

# Identification of Fake News Using Machine Learning in Distributed System

by

Mehruz Saif

19101665

MD. Kamal Haque Kanon

19201139

Nazmul Hasan

19301277

MD. Shamim Hossen

15301101

Fatema Zohra Anannya

17101176

A thesis submitted to the Department of Computer Science and Engineering  
in partial fulfillment of the requirements for the degree of  
B.Sc. in Computer Science

Department of Computer Science and Engineering  
Brac University  
June 2021

© 2021. Brac University  
All rights reserved.

# Declaration

It is hereby declared that

1. The thesis submitted is my/our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

## Student's Full Name & Signature:

*M. Saif*

---

Mehruz Saif  
19101665

*K. Haque*

---

MD. Kamal Haque Kanon  
19201139

*Nazmul Hasan*

---

Nazmul Hasan  
19301277

*Shamim*

---

MD. Shamim Hossen  
15301101

*Fatema Zohra Anannya*

---

Fatema Zohra Anannya  
17101176

# Approval

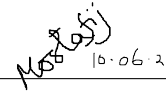
The thesis/project titled “Identification of Fake News Using Machine Learning in Distributed System” submitted by

1. Mehruz Saif (19101665)
2. MD. Kamal Haque Kanon (19201139)
3. Nazmul Hasan (19301277)
4. MD. Shamim Hossen (15301101)
5. Fatema Zohra Anannya (17101176)

Of Spring, 2021 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on June 06, 2021.

## Examining Committee:

Supervisor:  
(Member)



---

Mostafijur Rahman Akhond  
Lecturer

Department of Computer Science and Engineering  
Jashore University of Science Technology

Program Coordinator:  
(Member)

---

Md. Golam Rabiul Alam, PhD  
Associate Professor

Department of Computer Science and Engineering  
Brac University

Head of Department:  
(Chair)



---

Sadia Hamid Kazi, PhD  
Professor and Chairperson

Department of Computer Science and Engineering  
Brac University

## Ethics Statement

We, Mehruz Saif, MD. Kamal Haque Kanon, Nazmul Hasan, MD. Shamim Hossen, Fatema Zohra Anannya consciously assure that for the thesis paper “Identification of Fake News Using Machine Learning in Distributed System” the following is fulfilled:

1. This material is our own original work, which has not been previously published elsewhere.
2. The paper is not currently being considered for publication elsewhere.
3. The paper reflects our own research and analysis in a truthful and complete manner.
4. The paper properly credits the meaningful contributions of all our teammates.
5. The results are appropriately placed in the context of prior and existing research.
6. All of the sources utilized are correctly credited (correct citation). Text that has been copied verbatim must be marked as such with quote marks and a suitable reference.
7. We have all contributed directly and actively to the paper’s development, and we’ll accept public responsibility for its content. I agree with the above statements and declare that this submission follows the policies of BRAC University.

Date: 6th June 2021 Corresponding authors’ signatures:

*M. Saif*

---

Mehruz Saif  
19101665

*K. Haque*

---

MD. Kamal Haque Kanon  
19201139

*Nazmul Hasan*

---

Nazmul Hasan  
19301277

*Shamim*

---

MD. Shamim Hossen  
15301101

*Fatema Zohra Anannya*

---

Fatema Zohra Anannya  
17101176

## Abstract

The World Wide Web's launch and the rapid adoption of social media platforms (such as Facebook and Twitter) paved the way for unparalleled levels of information diffusion in human history. Consumers are creating and sharing more information on social media platforms than ever before, some of it is erroneous, deceptive, or has no influence on reality. Access to news information has become considerably simpler and more comfortable thanks to the Internet and social media. Online users may often follow events of interest, and the widespread usage of mobile devices makes this process easier. However, with great potential comes enormous responsibility. There are also a number of websites dedicated nearly entirely to the dissemination of fake news. Since it's a serious issue with a large-scale dataset, identification of fake news is very vital in this era, as social media and online newspapers are in large numbers in the web arena. That's why it is easy to spread rumors and create chaos. Also, the size of data sets is increasing day by day. Data is expanding at a quicker rate than processing rates. As a result, algorithms that need a huge quantity of data and processing are frequently conducted on a distributed computing system that separates multiple nodes on several machines which have concurrency of components and lack of a global clock. Also, nobody has used a distributed system to detect fake news before. In our paper, we tried to run 4 PySpark algorithms based on SPARK-Context which provides massive storage for big data processing and analysis and also has been found to be 100 times quicker in-memory, while disk performance was shown to be 10 times quicker on several devices at the same time. So that we can control and real-time monitoring over the news and data before it goes viral in the media.

**Keywords:** PySpark ML; RDD(Resilient Distributed Dataset); Random Forest; Factorization Machine Classifier; Linear SVC; Logistic Regression.

## **Dedication**

This thesis is dedicated to our parents and our honorable Supervisor Mostafijur Rahman Akhond, Lecturer. For their endless love, support, and encouragement. We the group members are really grateful to them for their valuable time and help to accomplish our work successfully.

## Acknowledgement

First and foremost, we want to express our gratitude to Allah for His unwavering support, which enabled us to continue our studies without severe setbacks.

Furthermore, we wanted to thank our Supervisor Mostafijur Rahman Akond. Our Sir was our guide throughout the entire endeavor. It would be hard to write a good research report without the help of our esteemed supervisor. He had backed us up by demonstrating a new way of data collecting. He also assisted us whenever we needed it and pointed us in the appropriate route toward the project's completion.

And it's also not possible to achieve our goals without the continued support of our parents. We are currently on the verge of graduating thanks to their generous support, endless love, and prayers.

# Table of Contents

<b>Declaration</b>	<b>i</b>
<b>Approval</b>	<b>ii</b>
<b>Ethics Statement</b>	<b>iii</b>
<b>Abstract</b>	<b>iv</b>
<b>Dedication</b>	<b>v</b>
<b>Acknowledgment</b>	<b>vi</b>
<b>Table of Contents</b>	<b>vii</b>
<b>List of Figures</b>	<b>ix</b>
<b>List of Tables</b>	<b>1</b>
<b>1 Introduction</b>	<b>2</b>
1.1 Introduction . . . . .	2
1.2 Problem Statement . . . . .	3
1.3 Aim of Study . . . . .	3
1.4 Research Methodology . . . . .	3
1.5 Thesis Outline . . . . .	4
<b>2 Related Work</b>	<b>5</b>
<b>3 Data Collection and Feature Selection</b>	<b>9</b>
3.1 Dataset Description . . . . .	9
3.1.1 Data Preprocessing . . . . .	10
3.1.2 Data Distribution . . . . .	11
3.1.3 Stop Word Removal . . . . .	15
3.2 Features Extraction . . . . .	17
3.2.1 TFI-DF (Term frequency–Inverse document frequency) . . . . .	17
3.2.2 Text Classification . . . . .	18
3.3 Feature Analysis . . . . .	18
3.3.1 Heat-map of Data . . . . .	19



<b>4</b>	<b>Proposed Model and Result Analysis</b>	<b>24</b>
4.1	Machine Learning . . . . .	24
4.2	Machine Learning with Pyspark . . . . .	24
4.3	Random Forest Implementation . . . . .	25
4.4	Factorization Machines Classifier Implementation . . . . .	27
4.5	Linear Support Vector Classifier Implementation . . . . .	29
4.6	Logistic Regression Implementation . . . . .	30
4.7	Spark-Context . . . . .	31
	4.7.1 RDD(Resilient Distributed Datasets) . . . . .	32
	4.7.2 MapReduce . . . . .	33
4.8	Results and Analysis . . . . .	33
<b>5</b>	<b>Conclusion and Future Work</b>	<b>40</b>
	<b>Bibliography</b>	<b>42</b>

# List of Figures

3.1	Dataset sample . . . . .	10
3.2	The blue bar is for true news and another bar is for fake news as we can see more than 40000 fake and real news are here. . . . .	11
3.3	Data distribution of fake and real data. . . . .	11
3.4	core-wise SpaCy Tokenization. . . . .	12
3.5	SpaCy Tokenization of sample text. . . . .	13
3.6	Explacy.py. . . . .	13
3.7	Explacy Tokenization of Sample Data. . . . .	14
3.8	Tokenization of sample data. . . . .	15
3.9	DisplaCy Render on sample data (different colors used for better visualization). . . . .	16
3.10	Plotting displaCy on sample data. . . . .	16
3.11	Plotting a date-time graph based on sample data. . . . .	17
3.12	Frequency Distribution plot on features using Text Classifier. . . . .	18
3.13	Heat map of training data based on Random Forest. . . . .	19
3.14	Heat map of testing data based on Random Forest. . . . .	19
3.15	Heat map of training data based on FM Classifier. . . . .	20
3.16	Heat map of testing data based on FM Classifier. . . . .	20
3.17	Heat map of training data based on Linear SVC. . . . .	21
3.18	Heat map of testing data based on Linear SVC. . . . .	21
3.19	Heat map of training data based on Logistic Regression. . . . .	22
3.20	Heat map of testing data based on Logistic Regression. . . . .	22
4.1	Random Forest Confusion Matrix. . . . .	27
4.2	Random Forest Roc Curve. . . . .	27
4.3	areaUnderROC 0.991430574968374. . . . .	28
4.4	FM Classifier Confusion Matrix. . . . .	29
4.5	Linear SVC Classifier Confusion Matrix. . . . .	30
4.6	only showing top 5 rows, f1: 0.9980664953490717, areaUnderROC 0.9998086271501443. . . . .	31
4.7	Logistic Regression Confusion Matrix. . . . .	32
4.8	Generating Map Partition of a job by RDD with Spark-submit. . . . .	35
4.9	Submitted job's time and Duration and completed tasks. . . . .	36
4.10	Summary Metrics of driver core. . . . .	36
4.11	Started a new stage after finishing stages by RDD. . . . .	37
4.12	Accuracy of all algorithm. . . . .	38
4.13	Visual representation of Accuracy indicating the difference between core 2 and core 4 after Distributing. . . . .	39

# List of Tables

3.1	Performance evaluation of algorithms on dataset. . . . .	23
4.1	Models and Distributed System core-wise accuracy. . . . .	38

# Chapter 1

## Introduction

### 1.1 Introduction

Fake news has become more prevalent in recent years as online social networks have grown in popularity [1]. The majority of the time, for different economic and political goals that are common in the Internet world. People who utilize online social networks are naturally infected by this bogus news, which has had a significant impact on offline culture [2]. Our goal in improving the trustworthiness of information on online social networks is to swiftly identify false news, evaluate what is true news, and share it. Detecting fake news items will be less important than identifying phony newsmakers and issues, which will help to remove a significant portion of fake news from its source in online social networks. The tasks of detecting a fake news item, its creator, and its subject are closely associated from an eminent standpoint since publications authored by a trustworthy individual should have better credibility, but someone who repeatedly uploads unauthentic material would have lower credibility. Correlations between news stories and news subjects have also been identified [3].

Bangladesh is a rumor-prone country in South Asia with a high percentage of social media penetration (22 percent in April 2019-January 2020). The country will have 36 million social media users by 2020, with the majority of them using Facebook (Kemp, 2020). Bangladesh's position differs from that of India, which is another rumor-prone neighbor. Rumor propagation is mostly focused on Facebook in Bangladesh (Al-Zaman, 2020a, 2020b), whereas it is primarily dependent on WhatsApp in India (Banaji, Bhat, Agarwal, Passanha and Sadhana Pravin, 2019) [4].

For example, a false news item regarding Kenya's election has been circulating on social media under the guise of being from CNN. Following that, on Friday, a bogus video replicating the BBC's Focus on Africa show was released. In both films, phony polls showed President Uhuru Kenyatta leading in polls before the August election. In reality, recent surveys show that neither he nor his opponent Raila Odinga have enough votes to win outright. According to a recent poll, 90 percent of Kenyans saw or heard fake news in the run-up to the election. CNN has confirmed the video report is a hoax on Twitter, while the BBC has asked viewers to check any reports claiming to be from the broadcaster on the BBC website [5].

The goal of this research is to achieve by using the main keywords of news and con-

necting and searching for those keywords all over the internet, web pages, and social media and articles, papers to see if that news is absolutely authentic or slightly changed or completely fake and unreliable. For this, we are going to use Apache SPARK, and Map Reducing System, which are unified analytic engines for big data processing, with built-in modules for streaming, machine learning, and graph processing. After finding, processing, and simulating data we are going to use a Distributed System to distribute that information as DS has high-performance stability.

