TRIBHUVAN UNIVERSITY
INSTITUTE OF ENGINEERING

Khwopa College of Engineering
Libali, Bhaktapur
Department of Computer Engineering

A REPORT ON
Fake News Detection Using Machine Learning

Submitted in partial fulfillment of the requirements for the degree

BACHELOR OF COMPUTER ENGINEERING

Submitted by

Ronish Shrestha	KCE075BCT032
Roshan Shrestha	KCE075BCT033
Rubin Baidhya	KCE075BCT034
Sairush Tamang	KCE075BCT036

Under the Supervision of
Er. Anku Jaisawal

Khwopa College of Engineering
Libali, Bhaktapur
April 18, 2022

Certificate of Approval

This is to certify that this minor project work entitled **"Fake News Detection Using Machine Learning"** submitted by Ronish Shrestha(KCE075BCT032), Roshan Shrestha(KCE075BCT033), Rubin Baidhya(KCE075BCT034) and Sairush Tamang(KCE075BCT036) has been examined and accepted as the partial fulfillment of the requirements for degree of Bachelor in Computer Engineering.

.....
Er. Santosh Giri
External Examiner
Assistant Professor
Dept. of Electronics and Computer Engineering
IOE, Pulchowk Campus

.....
Er. Anku Jaiswal
Project Supervisor
Assistant Professor
IOE, Pulchowk Campus

.....
Er. Dinesh Gothe
Head of Department
Department of Computer Engineering
Khwopa College of Engineering

Copyright

The author has agreed that the library, Khwopa College of Engineering may make this report freely available for inspection. Moreover, the author has agreed that permission for the extensive copying of this project report for scholarly purpose may be granted by supervisor who supervised the project work recorded here in or, in absence the Head of The Department where in the project report was done. It is understood that the recognition will be given to the author of the report and to Department of Computer Engineering, KhCE in any use of the material of this project report. Copying or publication or other use of this report for financial gain without approval of the department and author's written permission is prohibited. Request for the permission to copy or to make any other use of material in this report in whole or in part should be addressed to:

Head of Department
Department of Computer Engineering
Khwopa College of Engineering
Liwali, Bhaktapur
Nepal

Acknowledgement

We would like to express our deepest gratitude to our supervisor of the project, **Er. Anku Jaisawal** for her support and guidance throughout our project. The completion of the project would not have been possible if it were not for her guidance.
We would also like to acknowledge our teachers **Er. Dinesh Gothe**, **Er. Bindu Bhandari** and **Er. Sajit Pyakurel** for guiding and encouraging us to initiate our project. We are deeply indebted to them for sharing us with their experiences that have helped us in many ways throughout our Bachelor's program.
We would also like tho express our gratitude towards **Er. Prakash Chandra Prasad** for the help and guidance he provided during our project. Also thanks to **Er. Prabesh Shrestha** and **Er. Siddhant Sharma** for their valuable suggestion.
Finally, we would like to thank each and everyone who have directly and indirectly assisted us in starting our project.

Ronish Shrestha	KCE075BCT032
Roshan Shrestha	KCE075BCT033
Rubin Baidhya	KCE075BCT034
Sairush Tamang	KCE075BCT036

Abstract

Fake news not only misleads the readers but also creates a devastating impact so it is a major concern. Thus the increasing fake news globally tends to encroach human right and public safety. A person needs to have vast knowledge about the current issue to discover whether a news is real or fake. So, we have implemented a system to detect the fake news using Natural Language Processing and Long Short Term Memory and Support Vector Machine. Based on the headlines and the content of the news the proposed system recognizes whether the news is fake or real in short amount of time. We did a number of tests on both SVM and LSTM models and we found that SVM model gave a slightly higher accuracy with 99.46 percent to LSTM model's accuracy of 98.64 percent. Also, we could only work on one specific domain of news viz. politics mainly from the USA.

Keywords:*Fake News, Natural language processing, Long Short Term Memory, Neural Networks, Support Vector Machine*

Contents

Certificate of Approval i
Copyright ii
Acknowledgement iii
Abstract iv
Contents vii
List of Tables viii
List of Figures ix
List of Abbreviation x

1 Introduction **1**
1.1 Background 1
1.2 Motivation 1
1.3 Problem Definition 1
1.4 Objectives 2
1.5 Scope and Applications 2

2 Literature Review **3**

3 Requirement Analysis **8**
3.1 Software Requirements 8
3.1.1 Python 8
3.1.1.1 Keras 8
3.1.1.2 TensorFlow 8
3.1.1.3 Beautiful Soup 8
3.2 Functional Requirements 8
3.3 Non-Functional Requirements 9
3.3.1 Reliability 9
3.3.2 Maintainability 9
3.3.3 Performance 9
3.3.4 Accuracy 9
3.4 System Requirement 9

4 Feasibility Study **10**
4.1 Time Feasibility 10
4.2 Space Feasibility 10
4.3 Technical Feasibility 10
4.4 Economic Feasibility 10
4.5 Operational Feasibility 10

5 Methodology **11**
5.1 Agile Method as Software Development Model 11
5.1.1 Iteration 1 11
5.1.1.1 Planning 11
5.1.1.2 Development 12
5.1.1.3 Testing 12

5.1.1.4 Evaluation . . . 12
5.1.2 Iteration 2 . . . 12
5.1.2.1 Planning . . . 12
5.1.2.2 Development . . . 12
5.1.2.3 Testing . . . 12
5.1.2.4 Evaluation . . . 13
5.1.3 Iteration 3 . . . 13
5.1.3.1 Planning . . . 13
5.1.3.2 Development . . . 13
5.1.3.3 Testing . . . 13
5.1.3.4 Evaluation . . . 13
5.2 Task Workflow . . . 14

6 System (or Project) Design and Architecture 15
6.1 Structure of System . . . 15
6.2 System Block . . . 16
6.3 Use Case Diagram . . . 17
6.4 Sequence Diagram . . . 18
6.5 Data Flow Diagrams . . . 19
6.5.1 Level-0 DFD . . . 19
6.5.2 Level-1 DFD . . . 20
6.6 Activity Diagram . . . 20
6.7 Dataset . . . 22
6.7.1 Dataset Collection . . . 22
6.7.2 Dataset Preprocessing . . . 23
6.8 Long Short Term Memory . . . 23
6.8.1 One hot for Embedding layers . . . 24
6.8.2 Padding embedded documents . . . 24
6.8.3 Creating LSTM model . . . 24
6.9 Support Vector Machine . . . 24
6.9.1 Tuning Parameter of SVM . . . 25
6.9.1.1 Kernel . . . 25
6.9.1.2 Regularization . . . 25
6.9.1.3 Gamma . . . 25
6.9.1.4 Margin . . . 26
6.9.2 Identifying the right hyper-plane . . . 26
6.10 TF-IDF Vectorizer . . . 28
6.10.1 Terminologies . . . 28
6.11 Web Scraping . . . 29
6.12 Detection Evaluation . . . 30
6.12.1 Confusion matrix . . . 30

