as news that might be factually incorrect, but the intent is not to deceive but rather to call out, ridicule, or expose behaviour that is shameful, corrupt, or otherwise “bad". This factor can have a detrimental fallacious impact in all aspect of our life hence requires a robust and scalable approach to vanquish the cause of this issue.

*Flow of Document: (Table with Section Name and Description) (On own)*

|  |  |
| --- | --- |
| Section | Description |
| Section 2 | Software Project Management Plan (SPMS) |
| Section 3 | Software Requirement Specifications (SRS) |
| Section 4 | Software Design Description (SDD) |
| Section 5 | Software Test Document (STD) |

# Sections Common to all Documents

This will contain information about the contents of the common sections present in all the documents.

For instance, Cover Page (Present in Every Document)

So, what all will be present on a cover Page should be mentioned and described in Short.

Other potential Common Sections – References/Terminologies/Appendix/Definitions/Acronyms

# Software Project Management Plan (SPMP)

## Introduction

### Project Overview

Disinformation detection system is a system for detecting disinformation in fake news. It is a web based application that enables the users from field of information security, information law and also normal users to determine the disinformation in the news articles and provides awareness so as to prevent getting fallacious information about the given topic Disinformation identification System mainly works to identify the fabricated content in the news and provide adaptive based approach to adapt itself and thereby enabling itself to a state as to determine with confidence that the given information generated from this source is of high chance being fake using machine learning .We are following an hybrid based approach which is and ensemble and feedback based approach by analyzing transitive relations and two level ensemble approach so as the domain in determining fake news can work on variety of news as an input .We focus on providing accurate results as of proving scrapping data from various sources so as to provide wider category for classification.

The purpose of this document is to present a detailed description of the Application. It will explain the purpose and features of the system along with its interfaces, the constraints under which it must operate and how the system will react to external stimuli. This document is intended for both the users and the developers of the system. The main purpose of the application is to gather all information from various news websites or from various social media platforms needed, in background so as to generate reports from our strong and robust machine learning models so as to provide diminutive overhead to the end users by providing the result what they need. The system also provides a NoSQL Database to store the information and add the news sources or articles which are detected fake inside the database so as the user can be presented with a graphical analytical user interface about fake sources or articles detected by the application. This is a single all in one application to the end user, which prevents the user to surf over the internet to guess whether a information is true or false, so that the end user can come to single platform to identify whether a given information is fake or correct by providing most accurate and relevant results .The Application is on web based platform on various OS support and utilize the power of AWS cloud for faster processing using EC2 instances and AWS Sagemaker for faster training of our machine learning models. It can be used by a broad spectrum of people in field of Information Security, Information Law, Normal Users, Business Stakeholders to manage for personal use or even for their organization. SPMP document is created to list the intended audience and provides suggestions for the same. This document is mainly for both developers and project manager to determine the procedure in the creation of further phases of the application and determine a road map or work flow for the same