## 1.2 Problem Statement

There are so many fake news detection algorithms already established but they are not reliable. They do not give complete accuracy of the real news. There are issues with collecting news from the web and most of the time they cannot find the actual author of the article. So, this can be a huge problem and general people will get wrong news and believe that. That can turn into something unthinkable which can cause problems in our society. We are proposing a better way to detect fake news through a distributed system which will not only give near perfect results but will also be able to deal with big data. Also, nobody has used a distributed system to detect fake news before. That is why we used a distributed system to fetch information from the internet and look for the actual author who wrote the article and see if the news is true or false. We strongly believe that our program works properly.

## 1.3 Aim of Study

Recovering the true story is our aim. In our thesis, we chose a base story to use as a unit story to find out if other articles related to the topics are valid or fake. We wish to stop people from getting misguided by altered news on the internet. Our focus is assisting the public in distinguishing authentic and forged news by real-time monitoring it. And also the data size of fake news is increasing day by day. For these large scales of data, we chose a distributed system for identifying fake news which not only helps us to keep a good amount of accuracy but also we can handle large sizes of data and try to minimize spreading false news among social media and the internet.

## 1.4 Research Methodology

Our plan is to develop a system that can detect fake news for distributed operating systems. Having this objective, we have gathered information and data from Kaggle, BuzzFeed which contain different types of news and accumulate data about false and real news. In addition to that, to choose the most significant highlights, we used feature selection algorithms like Term Frequency-Inverse Document Frequency (TF-IDF) and Text Classification. For this study, we had to learn about a distributed operating System like Spark, how they developed with programming models like RDD and map-reduce. After that, we chose four machine learning algorithms and we developed algorithms for our System with PySpark Machine Learning. Likewise, we had to choose methods used in Natural language processing and supervised learning

such as Random Forest, Factorization Machines Classifier, Linear SVC, and Logistic Regression. We took two datasets to train and test our system. Then we got different types of results based on our used algorithms. Since we work with big data, we also run those algorithms in different cores, and each time we set the core numbers with Spark-submit in Spark-context. After applying algorithms we got different results for each algorithm depending on core numbers and we analyzed every result to find the most suitable model for our prediction.

## 1.5 Thesis Outline

This report has an impact on developing a prediction model that would be useful in detecting fake and misleading data. The authors' goal is to deal with a large size dataset from the real world that can be utilized to train current supervised machine learning models to categorize fresh observations. The main focus of the article is on the actions taken by the researchers.

To begin, the introduction (Chapter 1) describes the reason for the study that led the writers to address this specific issue statement. The objectives of our study and a summary of our findings are briefly explained below.

In the Literature review section (Chapter 2), we covered works from a computer science background that tackled comparable issues. The goal of the background study was to identify the flaws in prior studies. Furthermore, we have explained our contribution and identified an appropriate way.

In the data collection phase (Chapter 3), we have explained from where we got data. This portion also included a description of the dataset. We also told how and which way to follow our train and test data and how reliable and consistent our dataset. Feature selection stated that by reducing the number of features, the time complexity may be minimized. The focus of the feature analysis was on the relevance and relevance of indicators in relation to the outcome 'Flag.'

Furthermore, model selection (Chapter 4) also covers our proposed models as well as a comparison of prediction rates among various models. In the context of our constructed data set, this section covers both conventional and complex algorithms. Furthermore, the results are shown to indicate which model works better for our data set.

# Chapter 2

## Related Work

Detecting fake news is a worthwhile aim that has the potential to positively influence our online and social media lives. We discovered numerous works in the field of fake news identification. One article details a technique for identifying bogus news. They attained an accuracy of roughly 74 percent for classification using the Naive Bayes approach. According to the author, fake news is both a worldwide issue and a global task. They employed machine learning techniques to address a variety of categorization challenges, including image, audio, and object detection. The author of this essay presents a general summary of the techniques accessible in this area. The author made use of two deception detection systems, one based on a vector machine and the other on a neural network. The first is a naive classifier. The author stated that they established a goal for themselves. Observe how this strategy performs when used to a specific problem. The author establishes a link between spam messages and bogus news. According to the author, spam message filtering and detecting fake news are pretty comparable. Their patterns are not identical [6]. He divulges some information that is matched between them, such as grammatical errors, the use of emotive color, and the request for Readers' perceptions, and they frequently rely on a comparable collection of data, etc. Additionally, the author demonstrated how a naive Bayes classifier is used to detect spam communications. The naive Bayes method was utilized. Possibility of obtaining the desired result.

The author also shows how they collect the data set from a different kind of platform like a news portal, social media. He also said they collect both fake news and true news and claim that it helps them to find a better result. The writer also shows the way they do the implementation process. He said that they use filters for all the news they have collected so far. Then they delete some useless data sets like broken data, text using Facebook API, the articles which are a mixture of true and false data, etc [7]. Then they shuffled data into three subsets like training dataset, validation dataset, and test dataset. They are facing some interesting parts when they use a classifier in the dataset. They found that the classifier ignores the word which is used less in the article and which is unknown to the classifier [8]. They are also making a point that if a word comes several times in the article it increases the probability of news being faked. After using their implementation process, they saw that the accuracy of the classifier for detecting both fake and true news is the same. The author shows that the classifier got 75 percent accuracy for overall detection from 927 datasets. The author gives some valuable suggestions so that we

can find better performance from this classifier methodology [9]. Firstly, he suggests collecting more dataset and using it for training as in machine learning methodology, the more data you get the better results you can collect. Secondly, the collection of datasets that is bigger in length is considered a better dataset. Thirdly, remove stop words from the news articles. Fourthly, he promotes the use of stem. Fifthly, he suggests treating words separately. Finally, he also suggests using a group of words instead of separate words for calculating probabilities. At last, the author concluded by saying that a simple AI algorithm like Naïve Bayes Classifier can show a good result in such important problems as fake news classification. The author also showed confidence that this research proves that AI techniques can also be successful for solving such important problems [10].

One paper represents the field of fake news detection, a little work has been done so far. There are some limitations to this kind of scenario. Despite this limitation, the researcher worked hard and developed some techniques to have a better result in this field. Among them, N-gram analysis gets better performance machine learning approaches are being used in this sector. Six alternative supervised classification approaches were used in conjunction with this methodology, including K-Nearest Neighbor (KNN), Support Vector Machine (SVM), Logistic Regression (LR), Linear Support Vector Machine (LSVM), Decision Tree (DT), and Stochastic Gradient Descent (SGD) [11]. These techniques have been used in both fake and real news and got significant results. Text analytics and predictive modeling techniques are also utilized to detect the news [12]. N-gram modeling, for starters, makes use of data pre-processing, which allows us to minimize the quantity of the real data. Secondly, stop word removal used to remove common words. Thirdly, Stemming reduces the number of similar words. Fourthly, features extensions performed by using Term Frequency (TF) and Term Frequency-Inverter document Frequency (TFI-DF). Finally, the classifier detects whether the news is true or fake. In several experiments Linear- based classifiers achieved better performance than nonlinear ones. Linear SVM got 92 percent accuracy in these experiments [13].

One paper represents, the creator asserts that dataset is a major issue. Measurable methodology for fighting phony news issues is restricted by the absence of marked benchmarked datasets. Publicly supporting methodology makes prepared information by nostalgic investigation. At that point, I consolidate the honest feeling. In any case, there is a jumble among preparing and testing, when we utilize genuine informational collections, the outcome can be problematic. Since positive information was made in a reenacted stage. The creator utilized an informational collection named LAIR, which size is huge, it contains 12.8 k physically named short explanations [13]. These announcements are from more regular settings like TV advertisements, Facebook posts, tweets, talk with, news discharge, and so on. The creator utilized well-known learning-based techniques like calculated relapse, vector machines, long transient memory organizations, and a convolutional neural organization model.

Surfaced-name semantic examples were used by the author. To demonstrate solid execution in the short content arrangement, the creator used five baselines: a regularized calculated relapse classifier (LR), uphold vector machine classifier (SVM),



bi-directional long momentary memory networks model (Bi-LSTMs), convolutional neural organization model (CNNs), and LIBSHORTTEXT toolbox for LR and SVM. TensorFlow is used to implement Bi-LSTMs and CNNs [14]. The developer employed 300-dimensional word2vec embeddings from Google News to warm up the content embeddings. Hyperparameters are fine-tuned when the dataset is approved. The hyperparameters for LR and SVM models were tuned using lattice search. The assessment metric is set to precision [15]. The inventor provides a network of implanting vectors and a convolutional layer to encode the metadata embeddings and catch the dependency between the meta-information vectors at random (s). In the dormant space, a typical max-pooling activity is done, followed by a bi-directional LSTM layer.

Then, feed the fully associated layer with a SoftMax actuation capacity the maximum pooled text portrayals and the meta-information portrayal from the bi-directional LSTM to generate the final expectation. TOn this dataset, the bigger portion gauge yields 0.204 and 0.208 precision on the approval. SVMs and LR models saw significant improvements, however, Bi-LSTMs did not. On the holdout test set, the CNN's outperformed all models, achieving an exactness of 0.270. If we compare the predictions from the CNN model to those from SVMs using a two-followed combination test, we find that CNN is significantly superior (p.0001). The model produced the best result on the test data after taking into account all meta-information and text. This corpus can likewise be utilized for position characterization, contention mining, theme demonstrating, gossip location, and political NLP research.

According to a survey of the current literature, the author of the study presented a three-step procedure for false news classification (consisting of feature extraction, relabeling, and learning) [15]. Two of these are repeated at different levels repeatedly. To begin, they use  $N = \{n1, n2, \dots, nn\}$  as a collection of news items,  $L0$  as a collection of initial labels, and  $M = \{m1, m2, \dots, mn\}$  as a collection of trained models. In their method, the data  $N$  is manually pre-processed before being utilized to train a classifier. This is the first time in N that the records have been cleaned manually. Furthermore, the training data, known as  $N$  which is achieved. Moreover, the training data which is referred to as  $N$ , is translated to numeric values. Performance of these features selection to omit some unimportant features that may add complexity while degrading the algorithm's performance and/or accuracy. Second, in the third step, the Author solves the multiclass label problem by relabeling records. Similar characteristics with multiple class labels are first treated as though they were single class labels. For example, in the case of a set of original labels  $L0 = \{1, 2, 3, 4, 5\}$ , the first three labels i.e. 1, 2, and 3 can be regarded as a single label.

Aside from that, 4 and 5 have greater ratings, indicating that they have high-label ratings. The Author then reduced the multiclass labels to two class labels, and the multiclass problem was addressed using a binary class solution from there. They trained the classifier using the new class labels assigned in the previous phase after the rebellious process. The result of the ML model is  $mi \in M$  capture these relationships. They were able to define the refinement process as the rebelling process and re-learning iteratively. The refining procedure is carried out for each

binary class separately, as seen in the preceding example, where the low-class label is relabeled as low (1, 2) and high as high (1, 2). (3). The data is then relabeled and utilized to train a new model  $m_j \in M$ .

In another paper, as discussed earlier on this topic, for testing the algorithm the Author used SVM and decision tree are two separate classification techniques. For experimenting, they divided the dataset 70:30 split between training and test datasets (10235 records, 7165 were used for training and 3071 records were used for testing purposes.). After testing,  $m_1$  delivers an accuracy of 85.5 percent for all true and false class labels. 2180 of the 2260 records fall into the true category, whereas 448 of the 811 entries go into the false category. Then  $m_2$  produces 80.59 percent accuracy and  $m_3$ ,  $m_4$ ,  $m_5$  produces respectively, 89.7 percent, 80.7 percent, and 85.8 percent accuracy. When  $k = 10$ , cross-validation using SVM with a dataset size of 10235 and  $k$ -fold classification. After training, the accuracy of  $m_1$ ,  $m_2$ ,  $m_3$ ,  $m_4$  and  $m_5$  are respectively 80 percent, 69.08 percent, 82.02 percent, 70.39 percent, and 72.38 percent [16].

The authors of another paper studied the fake news article, creator, and subject detection problem. Based on the connections among news articles, creators, and news subjects, a deep diffusive network model has been proposed to incorporate the network structure information into model learning. They also introduce a new diffusive unit model, namely GDU. Model GDU accepts multiple inputs from different sources simultaneously, and can effectively fuse these inputs for output generation with content "forget" and "adjust" gates [17]. Another study looked into the challenge of detecting false news articles, creators, and subjects. A deep diffusive network model has been developed to include network structure information into model learning based on the links between news pieces, creators, and news subjects. They also offer the GDU, a new diffusive unit concept. Model GDU receives many inputs from many sources at the same time and uses content "forget" and "adjust" gates to efficiently fuse these inputs for output creation [18].