7 Result and Discussion 32
7.1 Desktop Outcome . . . 32
7.2 Unit Tests . . . 32
7.2.1 Models Unit . . . 32
7.2.2 Scrapping Unit . . . 34
7.2.3 Integration Testing . . . 34

7.3 Model prediction result 35

8 Conclusion and Future Enhancements **39**

Bibliography **40**

Appendix **41**

A Assistance for Running the Project 41

A.1 Fake News Detection Using Machine Learning 41

A.1.1 Clone Git Project 41

A.1.2 Installation Required 41

A.1.3 Running the Project 41

B Snapshot 42

B.1 Article Tag News 42

B.2 Div Tag News 42

B.3 True news word cloud 43

B.4 Fake news word cloud 43

B.5 Outcome of desktop site 44

B.6 Unit test for Scraping 45

List of Tables

2.1 Literature Review Matrix 5

6.1 True and Fake News dataset 22
6.2 Splitting dataset for SVM Classifier 22
6.3 Splitting dataset for LSTM Classifier 22

7.1 SVM model testing requirements 32
7.2 SVM model test cases 33
7.3 LSTM model testing requirements 33
7.4 LSTM model test cases 33
7.5 Scrapping testing requirements 34
7.6 Scrapping test cases 34
7.7 Integration testing scenario and requirements 34
7.8 Integration Testing 35
7.9 LSTM classification report 35
7.10 SVM classification report 35

List of Figures

5.1 Software Development Approach 11

6.1 Client Server Architecture 15
6.2 System Block Diagram 16
6.3 Use Case Diagram 17
6.4 Sequence Diagram 18
6.5 Level-0 DFD 19
6.6 Level-1 DFD 20
6.7 Activity Diagram 21
6.8 Bar graph showing fake and real news count 22
6.9 Feature extraction 23
6.10 High Regularization 25
6.11 Low Regularization 25
6.12 High Gamma 26
6.13 Low Gamma 26
6.14 Good margin 26
6.15 Bad margin 26
6.16 Hyperplane in SVM 27
6.17 Non-linear hyperplane 27
6.18 Flowchart of Web-Scraping 29
6.19 Confusion Matrix 30

7.1 LSTM confusion matrix 36
7.2 SVM confusion matrix 36
7.3 LSTM loss plot 37
7.4 LSTM accuracy plot 37

List of Abbreviation

Abbreviations	**Meaning**
AI	Artificial Intelligence
CNN	Convolutional Neural Network
CSS	Cascading Style Sheets
DL	Deep Learning
GRU	Gated Recurrent Unit
HTML	Hypertext Markup Language
IDS	Intrusion Detection System
LSTM	Long Short Term Memory
ML	Machine Learning
NLP	Natural Language Processing
NN	Neural Network
PCA	Principal Component Analysis
RBF	Radial Basis Function
RNN	Recurrent Nerual Network
SDLC	Software Development Life Cycle
SVM	Support Vector Machine
TF-IDF	Term Frequency Inverse Document Frequency

Chapter 1
Introduction

1.1 Background

With tremendous amount of data flowing over internet, most of them are found fake but still people prefer the internet source as it is the easiest means. Unfortunately the open source has become the platform to release news without effective supervision. Fake news are delivered cleverly in a traditional method that implies to be truthful but contain misleading information. By utilizing different philosophies, realizing the correctness of the news is a fascinating issue [1]. As defined by the New York Times, fake news is "made up stories written to deceive" [2].
As a result, we have created a tool to assist consumers in determining whether a piece of news is true or fake.
In this project, we use machine learning algorithm to predict and classify news and give information about it regarding its classification.

1.2 Motivation

In this century where news reporter and news channel growth is exponential, it is really difficult to determine a piece of news as real or fake coming from different sources. Not only for public but also for experts in certain field feel difficult to classify a news. So as the fake news deceives us to believe it as true, the biggest challenge in today's online news portal is that we fail to understand the news and its sentiment behind its writing. Obviously, without any experts saying about the news we are always in confusion to believe whether a news is real or fake. And also, it is not possible to give credibility to some online news without any further knowledge about it.

1.3 Problem Definition

The daily life of people is hugely influenced by information. Most of the people plan their daily routine based on the information they obtain from various sources. This shows our reliance on news and also presents the importance of real news on daily basis.
Although a simple piece of news carry huge importance in every social, economic, political and environmental fields, the credibility of that piece of news can cause great turmoil in the world. A true news will not create any negative impacts on our society but if fake news is spread across the society and country then it will very badly impact the society, country and the people.
People consume news from various medias and there are presently all sorts of news available in online sources. Some people use such online platforms to spread

various fake news for their own gains. This creates confusion among the people and create havoc among the people. People themselves may not be able to classify whether the news is real or fake and blindly follow such news. This is a great problem that we are facing nowadays. But now we can classify a news as real or fake using artificial intelligence and deep learning. However most of the current existing fake news detection system only classify news based on the headline and the content. Since news can also be an image or video, these format are not processed in existing systems.
Integrating AI can help us identify fake news and filter them before they create confusion among the people.

1.4 Objectives

The major objective of the project "Fake News Detection" are:

- To reduce the effort and time to detect fake news.
- To classify news as fake or real as it is a major problem.

1.5 Scope and Applications

Fake news creates confusion and disbelief among the people. They may appear in different forms such as click-baits, misleading headlines, fabricated content and biased news.
So implementing fake news detecting techniques might help in minimizing the spreading of such disinformation and also protect people's right to true information.
But the advancing technology is posing more and more challenges in the detection of fake news, as the influence of social media on people grows.. Similarly the detection of false news is becoming more difficult, as news is becoming more elegantly titled as if it were true.
Similarly, the presently available machine learning and deep learning techniques still face many challenges to detect the fake news as the content are being planned in such a way that they resemble truth.
Also, the irregular and unpredictable data may lead the prediction model to make mistakes during detection.

Chapter 2

Literature Review

Fake reviews and fake news both are responsible for spreading misconceptions and disbeliefs among the people, according to the article "Detecting opinion spams and fake news using text classification" [3]. The article has also emphasized that detecting fake news is even harder than the detection of fake reviews. It has categorized fake news into 3 groups. The first being false news, second as fake satire news and lastly poorly written news articles. Here, the authors have introduced models for detecting both fake news and fake reviews. The authors have also mentioned them being the first to detect both the fake news and fake reviews at the same time. The authors have mentioned that they had used TF-IDF as the feature extraction model for their datasets. The prediction model was designed using six different classifiers by checking their respective accuracy and efficiency.

The article [4] has proposed a method to measure the user's credibility in social media. The authors have proposed CredRank algorithm for measuring user credibility in social media. This algorithm is supposed to detect user's online behavior and measure the credibility of user accordingly. Here, Abbasi, Mohammad-Ali and Liu, Huan have taken into consideration the Arab Spring movements in social media sites, and shown how the users can spread misinformation in the society.