### Project Deliverables

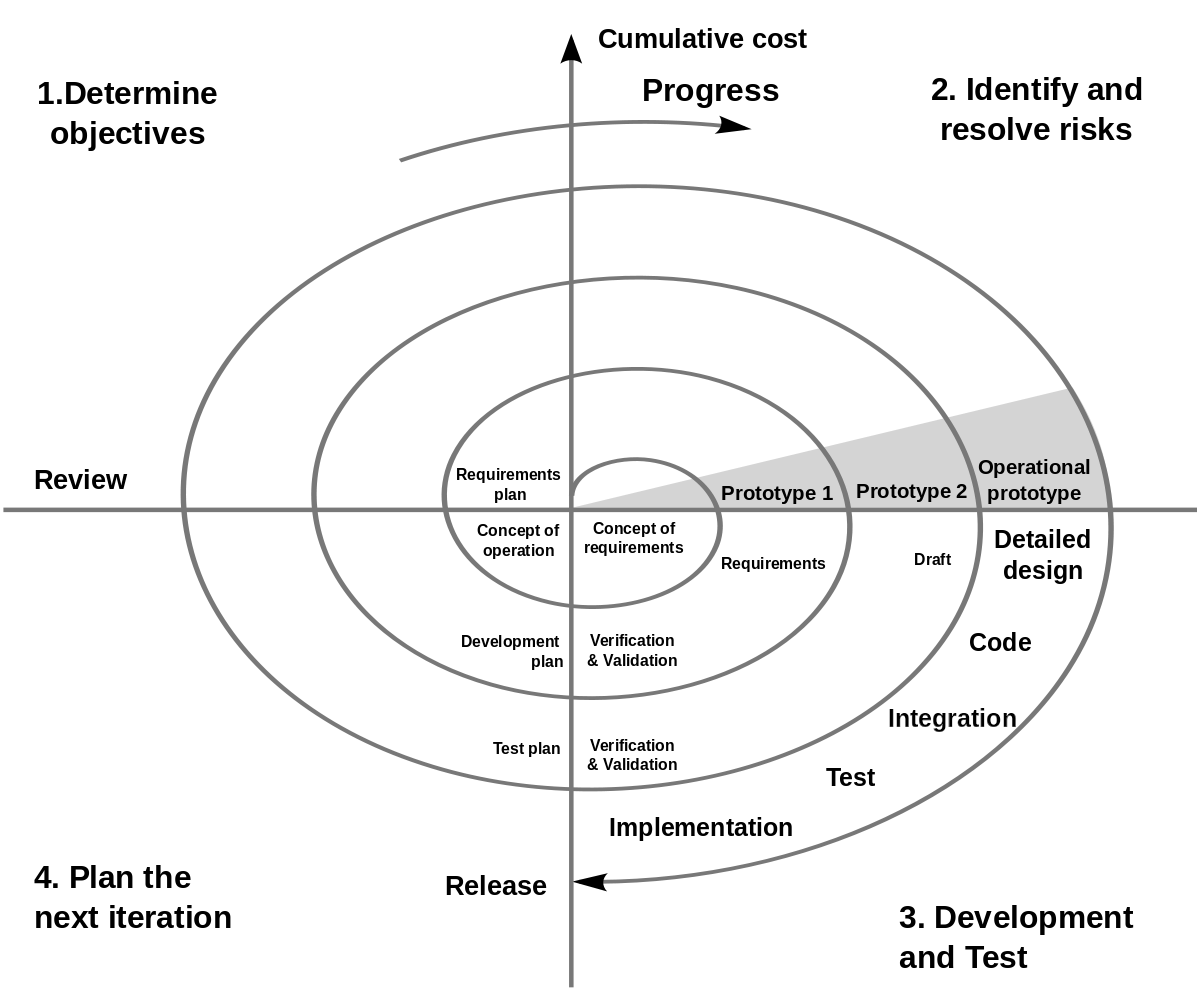
* Computation model
* Auto generated analysis report
* Interface for a user to interact with the model

Intelligent disinformation detection system will be a cloud-based solution that will be deployed on AWS/Azure server. Additionally, for lower latency and we are making use of elastic load balancer on cloud so as the end-user never gets any delay in request processing and corresponding result display.

## Project Organization

### Software Process Model

Intelligent Model for detecting disinformation in Fake news is a project that demonstrates its need in the current scenario and also provides features which are more streamlined and efficient approach than before. In order to achieve this the application needs to approved by the new kinds of trends in sources and strategies for fake news generation. For the same, their inputs are taken at the highest level of seriousness. We use the Spiral Software Process Model in order to keep a constant sync in process of development along scalability and software maintenance.



The SPIRAL process model used for Disinformation in fake news will consist of mainly 2 to 3 iterations. Each iteration will consist of Research, Planning, Development, Review and Deployment phases. The Research phase will consist of the Literature Review and Requirement survey and analysis. Followed by the Planning phase which will include the designing of System Architecture and Documentation. Then comes the Development phase which include some research on development tools and the actual development of the software. The penultimate phase is the review phase where testing of software is done, and based on the issues the system architecture is re planned or the implementation of the software is improved. After this the cycle is repeated and the Deployment stage is reached when the software is working as expected and there are no issues generated during the testing phase.

### Roles and Responsibilities

* + - Project Manager – Vedang Parasnis
      * Description - Project Manager is responsible for the timely execution and completion of the project. She will work with all the group members and will see that every group member is performing his/her task. She will communicate with the Faculty and inform him/her about the development of the project. She can schedule group meetings to look over the development of the project.
    - Developer – Hritik Jaiswal, Heet Sakaria, Vedang Parasnis
      * Description - Developer will code for applications and programs for backend processing systems to build a working project as proposed by the manager and team. Once the core of the software is developed, the software is passed on to the next Team member that is to the Tester.
    - Designer – Hritik Jaiswal
      * Description - Designer deals with the look and feel of the software. Designer’s task will be to work hand-in-hand with the developer and help him with creative styling ideas, improving frameworks to make the project more user-friendly.
    - Tester – Vedang Parasnis, Heet Sakaria
      * Description - Tester’s role will be to perform checks on the services provided by

the software, to see if they are functioning properly or are bugged for a given

condition.

* + - Analyst – Heet Sakaria, Hritik Jaiswal
      * Description - Project analyst is responsible for managing the development of project through special research, data analysis, and data collection to facilitate strategic decision-making.