In this article, the Author needed to clarify the accessible methods of the phony news location. As indicated by the Author, these days the recognition of phony news is a hot territory of exploration and increased considerably more examination enthusiasm among the scientists. Individuals could identify counterfeit news at two levels, to be specific the reasonable level and operational level. To characterize that thoughtfully there are three kinds of phony news: viz I. Genuine Fabrications ii. Scams and iii. Parody. In this article Author utilized two profound learning-based models to address the issue of phony data recognition in the multi-space stage. Right off the bat, for the model they dealt with, Embedding Layer. Along these lines, of each word acquired from both the passes. In the third layer, Word Level Attention: they applied the consideration model at word level that the goal is to let the model choose which words are significantly contrasted with different words while foreseeing the objective class (counterfeit/genuine).

# Chapter 3

## Data Collection and Feature Selection

In this study, the datasets we used are freely available online and open source. The set of data contains both false and real news articles from multiple domains and channels. The real news articles published include truthful descriptions of real-world events. On the other hand, fake news blogs and websites make statements that are not based on evidence. Many of those stories' assertions from the domain of politics can be manually reviewed using fact-checking services such as polifact.com and snopes.com. In this research, we have used two different datasets, which will be described as follows.

### 3.1 Dataset Description

BuzzFeed News gathered the data, which was used to learn and evaluate the Random Forest, FM classifier, linear SVC, and logistic models. This dataset provides information on Facebook postings, which is equivalent to a news item. They were all gathered from Facebook pages and three major political news websites (ABC News, Politico, CNN). The dataset's shape is (20800, 5), which means it contains 20800 titles along with 20800 texts and authors.

The first dataset we used is from ISDDC 2017 and available at Kaggle, which contains a total of 21417 articles used for training and testing. The dataset was compiled from a variety of online databases. The papers are not restricted to a particular area, such as politics, and contain both false and real articles from a variety of other fields.

The second dataset, which comprises a total of 44,898 articles, both false and actual, is also accessible on Kaggle. The true news pieces originate from renowned news organizations such as CNN, Reuters, and the New York Times, while the bogus news items originate from dodgy news websites. Sports, entertainment, and politics were among the topics discussed. All have four features and one label.

	title	text	subject	date
0	As U.S. budget fight looms, Republicans flip t...	WASHINGTON (Reuters) - The head of a conservat...	politicsNews	December 31, 2017
1	U.S. military to accept transgender recruits o...	WASHINGTON (Reuters) - Transgender people will...	politicsNews	December 29, 2017
2	Senior U.S. Republican senator: 'Let Mr. Muell...	WASHINGTON (Reuters) - The special counsel inv...	politicsNews	December 31, 2017
3	FBI Russia probe helped by Australian diplomat...	WASHINGTON (Reuters) - Trump campaign adviser ...	politicsNews	December 30, 2017
4	Trump wants Postal Service to charge 'much mor...	SEATTLE/WASHINGTON (Reuters) - President Donal...	politicsNews	December 29, 2017
5	White House, Congress prepare for talks on spe...	WEST PALM BEACH, Fla./WASHINGTON (Reuters) - T...	politicsNews	December 29, 2017
6	Trump says Russia probe will be fair, but time...	WEST PALM BEACH, Fla (Reuters) - President Don...	politicsNews	December 29, 2017
7	Factbox: Trump on Twitter (Dec 29) - Approval ...	The following statements were posted to the ve...	politicsNews	December 29, 2017
8	Trump on Twitter (Dec 28) - Global Warming	The following statements were posted to the ve...	politicsNews	December 29, 2017
9	Alabama official to certify Senator-elect Jone...	WASHINGTON (Reuters) - Alabama Secretary of St...	politicsNews	December 28, 2017

Figure 3.1: Dataset sample

As we can see the figure 3.1, the dataset has five columns and among them four are features and one is the label. Title, text, subject, and date are the target variables for detecting fake and true news. More on this will be discussed in the data pre processing part.

### 3.1.1 Data Preprocessing

For fake news detection, here a target column has been built. The task of this column is when 1 will be in the target column then it is fake news, while it is true news when the target column is 0. After reading the data, the two datasets are concatenated and sorted and the index column is ignored. The sorting has been done based on data.

### 3.1.2 Data Distribution



Figure 3.2: The blue bar is for true news and another bar is for fake news as we can see more than 40000 fake and real news are here.

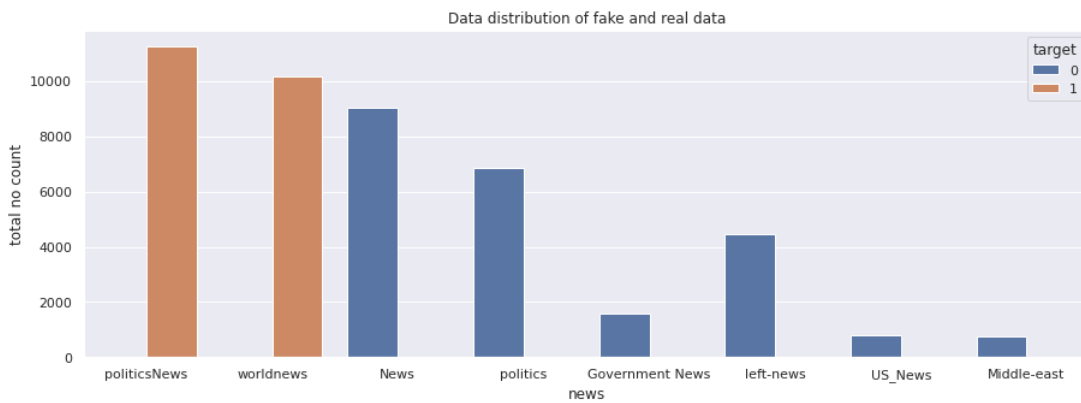


Figure 3.3: Data distribution of fake and real data.

In 3.3 the data distribution part, we can see news and information coming from topics like political news, world news, government news, left news, US and Middle-East news. In the dataset, the text and title parts are separate but when we come to the visualization part we will enter the title into the text part.

Since our data has a large number of texts, we are using it to tackle an interesting natural language processing (NLP) like sentiment or text classification. We used to explore textual data using the spaCy library and it's available on the GitHub repository which will help to explain the NLP datasets and build a text classification model. From spaCy, the text and the target parts are concatenated which has been created in another form and put on data.

We used extract linguistic features like

- Tokenization.
- Part-of-speech tagging.
- Dependency parsing.
- Lemmatization.
- Named entities recognition.
- Sentence Boundary Detection for building language models later.

```
spacy_tok = spacy.load('en_core_web_sm')
sample_data=data.text[100]
sample_data
```

Figure 3.4: core-wise SpaCy Tokenization.

And on this data, spacy.load has a core web. The task of the core-web is that the whole data is loaded core-wise and processed, which is almost the type of NLTK but different. After that, the Explacy repository has been used which is another Github repository. It works in a different way from spaCy. It will show a tree and give a visualization of how it is working.

## Visualizing Data

- explacy - explaining how parsing is done
- displacy - visualizing named entities

Senator Warren hits out at 'effort to politicize' U.S. consumer agency  
WASHINGTON (Reuters) - Democratic Senator Elizabeth Warren is taking aim at budget chief Mick Mulvaney's plan to fill the ranks of the U.S. consumer financial watchdog with political allies, according to letters seen by Reuters, the latest salvo in a broader battle over who should run the bureau. President Donald Trump last month appointed Mulvaney as acting director of the Consumer Financial Protection Bureau (CFPB), though the decision is being legally challenged by the agency's deputy director, Leandra English, who says she is the rightful interim head. Mulvaney told reporters earlier this month he planned to bring in several political appointees to help overhaul the agency, but Warren warned in a pair of letters sent Monday to Mulvaney and the Office of Personnel Management (OPM), which oversees federal hiring, that doing so was inappropriate and potentially illegal. The CFPB is meant to be an independent agency staffed primarily by non-political employees. Hiring political appointees could violate civil service laws designed to protect such employees from undue political pressure and discrimination, Warren said. "Your naked effort to politicize the consumer agency runs counter to the agency's mission to be an independent voice for consumers with the power to stand up to Wall Street banks," Warren, who helped create the CFPB, wrote to Mulvaney. In a separate letter, Warren asked the OPM to review Mulvaney's "unprecedented and unjustified" plans. In a third letter sent to Mulvaney and English, Warren asked for information about a review of ongoing enforcement actions at the CFPB. Reuters reported earlier this month that a potential multimillion-dollar settlement with Wells Fargo is among the enforcement actions under review amid the change in CFPB leadership. Spokespeople for Mulvaney and the OPM did not immediately respond to requests for comment. Mulvaney, who also serves directly under Trump as the head of his Office of Management and Budget (OMB), said in the long term he would like to see professional staff alongside political appointees, mirroring an arrangement in place at the OMB. "We will be staffing up with more permanent political people so the professional staff here have a better feel for where the administration wants to take the bureau," Mulvaney said. But Warren said such an arrangement, though understandable for bureaus like the OMB which sit directly beneath the White House, was not suitable for independent financial regulators. The leadership of the CFPB has been in question since the agency's first director, Richard Cordray, resigned in November.

Figure 3.5: SpaCy Tokenization of sample text.

```
!wget https://raw.githubusercontent.com/tylernelon/explacy/master/explacy.py

--2021-05-25 00:32:55-- https://raw.githubusercontent.com/tylernelon/explacy/master/explacy.py
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.111.133, 185.199.108.133, 185.199.109.133, ...
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.111.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 6896 (6.7K) [text/plain]
Saving to: 'explacy.py'

explacy.py          100%[=====] 6.73K  --.-KB/s   in 0s

2021-05-25 00:32:55 (68.8 MB/s) - 'explacy.py' saved [6896/6896]
```

Figure 3.6: Explacy.py.

Dep tree	Token	Dep type	Lemma	Part of Sp
	U.S. military to accept transgender recruits on Monday : PentagonWASHINGTON ( Reuters ) - Transgender people will be allowed for the first time to enlist in the U.S. military starting on Monday as ordered by federal courts , the Pentagon said on Friday , after President Donald Trump 's administration decided not to appeal rulings that blocked	compound ROOT aux relcl compound dobj prep pobj punct punct appos punct punct punct compound nsubjpass aux auxpass ccomp prep det amod pobj aux relcl prep det compound pobj advcl prep pobj mark advcl agent amod pobj punct det nsubj ROOT prep pobj punct prep compound compound pobj compound nsubj ROOT neg aux xcomp dobj nsubj relcl	U.S. military to accept transgender recruit on Monday : pentagonwashington ( Reuters ) - transgend people will be allow for the first time to enlist in the U.S. military start on Monday as order by federal court , the Pentagon say on Friday , after President Donald Trump 's administration decide not to appeal ruling that block	PROPN NOUN PART VERB NOUN NOUN ADP PROPN PUNCT NUM PUNCT PROPN PUNCT PUNCT ADJ NOUN VERB AUX VERB ADP DET ADJ NOUN PART VERB ADP DET PROPN NOUN VERB ADP PROPN PUNCT ADP PROPN PROPN PART NOUN VERB PART PART VERB NOUN DET VERB

Figure 3.7: Explacy Tokenization of Sample Data.



When the sapCy tokenization of the sample has been parsed the data to the explacy then it shows how to make a tree by connecting word by word. From figure 3.7, we can see that first “US”, then “military”. And it is linked with “US”. Again “Pentagon” is connected with “Reuters”. So, in this place, the whole news is created like a tree from word by word and we can assume and determine what the tree looks like.

And by the help of the explacy it will also find the word description type, lemmatization, and whether the word is noun, pronoun, adjective or verb, and so on. In lemmatization, if the “ing”, “ed” is linked with the word, then it will remove those for better prediction and we can count the comparison. And the tree is regenerating when the text will go from one full-stop to another full-stop.

	text	lemma	pos	dep	is_punctuation	shape	is_alpha	is_stop
0	Senator	Senator	PROPN	compound	False	Xxxxx	True	False
1	Warren	Warren	PROPN	nsubj	False	Xxxxx	True	False
2	hits	hit	VERB	ROOT	False	xxxx	True	False
3	out	out	ADP	prt	False	xxx	True	True
4	at	at	ADP	prep	False	xx	True	True

Figure 3.8: Tokenization of sample data.