The research article [5] has shown the use of Naive Bayes classifier for fake news detection. The authors used the data from facebook news posts for implementing the Naive Bayes classifier and also confirmed that they got decent accuracy of around 74% using the classifier. They even stressed on the point that using a complex model can result in even greater accuracy and efficiency.

Gilda [6] has explored the usage of neural networking algorithms for fake news detection in his paper. The author experimented his data in various classifiers and calculated their respective accuracies. The author has also stressed that Stochastic Gradient Descent model identifies non-credible sources with higher accuracy with respect to other models.

Bahad et. al. [7] has presented bi-directional LSTM model for fake news detection as a superior model as compared to other models namely uni-directional LSTM, CNN and vanilla RNN. The authors have also expressed that the traditional text mining and machine learning techniques are inadequate in handling the large and complex datasets available in present days. The proposed model analyzes the relationship between the article's titles and it's content to detect whether the article is real or fake. The proposed RNN-LSTM trains the data iteratively for improving the accuracy. It was concluded by the authors that the LSTM-RNN classifier is suitable long-range semantic dependency based classification. Also, the model was very capable in working with both balanced and imbalanced high dimensional

news data set.

In article [1] various Machine Learning models like Naive Bayes, K nearest neighbour, Decision Tree, Random Forest and Deep Learning networks(CNN, LSTM, GRU) are used. He has also explored and used features like n-gram, TF-IDF. All these models were used and their corresponding accuracies were pointed out due to which evaluation becomes easier. On comparing all these models, CNN LSTM using together was found to have the highest accuracy of 97.3%.

Rohit kumar kaliyar et. al [2] has proposed FNDNet, a deep convolutional neural network for fake news detection. To analyze FNDNet, GloVe is used as pre-trained word embedding which is unidirectional in training. They have also used various machine learning as well as deep learning algorithms for classification. With the accuracy of 98.3% the authors have stated that their proposed model provides "state-of-art" result for predicting fake news. Despite the high performance of their classifiers, they have realized that there is still room for improvement. The authors have concluded that their further plan is to use hybrid approach for fake news classification as it provides broader generalization which is lacking in their model. FNDNet with Glove has proved to be the highest accuracy yielding algorithm amongst the other papers so far. So it is convincing that it needs to be implemented in the field of fake news detection and classification.

Oshikawa, Ray. Jing, Qian. Wang, William. Y. in their survey [8] has revealed the importance of automatic fake news detection. This survey for sure has emphasized on the related problems regarding fake news detection and has proposed a method using NLP. To compare results they have mainly focused on three datasets: LIAR, FEVER, and FAKENEWSNET. The authors have suggested to examine if handcrafted features may be combined with neural network, as well as the suitable use of non-textual data and extending the way of verification with contents.

The authors in [9] have proposed a fake news stance detection model that classifies the news articles with stance labels of either agree, disagree, unrelated, or discuss. The proposed model uses Principal Component Analysis (PCA) and Chi-square with CNN-LSTM. PCA and Chi-square together perform component level analysis and obtain the reduced feature set which are passed to CNN-LSTM model. This model produces a promising accuracy by scoring 97.8%. However it is seen that while preserving the high performance of classifiers by using Dimensionality reduction techniques(for removing redundant features, noisy and irrelevant data) it can reduce the number of features. So in order to boost the performance of their model, the authors have mentioned that their future work will analyze different textual features, tree-based learning for simple approach and validate the performance of their proposed model for larger datasets. Amongst all other papers reviewed, this model gives the second highest accuracy for fake news detection.

Their research in [10] offered an experimental examination into detecting fake news on Bangla social media, as this subject still required a great deal of attention. To distinguish Bangla fake news, we used two supervised machine learning algorithms, Support Vector Machine(SVM) and Multinomial Naive Bayes(MNB) classifiers,

throughout this research work. The feature extraction algorithms were Term Frequency- Inverse Document Frequency(TF-IDF) Vectorizer and CountVectorizer. According on the polarity of the related post, their proposed algorithm detects fake news. Finally, their research showed that SVM with linear kernel has a 96.64 percent accuracy, compared to 93.32 percent for MNB.

The authors in [11] employed classification approaches as Support Vector Machine(SVM), Naive Bayes, and Passive Aggressive Classifier in this model. Their model's output had 95.05 percent accuracy when employing feature exrtraction approaches like term Term Frequency- Inverse Document Frequency(TF-IDF) Vectorizer and a SVM as a classifier.

Table 2.1: Literature Review Matrix

Author	**Title**	**Features**	**Accuracy**
Kaliyar,Rohit.K.	Fake News Detection Using A Deep Neural Network	It examines which model will provide the most accurate results and classify the news as fake or real	Naive Bayes: 90.19% Decision Tree: 75% Random Forest: 72% K Nearest Neighbour: 55% CNN and LSTM: 97.3%
Kaliyar, Rohit.K. Goswami, Anurag. Narang, Pratik. Sinha, Soumendu.	A deep convolutional neural network for fake news detection	1) It uses pre-trained word embedding models 2) It uses GloVe enabled deep convolutional-based approach.	98.3 % accuracy using FNDNet with Glove
Oshikawa, Ray. Jing, Qian., William. Y.	A Survey on Natural Language Processing for Fake News Detection	1) It provides an overview of research efforts for fake news detection. 2) It analyzes how fake news detection is aligned with existing NLP tasks	Not available

Author	**Title**	**Features**	**Accuracy**
Ahmed, Hadeer Traore, Issa Saad, Sherif	Detecting opinion spams and fake news using text classification	1) Detects both the fake reviews and fake news 2) Classification of fake news and fake reviews	SVM: 84% LVSM: 94% KNN: 83% DT: 89% SGD: 89% LR: 89%
Abbasi, Mohammad-Ali Liu, Huan	Measuring User Credibility in Social Media	1) Detect user's online behaviour and measure the credibility of user accordingly. 2) Shows how the users can spread misinformation in the society.	Not available
Umer, Muhammad. Imtiaz, Zainab. Ullah, Saleem. Mehmood, Arif. Choi, Gui. S. On, Byung-Won.	Fake News Stance Detection Using Deep Learning Architecture(CNN-LSTM)	1) Automatically classifies the news articles with stance labels of either agree, disagree, unrelated, or discuss	1)CNN-LSTM without Preprocessing: 78.4% 2)CNN-LSTM with Preprocessing: 93% 3)CNN-LSTM with chi-square: 95.2% 4) CNN-LSTM with PCA: 97.8%
Bahad, Pritika Saxena, Preeti Kamal, Raj	Fake News Detection using Bi-directional LSTM-Recurrent Neural Network	1) Analyzes relationship between the article's titles and it's content 2) Trains data iteratively to increase accuracy	RNN-LSTM: 98%

Author	**Title**	**Features**	**Accuracy**
Granik, Mykhailo Mesyura, Volodymyr	Fake news detection using Naive Bayes Classifier	1) Use Bag-of-words feature to identify spam 2) Much simplew AI algorithm to implement	Naive Bayes: 74%
Gilda, Shlok	Evaluating Machine Learning Algorithms for Fake News Detection	Comparison among various models	Bounded Decision Trees: 67.6% Gradient Boosting: 65.7% Random Forests:64.8% Stohastic Gradient Descent: 65.7% SVM: 73.6%

Chapter 3

Requirement Analysis

3.1 Software Requirements

3.1.1 Python

Python is an interpreted high-level general-purpose programming language. Its design emphasizes code readability by using a lot of indentation. The language features and object-oriented approach of python are designed to help programmers write clear, logical code for both small and large-scale projects.