### Tools and Techniques

For the development of the product following tools will be used.

* + - The front-end of the web application is developed using HTML / Bootstrap CSS framework.
    - TypeScript for front-end development
    - Express Js which is the backend web application framework for NodeJS and Flask will be used for back-end development and serve as an application server
    - Python Scrapy and Beautiful soup is used for web scrapping, News API is used for information gathering
    - Firebase cloud Storage will be used as the database for storage of application data.
    - Visual Studio IDE / Sublime Text, Jupyter Notebook, Google Colab will be used for developing the application and testing of machine learning models.
    - Aws EC2 Instances will be used for faster computation of application server request
    - AWS Sagemaker will be used for faster training of Machine Learning models on the cloud
    - StarUML will be used for creating UML diagrams.

## Project Management Plan

### Tasks

#### Documentation

Requirement Analysis/Writing SRS

1. Description:

To get all the requirements from the client and after proper analysis, forming the final SRS document.

1. Deliverables and Milestones:

* Listing all the requirements.
* Performing proper Requirement Analysis on it.
* Forming the SRS document.
* After some revisions finalizing the SRS.

1. Resources Needed

* MS Word

1. Dependencies and Constraints
   * None
2. Risks and Contingencies

* Only risk here is if any requirement is missed out. Which can be solved by communicating with client on every stage of the development of the project.

Designing System Architecture

1. Description

Designing the system architecture of the project which include on deciding the file structure, database to be used, language to be used for development and communication between all the components.

1. Deliverables and Milestones

* Designing Conceptual Architecture.
* Designing Detailed logical Architecture.
* Verifying internal design of each component and interaction among components.
* Finalizing the System Architecture Design.

1. Resources Needed

* UML Design Software

1. Dependencies and Constraints

* Requirements must be clear before designing the Architecture.

1. Risks and Contingencies
   * Frameworks used should have a continuous long-term support and receive security updates

#### Development

Computation model

1. Description  
   The algorithm is the chief element of the project which analyses and classifies the text. All other components revolve around this central piece.
2. Deliverables and Milestones

* Compilation of a relevant dataset
* Defining a model
* Evaluation of the model

1. Resources Needed

* Python
* Google Colab
* Jupyter Notebook

1. Dependencies and Constraints

* Dependency on the underlying libraries
* Data dependency

1. Risks and Contingencies
   * False positives and False negatives

Auto generated analysis report

1. Description

The report contains detailed information about the output of the model.

1. Deliverables and Milestones

* Development of a reliable model
* Listing parameters that the report covers
* Development of a script to auto-generate the report

1. Resources Needed

* Python
* ReactJS

1. Dependencies and Constraints

* Dependency on the model’s outcome

1. Risks and Contingencies
   * Error in model outcome

Interface for a user to interact with the model

1. Description

User interface that will help users interact and understand the model.

1. Deliverables and Milestones

* Selection of an Interface type
* Design of the interface
* Development of components
* Interfacing it with the model

1. Resources Needed

* Redis
* React JS

1. Dependencies and Constraints

* Responsive to the user’s environment

1. Risks and Contingencies
   * None

Software Testing

1. Description

Testing the software developed at each stage of development.

1. Deliverables and Milestones

* Performing Lighthouse testing for checking performance of software.
* Implementing Continuous Integration to maintain that code change does not break testing
* Extensive Selenium based tests to keep track of UI/UX actions working correctly even under load
* Creating final test report after final testing of software.

1. Resources Needed
   * TravisCI
   * Lighthouse
   * Sucuri/Pentest-tools
   * Selenium
2. Dependencies and Constraints
   * None
3. Risks and Contingencies
   * None

Software Deployment

1. Description

Deploying the application on an online website provider / google chrome store, with a domain name, and made available around the clock, publicly for website and associated rules for the.