To accurately model and for better prediction, we are using tokenization in the data frame. In figure 3.8, as we can see, has nine columns. The first column is the index column and the next columns are text, lemmatization, preposition, description, punctuation, shape, alpha, and stop. For example, in index 2, the lemmatization of “hits” text is “hit”, it is “verb”, it is root type, it has no punctuation, it’s shaped like it has 4 letters, whether it is an alphabet or a stop word.

### 3.1.3 Stop Word Removal

Stop words are meaningless words in a language that, when used as text classification features, produce noise. There are terms that are often used in sentences to help link ideas or aid in sentence construction. Stop words include articles, prepositions, conjunctions, and certain pronouns. Popular terms like a, about, an, are, as, at, be, by, with, from, how, in, is, of, on, or that, the, these, this, too, was, when, where, where, how, will, and so on were deleted. The processed papers were preserved and carried on to the next level after those terms were deleted from each paper. For an easy and quickest way to predict fake news, we used the displaCy.

Visualizing a dependency parse or called entities in a text isn’t just an enjoyable NLP demonstration; it can also help us speed up the creation and debugging of our code and training. In the displaCy, when we parsed the text part it was showing how it was rendering the words of the text.

Senator **Warren PERSON** hits out at 'effort to politicize' **U.S. GPE** consumer agency WASHINGTON ( **Reuters ORG** ) - **Democratic NORP** Senator **Elizabeth Warren PERSON** is taking aim at budget chief **Mick Mulvaney PERSON** 's plan to fill the ranks of the **U.S. GPE** consumer financial watchdog with political allies, according to letters seen by **Reuters ORG** , the latest salvo in a broader battle over who should run the bureau. President **Donald Trump PERSON** **last month DATE** appointed **Mulvaney LOC** as acting director of **the Consumer Financial Protection Bureau ORG** ( **CFPB ORG** ), though the decision is being legally challenged by the agency's deputy director, **Leandra English PERSON** , who says she is the rightful interim head. **Mulvaney PERSON** told reporters **earlier this month DATE** he planned to bring in several political appointees to help overhaul the agency, but **Warren PERSON** warned in a pair of letters sent **Monday DATE** to **Mulvaney GPE** and **the Office of Personnel Management ORG** ( **OPM ORG** ), which oversees federal hiring, that doing so was inappropriate and potentially illegal. The **CFPB ORG** is meant to be an independent agency staffed primarily by non-political employees. Hiring political appointees could violate civil service laws designed to protect such employees from undue political pressure and discrimination, **Warren PERSON** said. "Your naked effort to politicize the consumer agency runs counter to the agency's mission to be an independent voice for consumers with the power to stand up to Wall Street banks," **Warren PERSON** , who helped create the **CFPB ORG** , wrote to

Figure 3.9: DisplaCy Render on sample data (different colors used for better visualization).

In figure 3.9, "Waren" is a person that's why it shows Person, besides it, is violet. The US, Mulvaney are "GPE" and Reuters, CFPB are organizations that's why it shows "ORG" beside them. Monday, Last month, Earlier this month show them as the date with the help of the displaCy. And this is not for all words, it is only for keywords.

Since our focus is the identification of fake and real news, there will be a large number of texts inside any news. That's why for a better view and easy to understand we used the displaCy which is another built-in library of the spaCy.

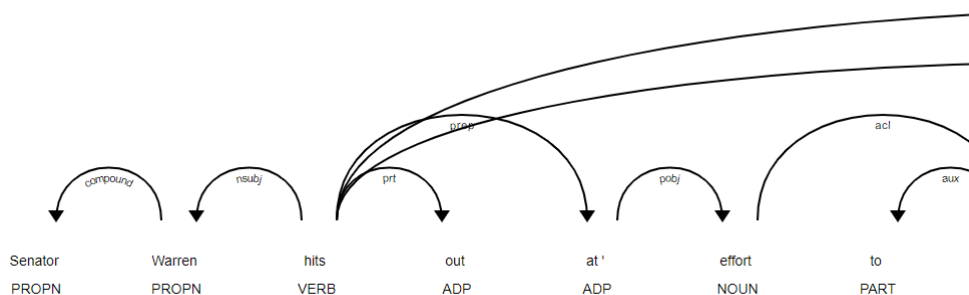


Figure 3.10: Plotting displaCy on sample data.

Basically, we can see here 3.10, hits come from Warren and Senators come from Warren. And it is also showing that they are verbs, propositions accordingly.

For detecting fake news, it is also necessary to check and monitor the time and date. So, we import a new library date time and it will distribute and separate the date and time of the news of the dataset. We use year, month as a target variable. After that, by labeling "Number of fake news" as Y-axis and "Month-Year" as X-axis we plot a graph.

By plotting a graph it is showing that in the dataset, most of the news are from 2017-09 to 2018-02.

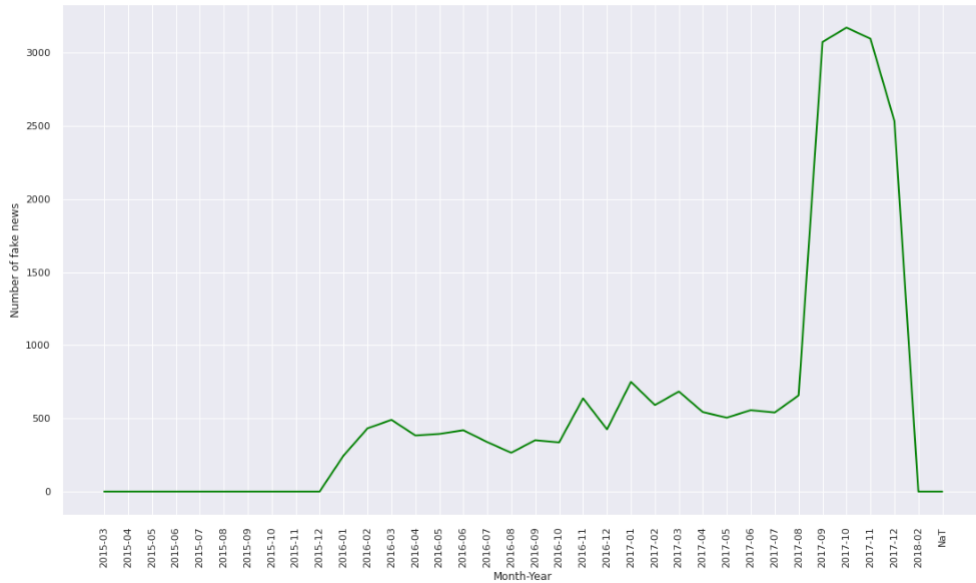


Figure 3.11: Plotting a date-time graph based on sample data.

## 3.2 Features Extraction

One of the most difficult aspects of text classification is learning from high-dimensional input. Documents have a vast number of expressions, sentences, and phrases, putting a heavy cognitive load on the learning process. Furthermore, obsolete and unnecessary features will degrade the classifiers' accuracy and efficiency. As a result, feature reduction can be used to decrease the size of text features and prevent using a broad space of features dimension. In this study, we looked at two separate approaches for selecting features: Term Frequency (TF) and Term Frequency-Inverse Document Frequency (TF-IDF). The following sections detail these techniques.

### 3.2.1 TFI-DF (Term frequency–Inverse document frequency)

In particular natural language processing and retrieval, the Term Frequency-Inverse Document Frequency (TF-IDF) is a weighting metric. It's a mathematical metric for determining the significance of a word to a text in a dataset. The number of times a phrase appears in the text increases its worth, although this is offset in the corpus by the word's frequency. Term frequency,  $tf(t,d)$ , is the frequency of term  $t$ ,

$$tf(t, d) = \frac{f_{t,d}}{\sum_{t' \in d} f_{t',d}} \quad (3.2.1)$$

Where  $f_{t,d}$  is the raw count of a term in a document, i.e. how many times the word  $t$  appears in document  $d$ .

The document frequency-inverse is a measurement of how much data and information a word gives across all documents, i.e. whether it's frequent or uncommon. It's the scaled inverse of logarithmically fraction of the documents that contain the word (The logarithm of the quotient is determined by dividing the total number of documents by the number of documents that include the term):

$$idf(t, d) = \log \frac{N}{|\{d \in D : t \in d\}|} \quad (3.2.2)$$

Then tf-idf is calculated as:

$$tfidf(t, d, D) = tf(t, f) \cdot idf(t, d) \quad (3.2.3)$$

One of the key features of IDF is that it scales up the uncommon terms while scaling down the common ones. Words like "the" and "then" appear often in the code, and if we just use TF, expressions like these would control the frequency count. Using IDF, on the other hand, reduces the importance of these expressions. We will take a slice of fake news, to see what vocabulary there looks like.

### 3.2.2 Text Classification

The next move is to preprocess the text after the dataset has been imported. Numbers, special characters, and unintended spaces are all possible in text. We may or may not need to delete these special characters and numbers from the code, depending on the issue. We would, however, delete all special characters, numbers, and needless spaces from our text for the sake of clarity. Here in the below figure, based on the Frequency Distribution, most of the texts are Trump, then said, people respectively.

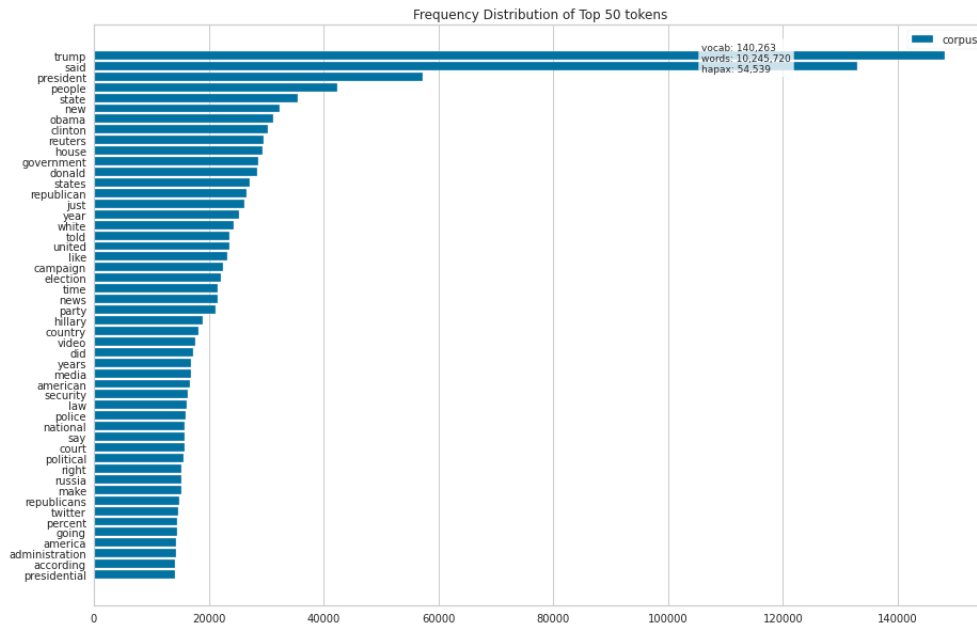


Figure 3.12: Frequency Distribution plot on features using Text Classifier.

### 3.3 Feature Analysis

In our research, the purpose is to identify fake and truthful news by real-time monitoring. As we discussed, we used two types of features and also 8 types of pySaprk

ML features SQLTransformer, RegexTokenizer, StopWordsRemover, CountVectorizer, Imputer, IDF, StringIndexer, VectorAssembler to train our model. After that, for testing our model we import a multi-classification evaluator and binary classification evaluator from pySpark ML classification to test our implemented models' results depending on precision, recall, and f1-score. We represent them by heatmap below.

### 3.3.1 Heat-map of Data

Secondly, based on our data we have generated the heat maps. For a big dataset with more than 60 features and around 500 samples, a heat map offers a superior visual representation. Because color tone takes up less room than digits for representing data. In our set of testing data of Random Forest in the below figure, the heatmap shows that accuracy, macro average, weighted average based on precision, recall, and f1-score.