3.1.1.1 Keras

Keras is a python-based deep learning framework that is open source. Francois Chollet, a Google artificial intelligence researcher, came up with the idea. Keras adheres to best practices for lowering cognitive load, such as providing uniform and simple APIs, limiting the number of user activities required for typical use cases, and providing clear and actionable error messages. It comes with a lot of documentation and developer instructions. Keras is presently used by Google, Square, Netflix, Huawei, and Uber, among others.

3.1.1.2 TensorFlow

TensorFlow is a machine learning and artificial intelligence software library that is free and open-source. It can be used for a variety of applications, but it focuses on deep neural network training and training and inference. TensorFlow is compatible with a wide range of programming languages, including Python, Javascript, C++, and Java. This adaptability lends itself to a wide range of applications in a variety of industries.

3.1.1.3 Beautiful Soup

Beautiful Soup is a python library that allows you to parse HTML and XML files (including having malformed markup, i.e. non-closed tags, so names after tag soup). It builds a parse tree for parsed pages, which may be used to extract data from HTML and is used for web scraping. It is available for python 2.7 and 3.

3.2 Functional Requirements

The main goal of this project is to determine whether or not the contents of a news story presented are genuine. So, followings can be listed as the functional requirements of our Fake News Detection Project:

- The system should be adaptable and elaborated from the point of view of the users.
- The system should allow users to provide both url and article as input.
- It should extract the news article from online news portal.
- The system should be able to classify the news provided in the form of both article or url as true or fake.

3.3 Non-Functional Requirements

The following non-functional requirements should be met by our project on fake news identification.

3.3.1 Reliability

After completion, the project will be extremely dependable. The system will be able to run without fail in a specific environment.

3.3.2 Maintainability

The maintenance of the finished system will be easily feasible. Also we can easily maintain our system for future enhancements. By adding additional examples to the database with which we train our system, the system's ability to identify can be easily upgraded.

3.3.3 Performance

Performance of the system depends on how fast the system gives the result. After its completion, the system will be able to detect fake news in a very less time.

3.3.4 Accuracy

The technology should be able to tell whether the news being read is phony or trustworthy. The system will be capable of accurately detecting fake news.

3.4 System Requirement

- Working PC with internet connection

Chapter 4

Feasibility Study

A feasibility study is an examination of a project's relevant variables, such as economic, technical, legal, and scheduling, with the goal of successfully completing the project. It considers the advantages and disadvantages of a project before committing time and money to it. It determines whether or not the proposed system is incompatible with the requirements. For instance, operation-wise, spacing, time, software legality, and so on. A project feasibility study is a report that evaluates the frames of analysis for a certain project in great depth.

4.1 Time Feasibility

This project's development cycle is estimated to take 5 months or perhaps longer. We want to complete the assignment as soon as possible in order to make it more practical while still meeting the deadline.

4.2 Space Feasibility

We're attempting to keep this project as minimal as possible in order to save disk space. To reduce space complexity, it will be preferable to use fewer libraries and frameworks.

4.3 Technical Feasibility

All of the data sets that are required for our project are easily available in the internet. We can complete the job using currently available tools. Open sourced and freely available tools will be adequate to meet the system's technical feasibility requirements.

4.4 Economic Feasibility

We want to employ the majority of open-source tools, libraries, frameworks, and data sets available on the internet. As a result, there is no expense. There is no investment of any budget in this project

4.5 Operational Feasibility

We are proposing to develop a web-based fake news detection, mainly focused on more user-friendly and informative environment where user will able to use the system in more convenient fashion.

Chapter 5

Methodology

5.1 Agile Method as Software Development Model

Implementing in website was quite a challenge which can be made easy by dividing project with a small section and developing it to a fully functioned project. Since the computing power of the computer may vary so implementing this whole project was a challenging part. Visualizing the defect and where our project was going was necessary therefore we used Agile model as Software Development model as it facilitated us to start our project from a small section and launch it.

Figure 5.1: Software Development Approach

5.1.1 Iteration 1

After some research and planning, we started our first iteration towards project development. The first Iteration consisted of following tasks:

5.1.1.1 Planning

In planning phase, we made plannings on how to move forward with our project. We discussed about the necessary requirements and documented them afterwards.

a. Data collection

b. Data preprocessing

5.1.1.2 Development

In development phase, the implementation process of the project was initiated. The dataset that we collected required pre-processing which was done in development phase.

5.1.1.3 Testing

Here, the testing process was done of the task done in the development phase. Checking for errors was done if any for the data processing part and the processed data was evaluated in following phase.

5.1.1.4 Evaluation

After the testing phase was completed, evaluation process was initiated. The overall work done by the members was properly evaluated. This phase evaluated the overall quality of the task done by the members and further modifications if needed, was discussed.

5.1.2 Iteration 2

After collecting and preprocessing data, we required to develop the model.

5.1.2.1 Planning

Developing the model required research and study on different models and algorithm for text classification.

a. Studying various machine learning models.

b. Developing the machine learning model.

c. Developing Web Scraper

5.1.2.2 Development

After proper plannings and analysis, the development process was started. Here, we have developed two classification models: SVM and LSTM. The required parameters were properly initialized and implemented in the models. Similarly, in this phase, a web scraper was also developed and included in the models.

5.1.2.3 Testing

The developed models were tested using different inputs. The web scraper was also tested here and checked if the news was being properly scraped or not. The test results were taken into consideration and were evaluated in the following phase.

5.1.2.4 Evaluation

After the models and scraper were tested, they were evaluated by the members. The models were checked if they were classifying the input accurately or not. Here, the web scraper was found to be working as our need.But for the models, since we found that the accuracy was not satisfactory, the models were further taken into next iteration.

5.1.3 Iteration 3

At the beginning of third iteration we had both the model as well as web scraper required for the text classification and parsing html. So the improvement required in the model was performed in third iteration along with developing some other task.

5.1.3.1 Planning

Our whole plan was to develop the fully functional project as a website. We planned the following tasks which is to be performed in our final iteration.