1. Deliverables and Milestones

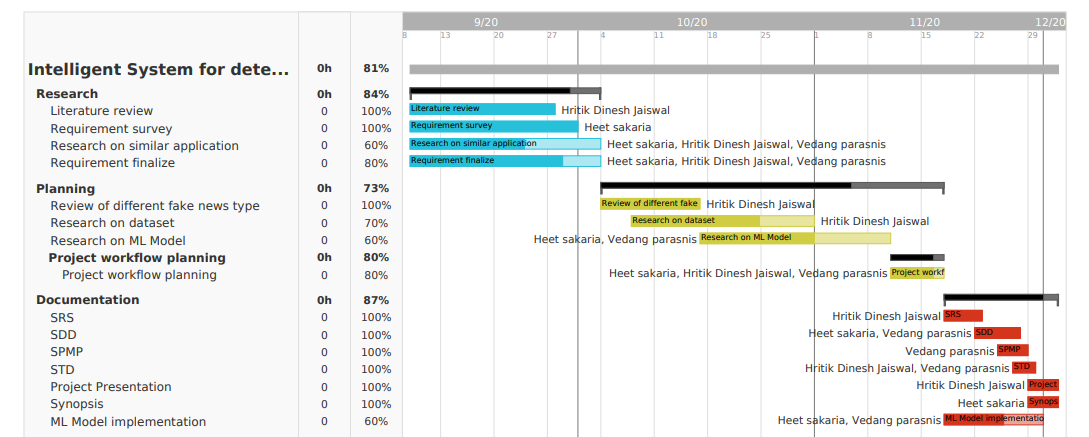
* Online Hosting with necessary measures

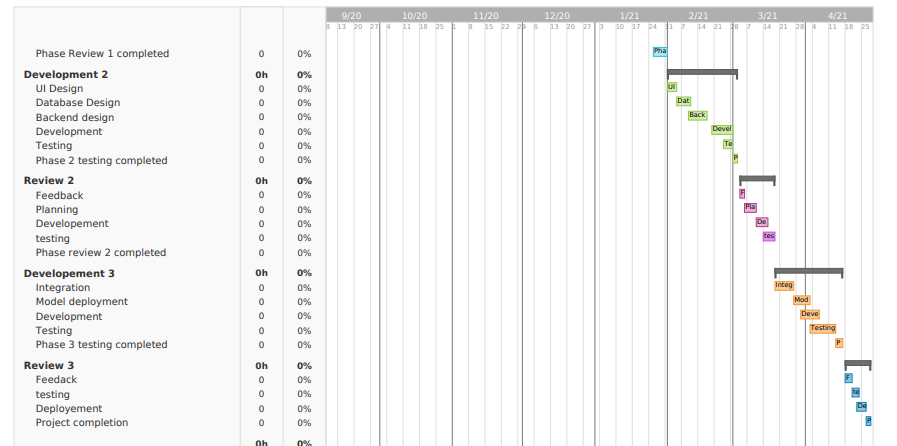
1. Resources Needed
   * AWS/Azure server
   * SSL certificate
   * Domain name
2. Dependencies and Constraints
   * AWS/Azure
3. Risks and Contingencies
   * Required domain name isn’t available: Solution isn’t available for this; a new different domain name will have to be sought and used.

### Assignments

|  |  |  |
| --- | --- | --- |
| Sr. No. | Task Name | Member Associated |
| 1 | Requirement Analysis/Writing SRS | Vedang Parasnis, Heet Sakaria |
| 2 | Designing System Architecture | Hritik Jaiswal , Vedang Parasnis |
| 3 | Articles Scrapping | Vedang Parasnis |
| 4 | Computational Model (Machine Learning Model) | Heet Sakaria, Hritik Jaiswal |
| 5 | Cloud Instance, Database Management | Vedang Parasnis , Heet Sakaria |
| 6 | Software Testing | Vedang Parasnis |
| 7 | Software Deployment | Heet Sakaria , Hritik Jaiswal , Vedang Parasnis |

### Timetable





# Software Requirements Specifications (SRS)

## INTRODUCTION

### Product Overview

Disinformation detection system is a system for detecting disinformation in fake news. It is s a web based application that enables the users from field of information security, information law and also normal users to determine the disinformation in the news articles and provides awareness so as to prevent getting fallacious information about the given topic Disinformation identification System mainly works to identify the fabricated content in the news and provide adaptive based approach to adapt itself and thereby enabling itself to a state as to determine with confidence that the given information generated from this source is of high chance being fake using machine learning .We are following an hybrid based approach which is and ensemble and feedback based approach by analyzing transitive relations and two level ensemble approach so as the domain in determining fake news can work on variety of news as an input .We focus on providing accurate results as of proving scrapping data from various sources so as to provide wider category for classification. For Working in real-time we are scraping articles from various renowned sources and thereby passing this raw data to our machine Learning models based on source generated by data and prediction provided as a analytical report to End User.

The application is to be developed in the format of a progressive-web-application (PWA). We also strive for providing a browser extension to monitor and provide necessary feedback and warnings about some malicious or fake data sources which are surfed by end user so the end user gets most accurate and relevant information.