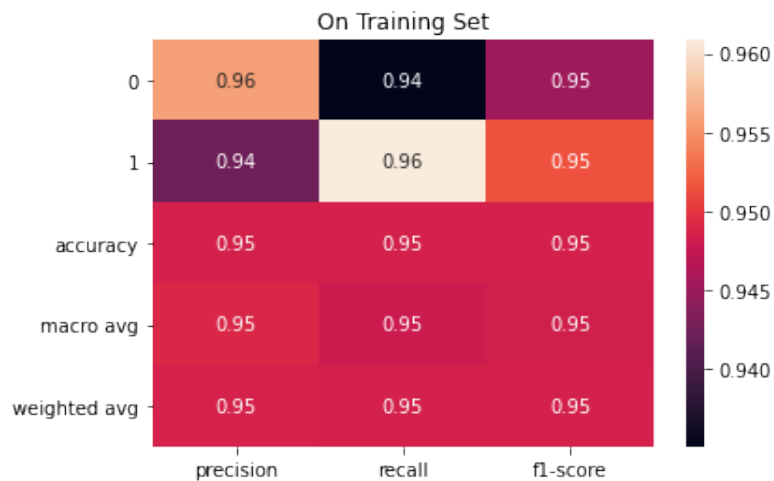


Figure 3.13: Heat map of training data based on Random Forest.



Figure 3.14: Heat map of testing data based on Random Forest.

For FM Classifier, we got different numbers of the accuracy of our training dataset macro and weighted average than Random Forest depending on precision-recall and f1-score.



Figure 3.15: Heat map of training data based on FM Classifier.

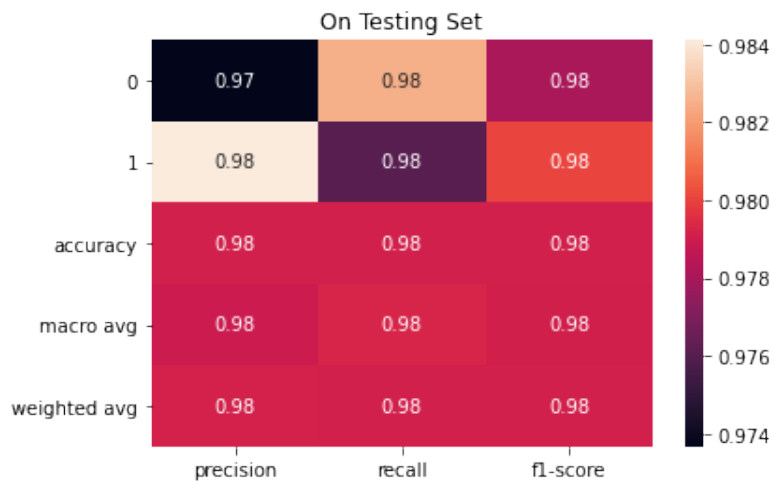


Figure 3.16: Heat map of testing data based on FM Classifier.

Same as those two models, we got the accuracy of our training dataset macro and weighted average depending on precision-recall and f1-score for Linear SVC model.



Figure 3.17: Heat map of training data based on Linear SVC.

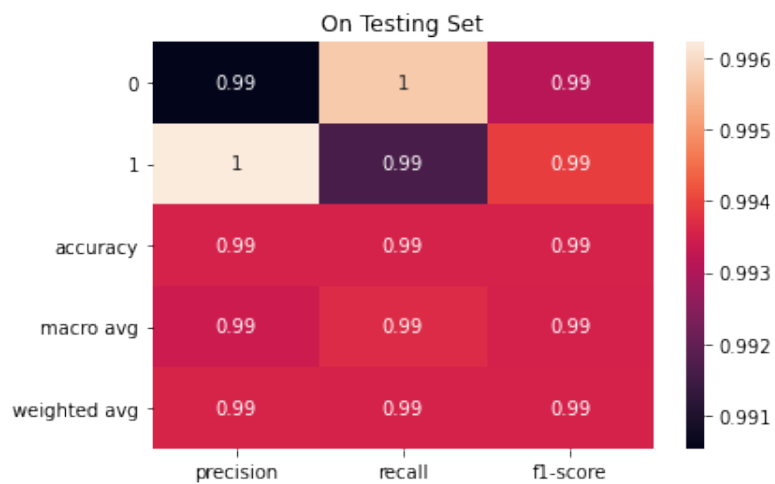


Figure 3.18: Heat map of testing data based on Linear SVC.

And the last model we used on Logistic Regression, the accuracy of our training dataset macro and weighted average depending on precision-recall and f1-score for Logistic Regression model and all of their numbers are 1.

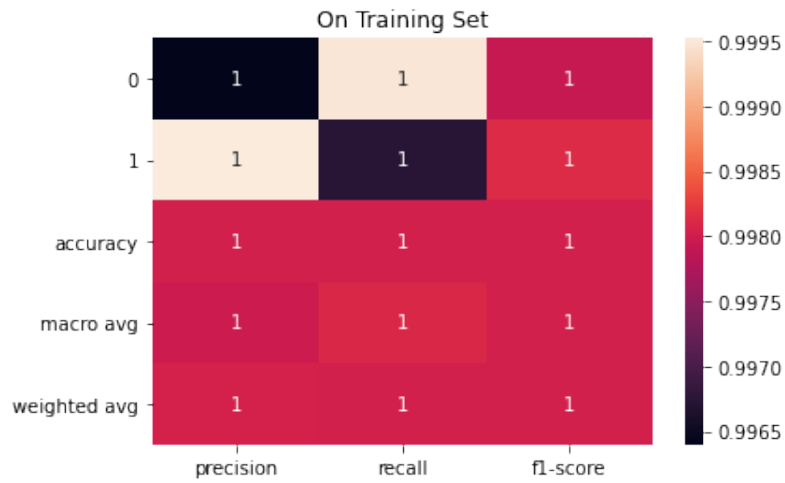


Figure 3.19: Heat map of training data based on Logistic Regression.



Figure 3.20: Heat map of testing data based on Logistic Regression.



Model name	Feature Name	Testing Data			Training Data		
		Precision	Recall	F1-score	Precision	Recall	F1-score
Random Forest	accuracy	0.95	0.95	0.95	0.99	0.99	0.99
	Macro-avg	0.95	0.95	0.95	0.99	0.99	0.99
	Weighted-avg	0.95	0.95	0.95	0.99	0.99	0.99
FM Classifier	accuracy	1	1	1	0.98	0.98	0.98
	Macro-avg	1	1	1	0.98	0.98	0.98
	Weighted-avg	1	1	1	0.98	0.98	0.98
Linear SVC	accuracy	1	1	1	0.99	0.99	0.99
	Macro-avg	1	1	1	0.99	0.99	0.99
	Weighted-avg	1	1	1	0.99	0.99	0.99
Logistic Regression	accuracy	1	1	1	1	1	1
	Macro-avg	1	1	1	1	1	1
	Weighted-avg	1	1	1	1	1	1

Table 3.1: Performance evaluation of algorithms on dataset.

# Chapter 4

## Proposed Model and Result Analysis

### 4.1 Machine Learning

The sort of data analysis that uses artificial intelligence to create analytical models is called machine learning. It's an artificial intelligence area predicated on the idea that data, spot patterns, and making decisions with little effort are learned by computers without human intervention. Thanks to advancements in computer technology, machine learning today is not the same as machine learning in the past. Pattern recognition and the notion are motivated by it that computers may learn to do tasks without being explicitly taught how to do so; artificial intelligence researchers wanted to see if computers could learn from the data [19]. The iterative feature of machine learning is crucial because models may evolve autonomously as they are exposed to fresh data. They use past computations to provide consistent, repeatable judgments and outcomes. It's a science that's not new, but it's gaining new traction. While many machine learning techniques have been known for a while, the capacity to apply difficult mathematical computations to large amounts of data automatically – again and again, quicker and quicker – is a relatively new phenomenon. In our study, we have incorporated a machine learning approach to identify fake news.

### 4.2 Machine Learning with Pyspark

Python may be used with Spark thanks to PySpark, an Apache Spark Community utility. It enables python users to manipulate RDDs (Resilient Distributed Datasets). PySpark Shell, which links Python APIs to Spark core to start Spark-Context, is also included. For machine learning applications, Apache Spark provides a very powerful API. Its purpose is to make machine learning accessible to everyone. It has primitives for lower-level optimization and APIs for higher-level pipelines. The most common applications are predictive analytics solutions, recommendation engines, and fraud detection systems. PySpark enables machine learning applications to run on distributed clusters, billions and trillions of data may be processed 100 times quicker and faster than traditional and standard Python programs. Apache Spark is a big data processing engine with built-in modules for streaming, SQL,

Machine Learning (ML), and graph processing that is known for being fast, easy to use, and general. Data engineers will benefit from learning Spark, knowing Spark will help data scientists with investigational data analysis (EDA), feature extraction, and, of course, machine learning. When we have to code on Spark, it builds an API on local. Depending on how to get that machine, we have to choose IP or website. In our study, we have incorporated a PySpark machine learning approach to identify fake news. We created RDD by `ss.createDataFrame`. In this case, we used different PySpark ML Features like `SQLTransformer`, `RegexTokenizer`, `StopWordsRemover`, `CountVectorizer`, `Imputer`, `IDF`.

### 4.3 Random Forest Implementation

First of all, we have used Random forest for the classification which is an ensemble learning is a classification, regression, and another task-solving approach that works by building a large number of decision trees at training time and then outputting the class that is the mode of the classes (classification) or the mean/average prediction (regression) of the individual trees. For splitting our dataset in the random forest classifier, we used 80 percent data as training data and the rest 20 percent data for testing. We set the feature column and label column for our dataset in this algorithm and we set the hyperparameters into 7 as `maxDepth` and maximum iteration as `numTrees`. The random forest training algorithm uses the common approach of bootstrap aggregation, or bagging, to train tree learners. Bagging repeatedly ( $B$  times) chooses a random sample with the training set replaced, and fits trees to these samples: Bagging regularly ( $B$  times) chooses a random sample from a training set  $X = x_1, \dots, x_n$  and responses  $Y = y_1, \dots, y_n$  and fits trees to these samples: For  $b = 1, \dots, B$

- $n$  training instances from  $X, Y$  are sampled with replacement; they are referred to as  $X_b, Y_b$ .
- Train a classification or regression tree  $f_b$  on  $X_b, Y_b$ .

Averaging predictions from all the separate regression trees on  $x'$  may be used to make predictions for unknown data after training.

$$\hat{f} = \frac{1}{B} \sum_{b=1}^B f_b(x') \tag{4.3.1}$$

Spark estimates the relevance of a feature for each decision tree by accumulating the gain, scaled by the number of samples flowing through the node:

$$f_i = \sum_{j: \text{nodes } j \text{ splits on feature } i} s_j C_j \tag{4.3.2}$$

- $f_i$  = the value of feature  $i$
- $s_j$  = samples' number reaching node  $j$
- $C_j$  = the value of node  $j$ 's impurity

The feature importance for each tree is first normalized in proportion to the tree to determine the final feature importance at the Random Forest level:

$$normfi_i = \frac{fi_i}{\sum_{j \in \text{all features}} fi_j} \quad (4.3.3)$$

- $normfi_i$  = the feature's normalized importance  $i$
- $fi_i$  = the value of the feature  $i$

After that, the feature importance values from each tree are added together and normalized:

$$RFfi_i = \frac{\sum_j normfi_{ij}}{\sum_{j \in \text{all features}, k \in \text{all trees}} normfi_{jk}} \quad (4.3.4)$$

- $RFfi_i$  =  $i$  computed the relevance of a feature from all trees in the Random Forest model.
- in tree  $j$ , the normalized feature importance for  $i$

In our scenario, max features are used to partition total features, resulting in an approximately increased number of trees for bootstrap aggregation. The Main Area Under Curve in our case is 0.9895, or around 98.95 percent, while the f1 score is 0.94. Our train and test set were scaled using Standard Scaler. It helped with data normalization within a range. We've created a graph to aid us in fine-tuning the estimator parameters for this method. The performance measurements were derived from the following confusion matrix

$$\begin{bmatrix} 4065 & 322 \\ 191 & 4485 \end{bmatrix}$$

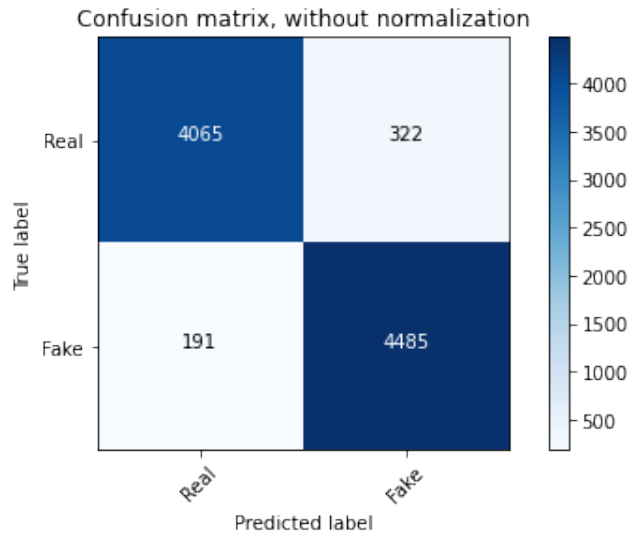


Figure 4.1: Random Forest Confusion Matrix.