1. Developing Frontend
2. Developing Backend
3. Integrating SVM and LSTM model
4. Final Testing

5.1.3.2 Development

After completion of frontend, we began to work on backend and itegrating the models and the web scraper. So the final development phase consisted of developing the frontend and backend and integrating all the required units required for the project

5.1.3.3 Testing

The overall developed outcome was tested after their integration. Here the testing was done for frontend, backend, scraper and the models after they were integrated. For the testing, the user gave input in the frontend/UI and the processing was carried out in the backend. The trained models were saved in the backend and the testing was performed with those models. Altogether the working of frontend, backend, scraper and the models was tested here.

5.1.3.4 Evaluation

After the testing was completed, the members evaluated if the models were giving satisfactory result or not. Also the quality and simplicity of the UI was evaluated here. With the completion of the third iteration, a fully functional system was developed.

5.2 Task Workflow

Each tasks were completed and stored in github.
The workflow of the tasks is as follows:

a. Create corresponding branch in github and fetch the branch in local

b. Checkout to the branch and do assigned tasks there

c. Push the task done to github

Chapter 6

System (or Project) Design and Architecture

6.1 Structure of System

The basic structure of our application is very simple. It is a client-server based architecture as shown in figure 6.1.

Figure 6.1: Client Server Architecture

The user delivers the piece of news or the url of any portal website's news article to the server from the frontend application, and the server simply returns if the news article is fake or true. The model developed using python and its libraries are kept on the server or in any cloud-based storage facilities.

6.2 System Block

Figure 6.2: System Block Diagram

6.3 Use Case Diagram

The system was trained using the dataset. The user can make use of the system's facility to determine whether the news he is reading is fake or trustworthy.

Figure 6.3: Use Case Diagram

6.4 Sequence Diagram

Sequence diagram depicts the objects involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of scenario. The following shows the sequence diagram.

Figure 6.4: Sequence Diagram

6.5 Data Flow Diagrams

The Data Flow Diagram shows the flow of the data between the subsystem of the inference system.

6.5.1 Level-0 DFD

The level-0 data flow diagram of the inference system is shown below:

Figure 6.5: Level-0 DFD

6.5.2 Level-1 DFD

The level-1 data flow diagram of the inference system is shown below:

Figure 6.6: Level-1 DFD

6.6 Activity Diagram

An activity diagram visually presents a series of actions or flow of control in a system similar to a flowchart or a data flow diagram. The following shows the activity diagram of our system:

Figure 6.7: Activity Diagram

6.7 Dataset

6.7.1 Dataset Collection

Dataset for Fake News Classification was taken from kaggle. It contains true and fake data separately which is then combined later and shuffled. After combining it requires labeling the column and separating into training and testing set. This dataset was splitted into 90% train and 10% test for SVM Classifier whereas 80% train and 20% test for LSTM.

Table 6.1: True and Fake News dataset

Type	Number of News
True News	21417
Fake News	23460
Combined News	44877

Table 6.2: Splitting dataset for SVM Classifier

Type	Train size	Test size
True News	90%	10%

Table 6.3: Splitting dataset for LSTM Classifier

Type	Train size	Test size
True News	80%	20%

Figure 6.8: Bar graph showing fake and real news count

Above bar graph shows the number of true and fake data that we had collected from Kaggle. Our dataset contains the title, text, subject and label as its columns where title consists of the headline of the news, text contains the news in its elaborated form, subject consists of the domain of the news articles and the label shows whether the news is true or fake. Our dataset contained news from different countries such as US news, world news and so on. But the overall dataset contains news from politics domain only.

About the dataset, most of the fake news is surrounded by Election news and about Trump since the dataset was created around the time of the US elections 2020. Similarly, most of the news is from Reuters.

6.7.2 Dataset Preprocessing

The dataset which we collected from kaggle consist of the fake and genuine news articles. These datasets also had duplicate news article. So, we preprocessed the datasets to remove all the duplicate news. A generic preprocessing function was developed that removes punctuation and non-letter characters(stop words) from each news article of the datasets.And, the datasets items were converted to lowered case.
Lemmatization looks beyond word reduction and considers a language's full vocabulary to apply a morphological analysis to words. The lemma of 'was' is 'be' and the lemma of 'mice' is 'mouse'. Also, Multiple variants of the same word are lemmatized to return to their common root, such as 'coming' and 'comes' into 'come'. Lemmatization is typically seen as much more informative than simple stemming. This will assist us in reducing the size of the actual data by deleting any unnecessary information. After preprocessing these datasets, feature extraction was performed which is shown in B.3, B.4.

Figure 6.9: Feature extraction

In above figure, more feature extraction from the dataset is done. The features extracted are polarity, review length and word count.Here, the polarity measures the sentiment of the news, Review length shows the length of the news and word count shows the number of words in the news.
From the feature extraction, we know that most of the polarity is neutral. This means there is both bad news and happy news. Similarly, the word count is between 0-1000 and the length of the news is between 0-5000. Also, we can see that a few news has near 10000 words.

6.8 Long Short Term Memory

Long short-term memory is a deep learning architecture that uses an artificial recurrent neural network. There are feedback connections in the LSTM. It can handle not only individual data points(such as photos), but also unsegmented,

connected handwriting identification, speech recognition, and anomaly detection in network traffic or IDS can all benefit from LSTM.

6.8.1 One hot for Embedding layers

In one hot encoding, we convert each categorical data or value into a new categorical column and assign a binary value of 1 or 0 to those columns. One hot encoding makes our training data more useful and expressive.
Here, we are one hot encoding the sentences in the corpus for embedding layers. We need a vocabulary size since in one-hot encoding, the words in sentences will take the index from the vocabulary size. Here, we have set the vocabulary size of 10000 looking at our dataset.

6.8.2 Padding embedded documents

The neural networks require the data or the input to be of same shape and size. Since during the preprocessing of our dataset, the sentences are not of the same length. So in this case, padding is necessary. Also, we are going to 'pre' pad our sentences with the common length of 5000. Due to this, the sentences will be of equal length.

6.8.3 Creating LSTM model

In our model, the first layer will be the embedding layer where we have the vocabulary size, vector features and sentence length. After this, we have used or added a 70% dropout layer to prevent our model from overfitting. Similarly, our model consists of a Bi-directional LSTM layer of 100 neurons. Using the bidirectional LSTM model, we will be able to predict the future use of words as there is backward information passed from the other RNN layer in reverse. This cannot be done using the unidirectional LSTM.
In the final layer, we have used the sigmoid activation function since we need to predict the probability as an output. We have used the adam optimizer in our model. This optimizer is computationally efficient. Furthermore, we have used binary cross-entropy as the loss function since our data consists of only two label and we also have only two outputs.

6.9 Support Vector Machine

SVM stands for Support Vector Machine. SVM is a supervised machine learning algorithm that is commonly used for classification and regression challenges. Common applications of the SVM algorithm are Intrusion Detection System, Handwriting Recognition, Protein Structure Prediction, Detecting Steganography in digital images, etc.
Support Vector Machine algorithm is mainly used to solve classification problems. Support vectors are nothing but the coordinates of each data item. Support Vector Machine is a frontier that differentiates two classes using hyper-plane.