## SPECIFIC REQUIREMENTS

### External Interface Requirements

#### Hardware Interfaces

* 1 AWS server to host the system
  + 1+ Core
  + 1GB+ RAM
  + 8GB RAM+ SSD/HDD Storage

#### Communication Protocols

* HTTPS Protocol for serving website
* TCP/IP for Redis/MongoDB

### Software Product Features

1. Fake News Detection for various categories of disinformation
2. Adaptable to new fake news sources and the corresponding information from them
3. Robust and novel hybrid machine learning model to work on various categories of data
4. Raw data collection from various websites with faster scrapping and low latency in processing of data.

**Software System Attributes**

Explain how our Software satisfies these attributes.

**Balancing timeliness v/s detection accuracy**: Early detection and mitigation are critical goals of any effective system. However, the available information for detection increases as time progresses, with only the content of the article being available at the start, followed by increasing user responses as propagation continues. Most existing methods either rely on content only or on user responses only, or do not utilize responses incrementally. Detection systems must aim to utilize incrementally available information to trade-off confidence in detection accuracy v/s timeliness of the detection and mitigation effort.

**Prioritization and cost-effectiveness**. The ability to optimally decide which contents to factcheck at what time, can equip the system in providing better responses by being able to quickly remove false information that can have a potentially larger and faster impact than those that might have a negligible or slower impact if allowed to propagate further in time. Also, human involvement in fact-checking increases not only the delay but also the cost of intervention, which necessitates the need for prioritization of information to manually fact-check, until reliable automated methods can be sought.

**Robustness, scalability and interpretability**. The high stakes and consequences of fake news necessitates the need for reliability in detection. Mistakenly removing true information from the platform, or not detecting and removing potentially viral false information would become problematic in practice. To move from manual and semi-automated solutions to fully automated ones will not be possible without robust and also interpretable predictions.

**Accuracy & Performance.** Most ML work reports on algorithm accuracy (often precision and recall), i.e., how “correct” the output is compared to reality. Further work looks more broadly at algorithm performance i.e including comparisons of performance in specific contexts.

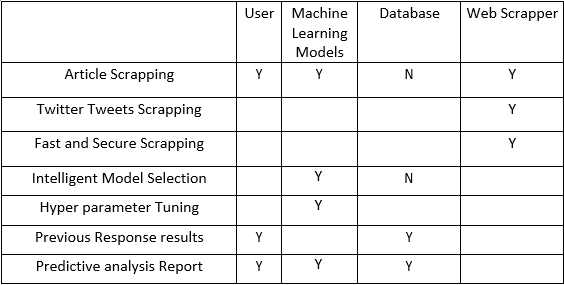
# Software Design Description (SDD)

## INTRODUCTION

### Design Overview

Disinformation detection system is a system for detecting disinformation in fake news. It is s a web based application that enables the users from field of information security, information law and also normal users to determine the disinformation in the news articles and provides awareness so as to prevent getting fallacious information about the given topic Disinformation identification System mainly works to identify the fabricated content in the news and provide adaptive based approach to adapt itself and thereby enabling itself to a state as to determine with confidence that the given information generated from this source is of high chance being fake using machine learning .We are following a hybrid based approach which is and ensemble and feedback-based approach by analysing transitive relations and two-level ensemble approach so as the domain in determining fake news can work on variety of news as an input .We focus on providing accurate results as of proving scrapping data from various sources so as to provide wider category for classification. For Working in real time, we are scraping articles from various renowned sources and thereby passing this raw data to our machine Learning models based on source generated by data and prediction provided as a analytical report to End User.

### Requirements Traceability Matrix



## SYSTEM ARCHITECTURAL DESIGN

## The chosen system architecture for this project is MVC. MVC is a software design pattern for developing a web application. MVT Structure consists of three parts namely Model, View and Controller.

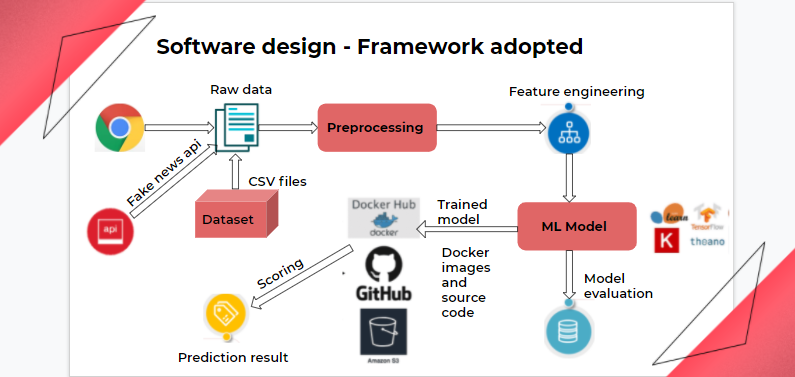
## 

## Model acts as the interface of the data. It is responsible for maintaining data. It is the logical data structure behind the entire application and is represented by a database (generally NoSQL database such as MongoDB, Redis).