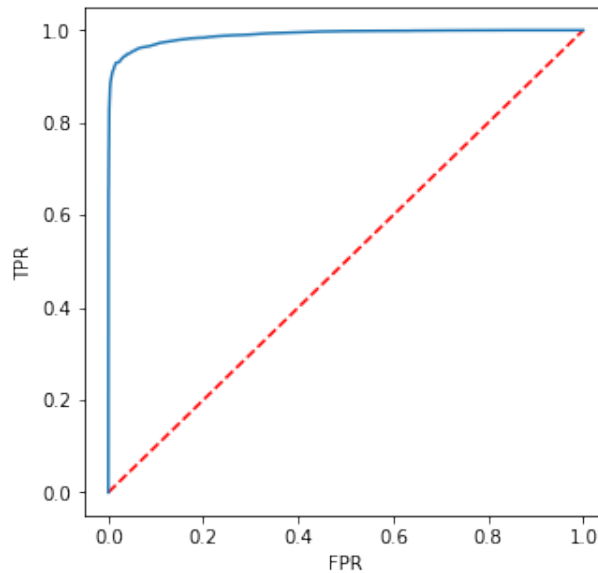


Figure 4.2: Random Forest Roc Curve.

From figure 4.1 confusion matrix the right prediction number for real and fake news is 4065 and 4485 and the wrong prediction number 322 and 191 respectively.

## 4.4 Factorization Machines Classifier Implementation

Factorization Machines are a type of supervised learning algorithm that may be used to solve both classification and regression issues. It's a linear model that's been extended to capture interactions between features in high-dimensional sparse datasets in a cost-effective manner. In a basic matrix factorization model that maps both users and objects to a shared latent component space of dimensions  $D$ . user-item interactions are considered as inner products. As a result, each item  $I$  is linked

to a vector  $q_i$ , whereas each user  $u$  is linked to a vector  $p_u$  [20]. In our study, we used to make research on matrix factorization-based machine learning (ML) models easier to predict fake news. Factorization is a term used to describe the process of Machines that may represent a wide range of latent component models and are commonly employed for collaborative filtering tasks (Rendle, 2012b). One of the most significant benefits of the Factorization Machine is that the model equation is simple.

$$\hat{y}(\mathbf{x}) := w_0 + \sum_{i=1}^n w_i x_i + \sum_{i=1}^n \sum_{j=i+1}^n \langle \mathbf{v}_i, \mathbf{v}_j \rangle x_i x_j \quad (4.4.5)$$

where the parameters of the model that need to be estimated are:

$$w_0 \in R, \quad \mathbf{w} \in R^n, \quad \mathbf{V} \in R^{n \times k} \quad (4.4.6)$$

And  $\langle \cdot, \cdot \rangle$  is the dot product of two  $k$ -dimensional vectors:

$$\langle \mathbf{v}_i, \mathbf{v}_j \rangle := \sum_{f=1}^k v_{i,f} \cdot v_{j,f} \quad (4.4.7)$$

For splitting our dataset in the Factorization Machines classifier, we used 80 percent data as training data and the rest 20 percent data for testing. Here we also set the feature column and label column for our dataset in this algorithm and we set the hyperparameters into 7 as maxDepth and maximum iteration as numTrees. We examined the model's performance using test data after it was built. As a result of our developed confusion matrix, we estimated an accuracy score of 0.9738 of 97.38 percent. Furthermore, the F1 Score, which was derived using a weighted average of precision and recall, was 97.38 percent.

```

+---+-----+
|FPR|                                TPR|
+---+-----+
|0.0|                                0.0|
|0.0|0.002020739165115661|
|0.0|0.004094655676681733|
|0.0|0.005902685455995746|
|0.0|0.007710715235309758|
+---+-----+
only showing top 5 rows

```

Figure 4.3: areaUnderROC 0.991430574968374.

FM Classifier doesn't produce any curve graph but it shows the ROC number. The ROC presented upper was generated using the FPR and TPR values acquired from the confusion matrix.

$$\begin{bmatrix} 4121 & 96 \\ 137 & 4551 \end{bmatrix}$$

In 4.4 confusion matrix the right prediction number for real and fake news is 4121 and 4551 and the wrong prediction number 96 and 137 respectively.

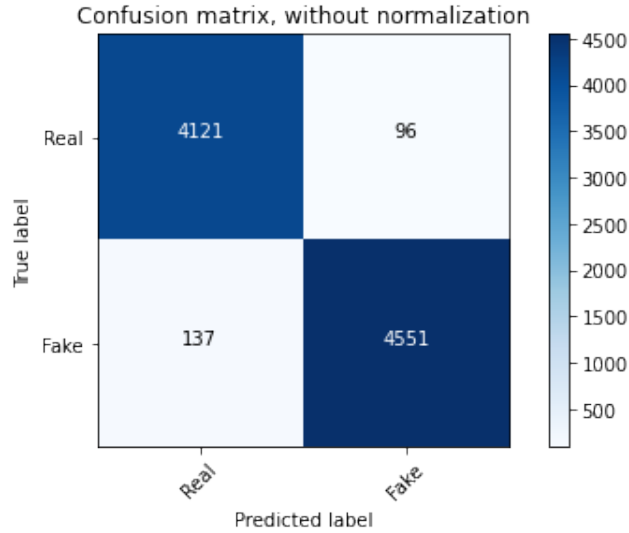


Figure 4.4: FM Classifier Confusion Matrix.

## 4.5 Linear Support Vector Classifier Implementation

A Linear SVC (Support Vector Classifier) is intended to match the data we offer and provide a "best fit" hyperplane that divides or categorizes our data [20]. Following that, we may input some characteristics to our classifier to check what the "predicted" class is once we've obtained the hyperplane.  $C$  is a regularization parameter that defines the trade-off between a low training error and a low testing error, which relates to the capacity of your classifier to generalize to new data. Consider the aim function of a linear SVM:

$$\min |w|^2 + C \sum \xi \quad (4.5.8)$$

The linear separator is often constructed with the greatest distance between the hyper-plane and the nearest negative and positive samples. This results in training data that is comparable to, but not identical to, testing data.  $C$  constrains the slack variables. We may accomplish the same results as the Hard Margin SVM by setting  $C$  to positive infinity. However, if we set  $C$  to 0, there will be no constraint, and we will end up with a hyper-plane that does not classify anything. Smaller  $C$  values result in a wider margin, but at the cost of some misclassifications; larger  $C$  values result in the Hard Margin classifier, which tolerates zero constraint violation.

$$\min_{w,b,\zeta} \frac{1}{2} \|w\|^2 + C \sum_{i=1}^m \zeta_i \quad (4.5.9)$$

$$\text{subject to } y_i (w \cdot x_i + b) \geq 1 - \zeta_i, \zeta_i \geq 0, i = 1 \dots m \quad (4.5.10)$$

A narrower margin will be acceptable for greater values of  $C$  if the decision function is better at accurately categorizing all training points. Gamma determines the amount of curvature in the context of a decision boundary. A greater gamma is indicated by more curvature. Gamma is a hypermeter that is placed before the training

model to give the decision boundary curvature weight, and C is a hypermeter that is positioned before the training model to control error. In our study for Linear SVC (Support Vector Classifier), we also used 80 percent of data for training data and the rest 20 percent for test data. In our study, we also plot precision, recall, and f1-score for Linear Support Vector classifiers based on training data and test data. The Linear SVC classifier doesn't produce any curve graph but it shows the ROC number (areaUnderROC: 0.9999998073628851). The ROC presented upper was generated using the Test and train values acquired from the confusion matrix.

[4186, 18]  
[40, 4743]

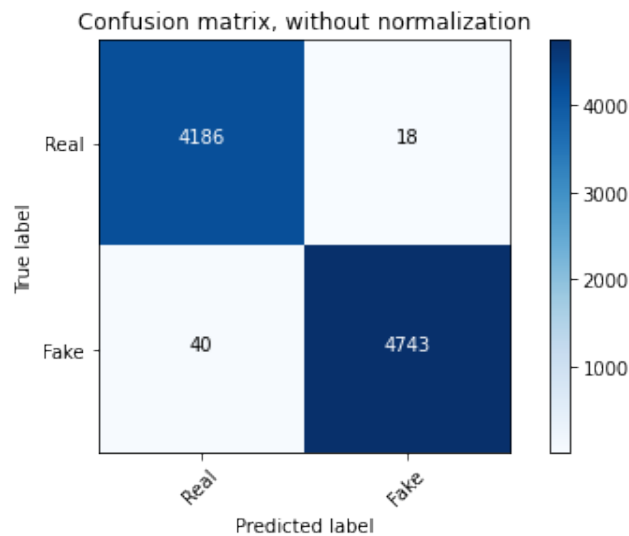


Figure 4.5: Linear SVC Classifier Confusion Matrix.

Figure 4.5 represents confusion matrix the right prediction number for real and fake news is 4186 and 4743 and the wrong prediction number 18 and 40 respectively. So, in this case, as we can see the number of wrong information of fake and real news is actually minimized.

## 4.6 Logistic Regression Implementation

In its most basic form, logistic regression is a statistical model that uses a logistic function to describe a binary dependent variable, however there are many more complicated forms. In regression analysis, logistic regression (or logit regression) is a methodology for estimating the parameters of a logistic model (a form of binary regression). The link function is  $\log(p/1-p)$ . By using a logarithmic transformation on the result variable we can represent a non-linear link in a linear fashion. In Logistic Regression, this is the equation that is employed. The odd ratio here is  $(p/1-p)$ . In order to anticipate an output value, input data (x) are linearly blended using weights or coefficient values (abbreviated as Beta in Greek) (y). The result is a binary value (0 or 1) rather than a numeric number, which is a significant difference from linear regression. We choose the following logistic regression equation:



$$y = e^{(b_0 + b_1 * x)} / (1 + e^{(b_0 + b_1 * x)}) \tag{4.6.11}$$

Where y is the expected output, b0 represents the bias or intercept term, and b1 represents the single input value coefficient (x). Our training data must be used to learn the b coefficient (a constant real value) for each column in our input data.

Here in our study, we used SQLTransformer, RegexTokenizer, StopWordsRemover, CountVectorizer, Imputer, IDF as libraries of pySpark ML features. We also import StringIndexer, VectorAssembler, StopWords- Remover, loadDefaultStopWords from pySpark ML features. From the dataset of title, we extracted tokens, removed stop words, and computed Term frequency-inverse document frequency from the title. After that, we removed the stop word from the text again. Then we apply VectorAssembler. Here, we set features as a feature column and fake as label a column. For better accuracy, we change max iteration as 50 regression parameter as 0.3 and elastic net parameter as 0.1. 80 percent of data is used for training data and the rest 20 percent is used for testing data. We estimated an accuracy score of 99.99 percent. Furthermore, the F1 Score, which was derived using a weighted average of precision and recall, was 0.9980.

```

+---+-----+
|FPR|          TPR|
+---+-----+
|0.0|          0.0|
|0.0|0.001794195250659...|
|0.0|0.003588390501319261|
|0.0|0.005382585751978892|
|0.0|0.007176781002638522|
+---+-----+

```

Figure 4.6: only showing top 5 rows, f1: 0.9980664953490717, areaUnderROC 0.9998086271501443.

The ROC presented upper was generated using the FPR and TPR values acquired from the confusion matrix

$$\begin{bmatrix} 4258 & 3 \\ 14 & 4517 \end{bmatrix}$$

Here in the upper confusion matrix the right prediction number for real and fake news is 4258 and 4517 and the wrong prediction number 3 and 14 respectively and using logistic regression, the number of wrong information of fake and real news is more and more minimized than others.