6.9.1 Tuning Parameter of SVM

6.9.1.1 Kernel

In linear SVM, learning of hyperplane is done by changing the issue with some linear algebra. This is where the kernel enters the picture.
For linear kernel the equation for prediction for a new input using the dot product between the input (x) and each support vector (xi) is calculated as follows:

$$f(x) = B(0) + sum(ai * (x, xi)) \tag{6.1}$$

This is an equation that involves calculating the inner products of a new input vector (x) with all support vectors in training data. The coefficients B0 and ai (for each input) must be estimated from the training data by the learning algorithm.

6.9.1.2 Regularization

In Python's sklearn module, the Regularization parameter (also known as the C parameter) tells the SVM optimization how much you want to prevent misclassifying each training example.
For large values of C, the optimization will choose a smaller-margin hyperplane if it does a better job of accurately classifying all of the training points. A very small value of C, on the other hand, will encourage the optimizer to hunt for a larger-margin separating hyperplane, even if it misclassifies more points. We find misclassification due to lower regularization value. Higher value leads to results like right one.
In our SVM model we have taken 1.5 as value of c. So this value of c is lower and we have chosen the lower value of c because lower the value, it allows misclassification and thus the model will be able to learn under the noisy data and discover a generalizable plane.

Figure 6.10: High Regularization
Source: Medium

Figure 6.11: Low Regularization
Source: Medium

6.9.1.3 Gamma

The gamma parameter determines how far a single training example's influence extends, with low values indicating 'far' and large values indicating 'near.' In other words, when the gamma is low, locations that are far away from a conceivable separation line are taken into account. Whereas a high gamma indicates that the

points closest to a feasible line are taken into account in the calculation.
This parameter is used only when we use the Gaussian RBF kernel. Since we used only linear kernel we require only c parameter

Figure 6.12: High Gamma
Source: Medium

Figure 6.13: Low Gamma
Source: Medium

6.9.1.4 Margin

A margin is a separation of line to the closest class points.A good margin is one where this separation is larger for both the classes. Images below gives to visual example of good and bad margin. A good margin allows the points to be in their respective classes without crossing to other class. If the margin is larger in between the classes, then it is considered a good margin, a smaller margin is a bad margin.

Figure 6.14: Good margin
Source: Medium

Figure 6.15: Bad margin
Source: Medium

6.9.2 Identifying the right hyper-plane

SVM classifies with a 'Hyperplane'. Now this hyperplane can be a single line in case of a 2-dimensional data and can be a plane in 3-dimensional one.
The Margin is defined with the help of the Support Vectors. In our example, Yellow stars and Yellow circles are the Support Vectors defining the Margin. The better the gap, the better the classifier works. Hence support vectors play an important role in developing the classifier.

Every new data point in test data will be classified according to this Margin. If it

Figure 6.16: Hyperplane in SVM
Source: Towards Data Science

lies on the right side of it, it'll be classified as a Red circle otherwise as a Blue star.

The best thing about SVM is it can also classify non-linear data. In such cases SVM uses the 'Kernel', it uses a kernel function to map the non-linear data to higher dimensions so that it becomes linear and finds the decision boundary there. A Kernel function is always used by SVM, whether it is linear or non-linear data, but its main function comes into play when the data is inseparable in its current form. Here, the Kernel function adds dimensions to the problem for classification.

Figure 6.17: Non-linear hyperplane
Source: Towards Data Science

6.10 TF-IDF Vectorizer

Term Frequency Inverse Document Frequency is very common algorithm to transform text into a meaningful representation of numbers which is used to fit machine algorithm for prediction.

6.10.1 Terminologies

- **Term Frequency:** In document d, the frequency represents the number of instances of a given word t. For each specific term in the paper, there is an entry with the value being the term frequency.

$$tf(t,d) = \frac{count\ of\ t\ in\ d}{number\ of\ words\ in\ d} \tag{6.2}$$

- **Document Frequency:** This tests the meaning of the text, which is very similar to TF, in the whole corpus collection. The only difference is that in document d, TF is the frequency counter for a term t, while df is the number of occurrences in the document set N of the term t. In other words, the number of papers in which the word is present is DF.

$$df(t) = occurrence\ of\ t\ in\ documents \tag{6.3}$$

- **Inverse Document Frequency:** Mainly, it tests how relevant the word is. Since tf considers all terms equally significant, it is therefore not only possible to use the term frequencies to measure the weight of the term in the paper. First, find the document frequency of a term t by counting the number of documents containing the term:

$$df(t) = N(t) \tag{6.4}$$

where
df(t) = Document frequency of a term t
N(t) = Number of documents containing the term t

Term frequency is the number of instances of a term in a single document only; although the frequency of the document is the number of separate documents in which the term appears, it depends on the entire corpus. Now let's look at the definition of the frequency of the inverse paper. The IDF of the word is the number of documents in the corpus separated by the frequency of the text.

$$df(t) = \frac{N}{N(t)} \tag{6.5}$$

The more common word is supposed to be considered less significant, but the element (most definite integers) seems too harsh. We then take the logarithm (with base 2) of the inverse frequency of the paper. So the if of the term t becomes:

$$idf(t) = log(\frac{N}{df(t)}) \tag{6.6}$$

- **Computation:** Usually, the tf-idf weight consists of two terms-
  1. Normalized Term Frequency (tf)
  2. Inverse Document Frequency (idf)

$$tf - idf(t, d) = tf(t, d) * idf(t) \tag{6.7}$$

6.11 Web Scraping

Figure 6.18: Flowchart of Web-Scraping

Web Scraping technique is applied in order to extract the article from the online news portal. By inspection of different online news portal, $< article >$ tag and $< div >$ tag are commonly used by different online news portal for publishing the news on their website as seen in B.1 B.2
So, we started by checking if the response is success and then we drive to check if the url of the online contains whether $< article >$ or $< div >$ tags. In the case of $< article >$ tag we simply extracted $< h1 >$ as the headline of the news and news was extracted from $< p >$ tags. But, in the case of $< div >$ tags, as we know the website may contain number of $< div >$ on their website. So, we only targeted to the div which has $< h1 >$ on it. And we extracted the news accordingly.

6.12 Detection Evaluation

Evaluation for output of Detection is done using F1 Score analysis. It is determined from confusion matrix as:

		Predicted	
		Negative	**Positive**
Actual	**Negative**	True Negative	False Positive
	Positive	False Negative	True Positive

Figure 6.19: Confusion Matrix

6.12.1 Confusion matrix

A confusion matrix is the matrix representation of prediction results. The matrix compares the actual target values with those predicted by the machine learning model.
Each prediction has 4 outcomes based on how predicted values match with the actual values:

- **True Positive(TP)**: Predicted True and True in reality
- **True Negative(TN)**: Predicted False and False in reality
- **False Positive(FP)**: Predicted True and False in reality
- **False Negative(FN)**: Predicted False and True in reality

The confusion matrix can be used to evaluate the performance of classification model through performance metrics such as Precision, Accuracy, Recall and F1-score.