## 

## The View is the user interface what we see in our browser when we render a website. 3 It is represented by HTML/CSS/JavaScript/TypeScript). A template consists of static parts of the desired HTML output as well as some special syntax describing how dynamic content will be inserted.

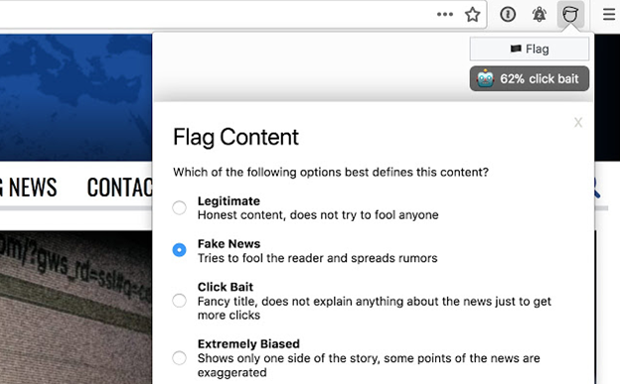
### **Chosen System Architecture**



## System Interface Description

### User Interface

### The application will be web based, so the first screen of the application shows the input field where user will pass the raw input data into it and then it will click on the submit button and after that the raw data will be fed to model which will give the output and then resultant output is shown with help of radio button and there is feedback button for user to cross verify whether the predicted output and actual output (actual output in a sense that, we are assuming the user is rational which mostly giving true feedback which is based on some analysis) are matching or not, we will collect this data to generate some new insight, which help us to find performance measure of model.



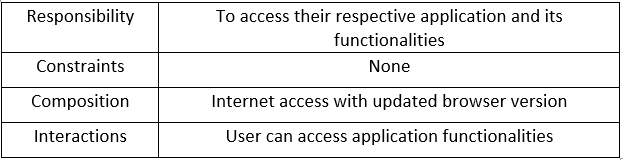
**Figure 2: user interface**

### 1.1.2 Software Interface

The software interface follows the Model-View-Controller (MVC) model for making and modelling data objects. The chosen operating system is Windows for its best support and user-friendliness. The interface must be managed via an external cloud instance, Application programming interface like AWS EC2 which controls AWS Sagemaker for our robust machine Learning models training and prediction. The application server has an ability to connect to an in-memory database or NoSQL database to store analytical results generated about the associate fake news sources from different sources. The interface is to be created using ReactJs as front-end library, ExpressJs/Flask for application server which is intern managed by our AWS EC2 instances in production environment we are trying to make a progressive web app, to provide faster production delivery leveraging the ability of cloud.

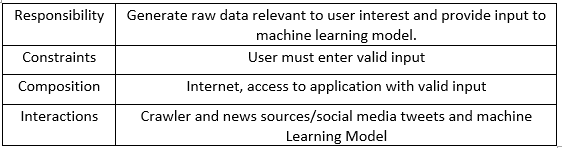
# Detailed Description of Components

## Component 1: User Interface

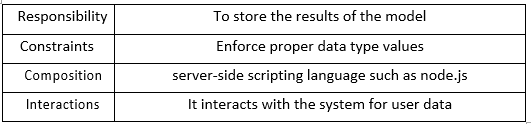


|  |  |
| --- | --- |
|  |  |

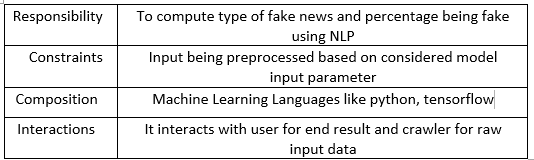
## Component 2: Web Crawler



## Component 3: Database



## Component 4: Machine Learning Model



# System Test Document (STD)

## INTRODUCTION

### System Overview

Disinformation detection system is a system for detecting disinformation in fake news. It is s a web based application that enables the users from field of information security, information law and also normal users to determine the disinformation in the news articles and provides awareness so as to prevent getting fallacious information about the given topic Disinformation identification System mainly works to identify the fabricated content in the news and provide adaptive based approach to adapt itself and thereby enabling itself to a state as to determine with confidence that the given information generated from this source is of high chance being fake using machine learning .We are following a hybrid-based approach which is and ensemble and feedback-based approach by analysing transitive relations and two-level ensemble approach so as the domain in determining fake news can work on variety of news as an input. We focus on providing accurate results as of proving scrapping data from various sources so as to provide wider category for classification. For Working in real-time we are scraping articles from various renowned sources and thereby passing this raw data to our machine Learning models based on source information is scrapped. prediction provided as an analytical report to End User.