## 4.7 Spark-Context

The size of data sets is increasing. Data is expanding at a quicker rate than processing rates. As a result, algorithms that need a huge quantity of data and processing are frequently conducted on a distributed computing system. Nodes (networked

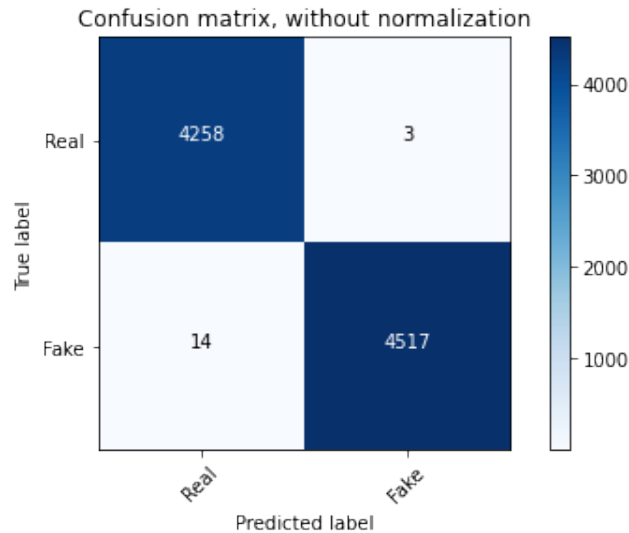


Figure 4.7: Logistic Regression Confusion Matrix.

computers) in a distributed computing system conduct processes in parallel and communicate. Since our study is related to big data, that’s why we implement spark for a distributed system. A cluster computing system is Spark (Apache’s open-source Big-Data processing engine). When compared to other cluster computing systems, it is quicker (such as Hadoop). It has Python, Scala, and Java high-level APIs. In Spark, writing parallel tasks is simple. Our models’ used codes are also in pySpark.

### 4.7.1 RDD(Resilient Distributed Datasets)

Since its creation, RDD has been Spark’s principal user-facing API. An RDD is a distributed immutable collection of data components partitioned among nodes in a cluster that can be processed in parallel with a low-level API that supports transformations and actions [21]. RDD avoids all HDFS reading and writing. RDD provides a quicker approach to obtain and process data in a Hadoop cluster by drastically lowering I/O operations. In fact, Hadoop MapReduce programs are thought to spend more than 90 percent of their time reading and writing to HDFS.

- There might be millions of nodes and edges in a large dataset.
- The SparkContext is set up in the first few lines. We make RDD lines out of it.
- The lines RDD are then transformed into RDD edges. The edges RDD stores the function on each line as well as key-value pairs of the pattern (1, 2), (1, 3), (2, 3), (3, 4),...
- ReduceByKey then collects all the key – pairs that correspond to a certain key and num. The Neighbours function is used to generate the degree of each vertex in a distinct RDD Adj list of the type (1, 2), (2, 1), (3, 1), and so on.
- The above code can be run by the following commands -

```
D: \ > cd D: \ Spark \ spark-3.0.2-bin-hadoop 2.7\ bin
```

```
D: | Spark (spark-3.0.2-bin-hadoop2.7 \ bin > spark-submit -driver-cores 4  
Fake-news-random-forest.py
```

## 4.7.2 MapReduce

MapReduce is the programming model that is utilized for distributed computing. Map and Reduce are the two steps of the MapReduce paradigm.

- Map- Each line of input data is processed by the mapper (it is in the form of a file), key-value pairs.

Input data  $\rightarrow$  Mapper  $\rightarrow$  list([key, value])

- After the Mapper's function, the reducer processes the list of key-value pairs. It creates a new collection of key-value pairs as a result.

list([key, value])  $\rightarrow$  Reducer  $\rightarrow$  list([key, list(values)])

## 4.8 Results and Analysis

Following the development of the model, its performance was assessed to see how well it could predict bogus and true news. The performance metrics we utilized were based on four factors from the confusion matrix. The parameters were TP, FP, TN, and FN, with True positive and True negative reflecting the number of correctly predicted observations. The ratio of accurately predicted samples to total samples was used to measure accuracy. The following equation was used to calculate the classifier's correct prediction rate:

$$Acc = \frac{TP + TN}{TP + TN + FP + FN} \quad (4.8.12)$$

Precision also showed the proportion of accurately predicted positive observations to the total number of positive observations in the test set.

$$Precision = \frac{TP}{TP + FP} \quad (4.8.13)$$

The sensitivity of the model determined how well it could predict sample outcomes when compared to all of the actual outcomes in the test set. In our experiment, when the model classified an observation as 'addicted' representing the person was vulnerable to addiction; sensitivity recognized the pattern of anticipating the correct flag. The equation to compute sensitivity or recall is

$$Sens = \frac{TP}{TP + FN} \quad (4.8.14)$$

The model's specificity was measured by how often it could predict negative values out of all the actual negative values. When the model produced the result 'sober,'

suggesting that the person was not prone to addiction, it was really predicting how often the algorithm could correctly anticipate negative events.

$$Spec = \frac{TN}{TN + FN} \quad (4.8.15)$$

Finally, the weighted average of accuracy and sensitivity is the F1-score. It was thought to be a superior measure of performance since it functioned even when the model's class distribution was unequal. The F1-score is calculated using the following formula:

$$F1 - score = \frac{2( Sensitivity * Precision )}{Sensitivity + Precision} \quad (4.8.16)$$

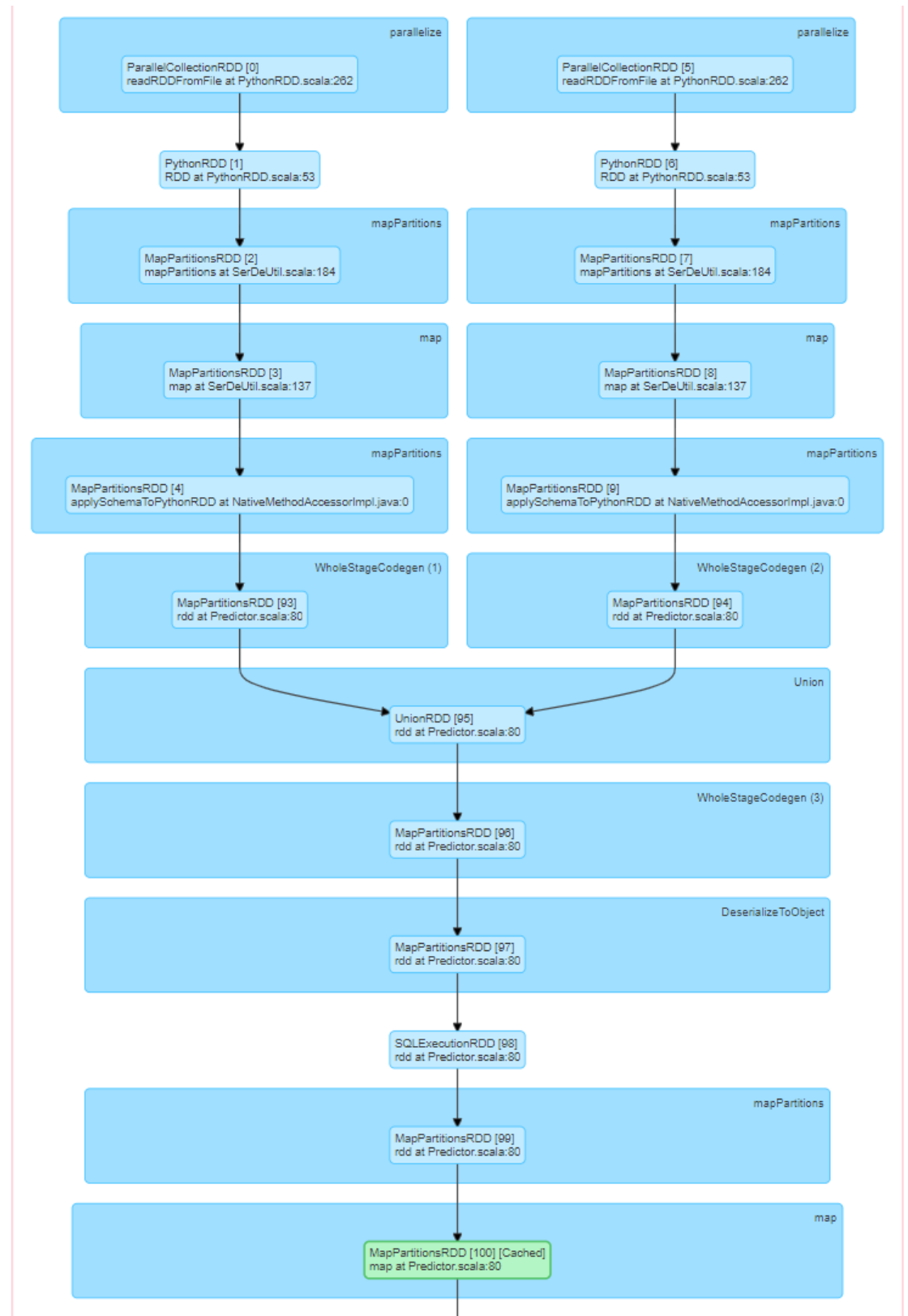


Figure 4.8: Generating Map Partition of a job by RDD with Spark-submit.

Figure 4.8, when our PySpark codes run with Spark-submit, it creates RDD blocks by partitioning maps. In a cluster, the nodes of the dataset are separated into different cores. In map(), DataFrame/Dataset is transformed to each row and returns the new altered Dataset by Spark map() by applying a function.

In mapPartitions(), This is identical to map(), with the exception that Spark mapPartitions() allows you to do expensive initializations (such as Database connection) once for each partition rather than on each DataFrame row. When dealing with heavy-weighted initialization on larger datasets, this improves job performance.

Job Id	Description	Submitted	Duration	Stages: Succeeded/Total	Tasks (for all stages): Succeeded/Total
82	crosstab at NativeMethodAccessorImpl.java:0 crosstab at NativeMethodAccessorImpl.java:0	2021/05/29 07:21:23 (kill)	7 s	0/2	57/102 (2 running)

Figure 4.9: Submitted job's time and Duration and completed tasks.

Figure 4.9 represents the computing time, duration of submitted jobs, and running and completed tasks. In 4.10 no figure, the blue bar represents scheduler delay, the red bar represents task deserialization time, green represents executor computing time, and so on. As a result, we can see the summary of each core metrics by the executors.

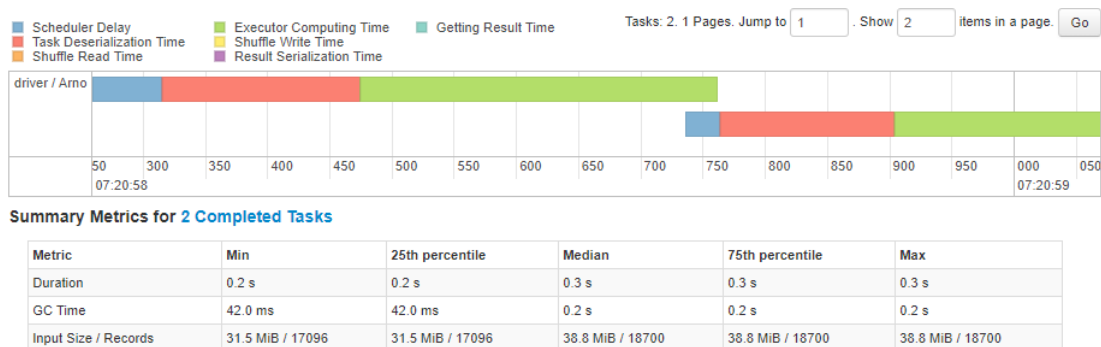


Figure 4.10: Summary Metrics of driver core.