- **Precision**: It is a measure of correctness that is achieved in true prediction. In simple words, it tells us how many predictions are actually positive out of all the total positive predicted.

$$Precision = \frac{TP}{TP + FP} \tag{6.8}$$

- **Recall**: It is a measure of actual observations which are predicted correctly, i.e. how many observations of positive class are actually predicted as positive.

$$Recall = \frac{TP}{TP + FN} \tag{6.9}$$

- **F1-Score**: The F1 score is a number between 0 and 1 and is the harmonic mean of precision and recall.

$$F1 - Score = \frac{2 * Precision * Recall}{Precision + Recall} \tag{6.10}$$

Chapter 7
Result and Discussion

7.1 Desktop Outcome

Initially, we designed the UI for our project B.5, and implemented accordingly. Smooth and consistent system design are what we expected after integrating models with our backend system. But, our actual outcome is some how inconsistent as we found some lagging in loading of the content in web and, in the processing of the article to mark the news article as true or fake. Keeping the lagging problem in mind, we made some minor changes to our backend and system performance was quite impressive than the earlier one.

For the prediction of the news article, we provide options if user wanted to test the section of the news article or to test whether the url of the website's news is true or fake.

7.2 Unit Tests

We performed several testing of our different units of the project after their development and before implementation. And, inspecting the tests, we performed necessary measures to solve the problems and errors aroused during the testing phase of different units.

7.2.1 Models Unit

Table 7.1: SVM model testing requirements

Test Scenario ID	Model-1
Test Case Description	Checking the performance of SVM model
Pre-requisitive	Developed SVM Classification model

Test execution steps:

A. Load the model

B. Give input

C. Predict

Table 7.2: SVM model test cases

S.N.	**Input**	**Expected Output**	**Actual Output**	**Result**
1	Give true news article	True output	True output	Pass
2	Give fake news article	Fake output	Fake output	Pass

Table 7.3: LSTM model testing requirements

Test Scenario ID	Model-2
Test Case Description	Checking the performance of the LSTM model
Pre-requisitive	Developed LSTM Classification model

Test execution steps:

A. Load the model

B. Give input

C. Predict

Table 7.4: LSTM model test cases

S.N.	**Input**	**Expected Output**	**Actual Output**	**Result**
1	Give true news article	True output	True output	Pass
2	Give fake news article	Fake output	Fake output	Pass

7.2.2 Scrapping Unit

Table 7.5: Scrapping testing requirements

Test Scenario ID	Scrapping-1
Test Case Description	Check if the contents of url is extracted
Pre-requisitive	Valid url of the website

Test execution steps:

A. Copying url

B. Insert the url

C. Run the scrapper

Table 7.6: Scrapping test cases

S.N.	**Input**	**Expected Output**	**Actual Output**	**Result**
1	Give URL of the website which has ¡article¿ tag	News Title and article	News Title and Article	Pass
2	Give URL of the website which has ¡div¿ tag and news content on ¡p¿ tag	News Title and article	News Title and Article	Pass
3	Give URL of the website which has ¡div¿ tag and news content on ¡div¿ tag	News Title and article	News Title Only	Fail

Web scraping is achieved using the Beautiful Soup library of the python. Web scraping is done to add the additional functionality of the system to allow user to test the url of the website contains true or fake news B.6.

7.2.3 Integration Testing

Table 7.7: Integration testing scenario and requirements

Test Scenario ID	Integration-1
Test Case Description	Check if the entire system works properly or not. Check if the user gets proper output for his input article
Pre-requisitive	All models should be tested and working correctly

Test execution steps:

A. Open website

B. Give input

C. Click predict button

D. Check prediction result

Table 7.8: Integration Testing

S.N.	**Input**	**Expected Output**	**Actual Output**	**Result**
1	Give news article from true dataset	True	True	Pass
2	Give news article from fake dataset	Fake	Fake	Pass
3	Give url of the online news portal	True	Fake	Fail

7.3 Model prediction result

Table 7.9: LSTM classification report

	Precision	**Recall**	**F1-Score**	**Support**
True	0.97	1.00	0.98	4317
Fake	1.00	0.97	0.99	4659
Accuracy			0.98	8976
Macro avg.	0.98	0.99	0.98	8976
Weighted avg.	0.99	0.98	0.98	8976

Table 7.10: SVM classification report

	Precision	**Recall**	**F1-Score**	**Support**
True	0.99	1.00	0.99	2097
Fake	1.00	0.99	0.99	2391
Accuracy			0.99	4488
Macro avg.	0.99	0.99	0.99	4488
Weighted avg.	0.99	0.99	0.99	4488

Figure 7.1: LSTM confusion matrix

Confusion matrix is implemented in the LSTM test datasets. Out of which, 4300 datasets which are fake have been predicted fake, 4539 data sets which are real have been predicted real, 17 data sets which are fake but, are predicted real and 120 datasets which are real but, are predicted fake.

Figure 7.2: SVM confusion matrix

Confusion matrix is implemented in the SVM test datasets. Out of which, 2089 data sets which are fake have been predicted fake, 2375 datasets which are real have been predicted real, 8 data sets which are fake but, are predicted real and 16 datasets which are real but, are predicted fake.

Figure 7.3: LSTM loss plot

The loss plot is showing the loss occuring during the training process of our LSTM model. We have used the binary cross entropy as our loss function. The loss function calculates the incurred loss across every data items during an epoch. This plot also helps us to know if our model is overfit or underfit or goodfit. Our training loss and testing loss plots are both decreasing and there is also a small gap between the plots. So, we can know that our plot is neither overfitting nor underfitting and it is a goodfit.

Figure 7.4: LSTM accuracy plot

The accuracy plot shows our model's performance on our dataset. From our accuracy plot, we can see that our model has a good performance. The curves for both train and test plots are rising and they are reaching to a stable point. Although there is a drastic deviation in the second epoch of the test plot, the plot is very stable for the remaining epochs.

Chapter 8

Conclusion and Future Enhancements

The implemented system presents another measure model and a bit by bit methodology for characterizing a bit of text (news) as valid or phony. The model we implemented uses Natural Language Processing, Web Scraping procedures for deciding the legitimacy of a news.
Here, we have implemented two classification models and although both of the models scored some respectable scores, we found out that the LSTM model need larger datasets for making better predictions whereas the SVM model can make predictions using lesser dataset size.
We have implemented the system considering past writing overviews on a similar point from ongoing years. Also taking into consideration about the current situation of the pandemic, where the requirement for credible data to arrive at the mass populace is a need, this thought gives an extraordinary chance to score and ascertain the realness of the news and help cook clients with a security net over phony news accessible over the web. This forms a good baseline for the amount of unverified news that floats around the various social media platforms. As we talk about the future enhancements that we could implement to our system, the following points come to mind:

1. We could use our own custom dataset to train our model.
2. We could expand our domain outside politics and include other domains such as sports, economics and so on.
3. We could use or make custom Nepali dataset for Nepali News Detection.
4. We could increase the accuracy for real world data as well.