**Main Modules under consideration:**

The Software Application is divided into 3 modules which need to be tested individually including the following:

* + 1. Articles Scrapping
       1. Scrapped News Articles
       2. Social Media tweets
    2. End Users
       1. Client End Articles validation
       2. Analytical Reports
       3. Non-Bot Validation

Browser

Extension Integration

* + 1. Database Instances

|  |  |
| --- | --- |
|  | Parity of news articles |
| Partially or Jargon Entries |

### Test Approach

As every aspect of the application is to be tested, we first decide on the types of testing to be performed. When a certain type of testing is to be performed, test cases are generated for that type of testing. And the generated test cases are performed on the application and based on the test report the web application will be further improved.

## TEST PLAN

# 1 Test Plan

The application will be tested as per the completion of the given modules.

|  |  |  |  |
| --- | --- | --- | --- |
| Sr No. | Deliverable Name | Author | Reviewer |
| 1 | Software Testing Document | Hritik Jaiswal | Project Manager |
| 2 | Test Plan | Vedang Parasnis | Project Manager |
| 3 | Functional Test Cases | Hritik Jaiswal | Developer |
| 4 | Logging Defects | Test Team | Test Lead |
| 5 | Daily/Weekly Status | Test Team | Project Manager/Test Lead |
| 6 | Test Closure Report | Heet Sakaria | Project Manager |

## 1.1 Features to be Tested

Software modules that need to be tested are as follows:

* + 1. Relevancy of Scrapped Articles:
    2. High Website traffic Scenarios
    3. Classification report generated by the machine learning model:
    4. Lower API Request Latency based on location
    5. Display of Analytical Reports:
    6. Testing against anti crawling Websites:
    7. Client Application Interfaces:
    8. Continuous Integration for new beta releases:

## 1.2 Features not to be Tested

Features that need not be tested are as follows:

* + 1. Cross platform support
    2. Security

## 1.3 Testing Tools and Environment

As soon as the development is completed for individual modules, the testers are to be ready with the test cases and start with its execution. Manual approach for creating the software will be applied. Each Module is developed separately using HTML, CSS,

Typescript, Python, Postman, Curl, RedisCLI or MongoCLI Aws SDK as its developing language and framework along with additional APIs and requirements as and when required. This developed application must be integrated together using version control software. For manual testing, Tester creates different environments and solves the issues using various manual as well as automated tools available for Python and Flask.

# 2 Test Cases

## 2.1 Relevancy of Scrapped Articles: TC-1

### 2.1.1 Purpose

To identify and generate most relevant articles or tweets based on user area of interest and pass the corresponding input to the machine learning model and generate corresponding analytical report.

2.1.2 Inputs

News Article or relevant News Headline Tags provided by End User.

2.1.3 Expected Outputs & Pass/Fail criteria

After Input by the user and corresponding news articles which are generated will be validated to the corresponding user news or headlines tags and then be processed in the machine learning model to generate corresponding analytical reports else error message will be displayed on screen s

### 2.1.4 Test Procedure

This module needs to be tested manually.

**Constraints:** A user can provide news tags or articles only in English Language.

2.2 High Website traffic Scensarios: TC-2

### 2.2.1 Purpose

To maintain faster and less delay to the end user once news tag requested by the user and then to analyse how the application perform after scrapping of copious number of articles and then providing result to end User in analytical format without request timeout even in high traffic by using proper load balancing

### 2.2.2 Inputs

News Article or relevant News Headline Tags provided by End User.

2.2.3 Expected Outputs & Pass/Fail criteria

Appropriate action will be performed for request response timeouts and if timeout display the corresponding server timeout issues else provide the result which is intended to user

2.2.4 Test Procedure

This module needs to be tested manually or through automated record and play tools using Postman API testing tool.

**Constraints:** The user must not be a Bot and have verified captcha for application domain.

## 2.3 Classification report generated by the machine learning model: TC-3

### 2.3.1 Purpose

To test the same model with hyper parameter tuning and generate a precision recall and F1 score values.

2.3.2 Inputs

Hyper parameters

2.3.3 Expected Outputs & Pass/Fail criteria

If current model generates better accuracy than the preciously trained model on different parameters then current parameters is saved else discarded

### 

2.3.4 Test Procedure

This module needs to be tested manually or through automated record.

**Constraints:** None

2.4 Lower API Request Latency based on location: TC-4

### 2.4.1 Purpose

To manage and provide best possible AWS EC2 Instance having proximity to the client so as to reduce latency in response and faster response to the client.