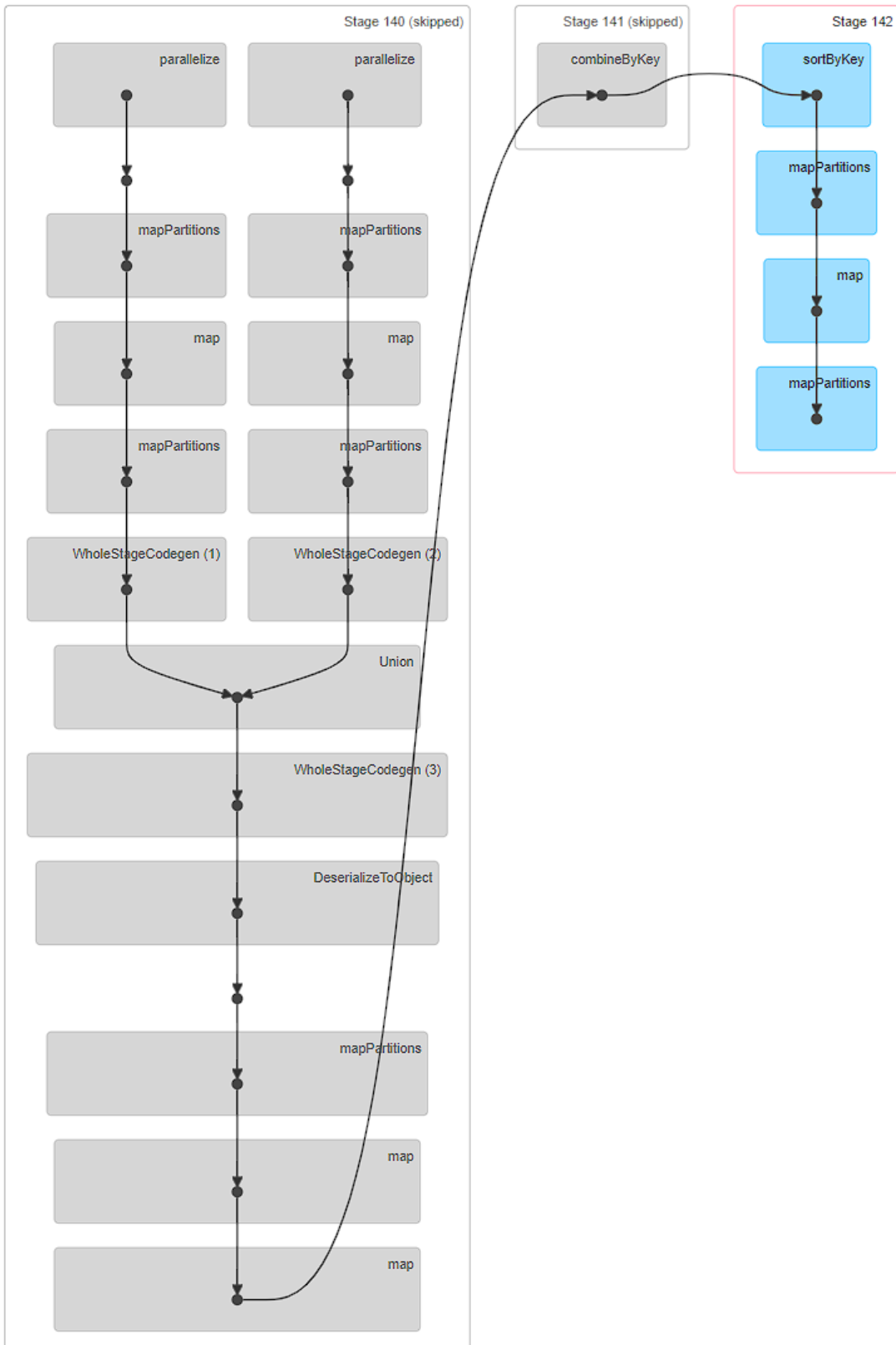


Figure 4.11: Started a new stage after finishing stages by RDD.

Model Name	In Distributed System		PySpark Accuracy
	core	accuracy	
Random Forest	2	95.76%	98.95%
	4	97.26%	
FM Classifier	2	95.91%	99.91%
	4	92.18%	
Linear SVC	2	99.30%	99.77%
	4	93.25%	
Logistic Regression	2	99.75%	99.88%
	4	96.82%	

Table 4.1: Models and Distributed System core-wise accuracy.

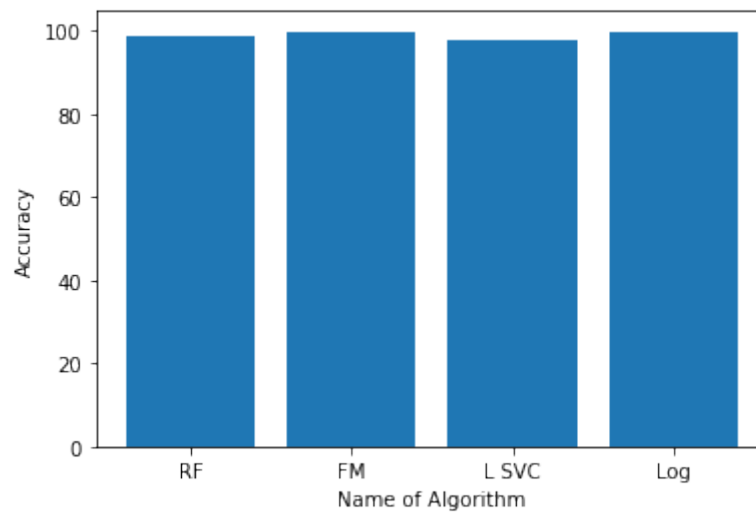


Figure 4.12: Accuracy of all algorithm.



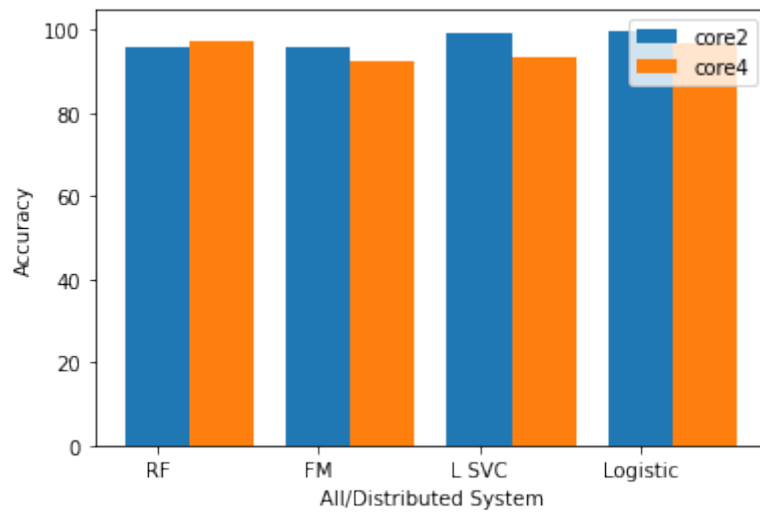


Figure 4.13: Visual representation of Accuracy indicating the difference between core 2 and core 4 after Distributing.

# Chapter 5

## Conclusion and Future Work

The main purpose of our work is to find out the truth behind the story within a short time. In this work, we tried to identify fake and truthful news using the distributed system where we used Spark-Context and different kinds of parameters and attempting to figure out which features work perfectly with our dataset and how it changes. We also built a model with PySpark machine learning that attempted to keep it as objective as possible when training the model in such a way that it will update itself after each cycle of training and testing. We also utilized TF-IDF and text classifiers in our dataset to get better accuracy. By applying 4 PySpark ML for the distributed system and they are Random Forest, Factorization Machine Classifier, Linear SVC, and Logistic Regression. On the other hand, since it is a serious issue and also we tried to deal with big data, for that reason we bring the distributed system which can give the near-perfect result. Our work is different from the previous works in many ways and nobody has used a distributed system to detect fake news before. In DS our PySpark ML codes run in a cluster and divide the nodes into multiple cores where we are able to set the core numbers. It also shows the completed and uncompleted jobs and tasks, executors in SparkUI or localhost from where we can see the computing each task, time, duration, and step by step the help of RDD map partition.

And our future work is to build a real version of a distributed system with the help of Apache Ambari and for this, we need multiple machines and labs. Since we do not have multiple machines in our hands at this moment for this pandemic and lockdown situation, we're going to make real use of it once the lockdown takes off and the situation turns into healthy pandemic-free.

# References

- [1] P. Ciprian, “The growing importance of social media in business marketing,” *Quaestus*, no. 7, p. 94, 2015.
- [2] C. Wardle and H. Derakhshan, “Information disorder: Toward an interdisciplinary framework for research and policy making,” *Council of Europe report*, vol. 27, pp. 1–107, 2017.
- [3] P.-H. Lambert, D. M. Ambrosino, S. R. Andersen, R. S. Baric, S. B. Black, R. T. Chen, C. L. Dekker, A. M. Didierlaurent, B. S. Graham, S. D. Martin, *et al.*, “Consensus summary report for cepi/bc march 12–13, 2020 meeting: Assessment of risk of disease enhancement with covid-19 vaccines,” *Vaccine*, vol. 38, no. 31, pp. 4783–4791, 2020.
- [4] M. S. Al-Zaman, S. A. Sife, M. Sultana, M. Akbar, K. T. S. Ahona, and N. Sarkar, “Social media rumors in bangladesh,” *Journal of Information Science Theory and Practices*, vol. 8, no. 3, pp. 77–90, 2020.
- [5] K. Somerville, “British media coverage of the post-election violence in kenya, 2007–08,” *Journal of Eastern African Studies*, vol. 3, no. 3, pp. 526–542, 2009.
- [6] M. Granik and V. Mesyura, “Fake news detection using naive bayes classifier,” in *2017 IEEE First Ukraine Conference on Electrical and Computer Engineering (UKRCON)*, IEEE, 2017, pp. 900–903.
- [7] H. Ahmed, I. Traore, and S. Saad, “Detecting opinion spams and fake news using text classification,” *Security and Privacy*, vol. 1, no. 1, e9, 2018.
- [8] K. P. Murphy *et al.*, “Naive bayes classifiers,” *University of British Columbia*, vol. 18, no. 60, 2006.
- [9] X. Li, S. Yang, R. Fan, X. Yu, and D. Chen, “Discrimination of soft tissues using laser-induced breakdown spectroscopy in combination with k nearest neighbors (knn) and support vector machine (svm) classifiers,” *Optics & Laser Technology*, vol. 102, pp. 233–239, 2018.
- [10] J. Singh and J. Singh, “A survey on machine learning-based malware detection in executable files,” *Journal of Systems Architecture*, p. 101 861, 2020.
- [11] E. Yaman and A. Subasi, “Comparison of bagging and boosting ensemble machine learning methods for automated emg signal classification,” *BioMed research international*, vol. 2019, 2019.
- [12] W. Y. Wang, ““liar, liar pants on fire”: A new benchmark dataset for fake news detection,” *arXiv preprint arXiv:1705.00648*, 2017.
- [13] H. Jwa, D. Oh, K. Park, J. M. Kang, and H. Lim, “Exbake: Automatic fake news detection model based on bidirectional encoder representations from transformers (bert),” *Applied Sciences*, vol. 9, no. 19, p. 4062, 2019.

- [14] M. G. Hussain, M. R. 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 (iC-CECE)*, IEEE, 2020, pp. 81–85.
- [15] S. Das and A. K. Kolya, “Predicting the pandemic: Sentiment evaluation and predictive analysis from large-scale tweets on covid-19 by deep convolutional neural network,” *Evolutionary Intelligence*, pp. 1–22, 2021.
- [16] T. Rasool, W. H. Butt, A. Shaukat, and M. U. Akram, “Multi-label fake news detection using multi-layered supervised learning,” in *Proceedings of the 2019 11th International Conference on Computer and Automation Engineering*, 2019, pp. 73–77.
- [17] J. Zhang, B. Dong, and S. Y. Philip, “Fakedetector: Effective fake news detection with deep diffusive neural network,” in *2020 IEEE 36th International Conference on Data Engineering (ICDE)*, IEEE, 2020, pp. 1826–1829.
- [18] X. Zhang and A. A. Ghorbani, “An overview of online fake news: Characterization, detection, and discussion,” *Information Processing & Management*, vol. 57, no. 2, p. 102 025, 2020.
- [19] S. Hakak, M. Alazab, S. Khan, T. R. Gadekallu, P. K. R. Maddikunta, and W. Z. Khan, “An ensemble machine learning approach through effective feature extraction to classify fake news,” *Future Generation Computer Systems*, vol. 117, pp. 47–58, 2021.
- [20] S. Rendle, “Factorization machines with libfm,” *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 3, no. 3, pp. 1–22, 2012.
- [21] Y. Ohno, S. Morishima, and H. Matsutani, “Accelerating spark rdd operations with local and remote gpu devices,” in *2016 IEEE 22nd International Conference on Parallel and Distributed Systems (ICPADS)*, IEEE, 2016, pp. 791–799.