Bibliography

[1] R. K. Kaliyar, "Fake news detection using a deep neural network," in *2018 4th International Conference on Computing Communication and Automation (ICCCA)*. IEEE, 2018, pp. 1–7.

[2] R. K. Kaliyar, A. Goswami, P. Narang, and S. Sinha, "Fndnet–a deep convolutional neural network for fake news detection," *Cognitive Systems Research*, vol. 61, pp. 32–44, 2020.

[3] H. Ahmed, I. Traore, and S. Saad, "Detecting opinion spams and fake news using text classification," *Security and Privacy*, vol. 1, no. 1, p. e9, 2018.

[4] M.-A. Abbasi and H. Liu, "Measuring user credibility in social media," in *Social Computing, Behavioral-Cultural Modeling and Prediction*, A. M. Greenberg, W. G. Kennedy, and N. D. Bos, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 441–448.

[5] M. Granik and V. Mesyura, "Fake news detection using naive bayes classifier," in *2017 IEEE First Ukraine Conference on Electrical and Computer Engineering (UKRCON)*, 2017, pp. 900–903.

[6] S. Gilda, "Notice of violation of ieee publication principles: Evaluating machine learning algorithms for fake news detection," in *2017 IEEE 15th Student Conference on Research and Development (SCOReD)*, 2017, pp. 110–115.

[7] P. Bahad, P. Saxena, and R. Kamal, "Fake news detection using bi-directional lstm-recurrent neural network," *Procedia Computer Science*, vol. 165, pp. 74–82, 2019, 2nd International Conference on Recent Trends in Advanced Computing ICRTAC -DISRUP - TIV INNOVATION , 2019 November 11-12, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050920300806

[8] R. Oshikawa, J. Qian, and W. Y. Wang, "A survey on natural language processing for fake news detection," *arXiv preprint arXiv:1811.00770*, 2018.

[9] M. Umer, Z. Imtiaz, S. Ullah, A. Mehmood, G. S. Choi, and B.-W. On, "Fake news stance detection using deep learning architecture (cnn-lstm)," *IEEE Access*, vol. 8, pp. 156 695–156 706, 2020.

[10] M. G. Hussain, M. Rashidul Hasan, M. Rahman, J. Protim, and S. Al Hasan, "Detection of bangla fake news using mnb and svm classifier," in *2020 International Conference on Computing, Electronics Communications Engineering (iCCECE)*, 2020, pp. 81–85.

[11] J. Shaikh and R. Patil, "Fake news detection using machine learning," in *2020 IEEE International Symposium on Sustainable Energy, Signal Processing and Cyber Security (iSSSC)*, 2020, pp. 1–5.

Appendix

A Assistance for Running the Project

A.1 Fake News Detection Using Machine Learning

A.1.1 Clone Git Project

```
git@github.com:iamrosan/Fake-News-Detection-BCT-Mini-Project-.git
```

A.1.2 Installation Required

To run the project, python3.6 or above required and pip3 with latest version 20.X.X

To run the required libraries run the pip install as:

```
pip install -r requirements.txt
```

A.1.3 Running the Project

Go to the root folder of the FakeNewsDetection and execute the command in the command prompt or bash shell

```
py manage.py runserver
```

Then, go to the local host server

```
127.0.0.1:8000
```

B Snapshot

B.1 Article Tag News

B.2 Div Tag News

B.3 True news word cloud

B.4 Fake news word cloud

B.5 Outcome of desktop site

B.6 Unit test for Scraping

```
IDLE Shell 3.9.6
File  Edit  Shell  Debug  Options  Window  Help
Python 3.9.6 (tags/v3.9.6:db3ff76, Jun 28 2021, 15:26:21) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
========== RESTART: E:\My Works\python\beautiful soup 4\cnn\app.py ==========
URL: https://www.theguardian.com/football/2022/feb/10/chelsea-braced-for-kepa-arrizabalaga-bids-and-open-to-summer-exit

<Response [200]>

Title: Chelsea braced for Kepa Arrizabalaga bids and open to summer exit

Last modified on Fri 11 Feb 2022 05.27 GMT Chelsea are braced for bids for Kepa Arrizabalaga this summer and are unlikely to stand in the goalkeeper's way if they rece
ive a suitable offer. The Spain international has had a mixed time since leaving Athletic Bilbao for £71.6m in 2018 and his recent improvement has not been enough to d
islodge Édouard Mendy as Thomas Tuchel's first choice. Tuchel is considering whether to bring Mendy into the side when Chelsea face Palmeiras in the Club World Cup fin
al on Saturday and Arrizabalaga, who impressed in the semi-final victory over Al-Hilal on Wednesday, could look for a fresh challenge at the end of the season. Arrizab
alaga's form has improved since Tuchel replaced Frank Lampard last year and Chelsea believe he could be a summer target for one of the big Spanish clubs. The 27-year-o
ld has done well when given opportunities this season and Chelsea accept he will not want to continue as their No 2 if he has a chance to play regularly elsewhere. Che
lsea have not had to deal with any offers for Arrizabalaga. He struggled under Lampard and his asking price and high wages meant that interest was minimal. However, Ch
elsea know that having such an expensive goalkeeper on the bench regularly is not ideal and there would be more disappointment for Arrizabalaga if he is left out again
st Palmeiras. Arrizabalaga, who has made tweaks to his game under Tuchel, impressed while Mendy was representing Senegal at the Africa Cup of Nations. He helped Chelse
a reach the Carabao Cup final at Tottenham's expense last month, made two stunning saves in the second half against Al-Hilal and has become an expert at saving penalti
es. He has played a key role in Chelsea winning three shootouts this season and denied Ryan Hardie from the spot when Tuchel's side knocked Plymouth out of the FA Cup
last weekend. Although the feeling at Chelsea is that Arrizabalaga should be Spain's No 1 instead of Unai Simón or David de Gea, Tuchel could still drop him against Pa
lmeiras. Mendy has joined the Chelsea squad in Abu Dhabi after winning Afcon and has emerged as one of the best goalkeepers in the world since leaving Rennes in 2020.
Chelsea have never won the Club World Cup and Tuchel could decide he does not want to leave anything to chance. If Arrizabalaga is dropped against the Copa Libertadore
s champions, his next chance to impress could come in the Carabao Cup final against Liverpool on 27 February. Chelsea are yet to end the uncertainty hanging over their
defence, with César Azpilicueta, Antonio Rüdiger and Andreas Christensen out of contract this summer. However, they have resumed talks with Rüdiger over a new deal and
hope to reach an agreement with the centre-back, who is a target for Bayern Munich, Real Madrid and Paris Saint-Germain.
>>>
```