2.4.2 Inputs None.

2.4.3 Expected Outputs & Pass/Fail criteria

Appropriate action will be performed for lower latency regions if lower latency then we would provide a lower latency option so as to provide a faster response else try to select the closest proximity instance to the client

2.4.4 Test Procedure

This module needs to be tested manually or through automated record and play tools.

**Constraints:** The Admin must have multiple parallel EC2 Instances available.

## 2.5 Display of Analytical Reports: TC-5

### 2.5.1 Purpose

To provide best possible and relevant output to the end user in a user friendly analytical form Also, the generated reports provide correct and specific results based on user query.

2.5.2 Inputs

Corresponding news articles tags and headlines

2.5.3 Expected Outputs & Pass/Fail criteria

If proper response is generated during the process corresponding analytical report will be displayed else error report will be displayed to the end user.

2.5.4 Test Procedure

This module needs to be tested manually or through automated testing tools.

**Constraints:** The client must have provided proper search query to the server

## 2.6 Testing against anti crawling Websites: TC-6

### 2.6.1 Purpose

How crawler performs on anti-crawling website sources and what action is to be taken for such websites

2.6.2 Inputs

None

2.6.3 Expected Outputs & Pass/Fail criteria

None

2.6.4 Test Procedure

This module needs to be tested manually or through automated robot tools

**Constraints:** None.

## 2.7 Client Application Interfaces: TC-7

### 2.7.1 Purpose

To provide error free and user-friendly application environment so as the user can be able to get all the necessary reports without any point of failure.

2.7.2 Inputs

None

2.7.3 Expected Outputs & Pass/Fail criteria

The analytical reports will be displayed on success else corresponding error actions will be shown to the end user

2.7.4 Test Procedure

This module needs to be tested manually or through automated testing tools.

**Constraints:** The End-user must have latest browser version installed in the system.

## 2.8 Continuous Integration for new beta releases: TC-8

### 2.8.1 Purpose

To manage future releases of application so as to maintain contiguous integration for upcoming release and provide bug free and better application support to the newer version as compared to earlier version.

2.8.2 Inputs

None

2.8.3 Expected Outputs & Pass/Fail criteria

Further application releases are well maintained bug free and also backward compatible

2.8.4 Test Procedure

This module needs to be tested manually or through automated tools. We utilize CI/CD tools like TravisCI or CircleCI using version controller like GitHub.

**Constraints:** None

# Conclusions and Further Work

The topic of disinformation detection is a fairly convoluted task and becomes more difficult with advances in social media and information transfer over digital media. We strive the best for providing the most accurate information thereby covering a wide set of useful data in our model. Hence, we have provided a novel approach for disinformation detection in all categories which are associated with disinformation considering unlabelled samples. In future we mainly focus on how to provide better intervention-based technique thereby optimizing the mitigation strategies so the user can most near to the accurate information. For our future work we mainly focus on advancement of landscape for following computational solutions

**Dynamic knowledge bases**. Although we can design many features which can potentially help us determine the truthfulness of a news article, the truthfulness is still ultimately defined by the statements it makes. The greatest challenge in developing automated fact-checking methods is the construction of dynamic knowledge bases, which can be regularly and automatically updated to reflect the changes occurring in a fast-paced world

**New intervention strategies**. For the design of useful intervention strategies for different environments, the most important question that needs to be answered is which environmental factors are most conducive to the spread of fake news. Studying and characterizing the relationship between user actions and utilities at the microscopic level of the individual, and the macroscopic impact in different networked environments, will be essential for explaining the spread of fake news and finding the best intervention strategies suited to that environment

**Datasets for intent detection**. Current datasets generally provide binary labels of information as fake or true. However, a more fine-grained classification of information by intent might be especially beneficial in identifying truly fake news from closely related information such as satire and opinion news. In the list of fake websites maintained by, there is a type label allowing up to 3 types for each website with tags such as political, satire, or biased. Some recent works have considered classification of fake vs satire news and fake vs hyper partisan news. Hence, we believe that this is an important direction for the future.

# REFERENCES

1. <https://www.aclweb.org/anthology/S17-2081.pdf>
2. <https://sci-hub.do/downloads/2020-08-29/b1/sangamnerkar2020.pdf?rand=5f9bdbe7b7dbd?download=true>
3. <https://dl.acm.org/doi/pdf/10.1145/3132847.3132877>
4. <https://arxiv.org/pdf/1703.09398.pdf>
5. [A Survey of Fake News: Fundamental Theories, Detection Methods, and Opportunities (arxiv.org)](https://arxiv.org/pdf/1812.00315.pdf)