Paper Title: *Performance Analysis and Comparison of Machine Learning Algorithms for Fake News Detection.*

Mir Mumit-Ul-Islam
Department-of-CSE
EastWestUniversity
Dhaka, Bangladesh
2016-2-60-108@std.ewubd.edu

Md. Rihab Uddin Mridha
Department of CSE
East West University
Dhaka, Bangladesh
2016-2-60-120@std.ewubd.edu

Md Abdullah Al Mamun
Department of CSE
East West University
Dhaka, Bangladesh
2017-3-60-085@std.ewubd.edu

Date Of Submission: 08 September 2021

Abstract

The triumph of social networks and human based network channels of news and validated information are on the epitome of success. Although there are hundreds of benefits but the other side of the coin does have some intrusive associations too. Fake news is intentionally written to mislead readers to believe false information, which makes it difficult and nontrivial to detect based on news content; therefore, we need to include auxiliary information, such as user social engagements on social media, to help make a determination with little or no knowledge about how tricks are staged people often gets endeavored to accept news and other information that has fake visuals in the form of texts. The following research studies converged on how these fraudulent activities can be figured out how people can be asserted with authentication in order to differentiate between the news carrying different visual representations of texts and governing features beneath that are either fake or real.

Keywords: fake-news, machine learning, random forest algorithm, naive bayes, logistic regression, Gradient Boosting Classifier. Fake news detection.

List of Figures & Tables

Introduction

1.1 Introduction

Fake News the name says it all. As we are indulging more and more into the hands of things broadcasted or posted online the more is the probability increasing of getting stumbled by news that are deceit in terms. Every day hundreds of thousands of news are getting posted both by organizations and people in singular having zero or none authentication. This is because the internet is free and an open source for everyone. So if someone feels like spreading fake news while having a cup of coffee on his breakfast table with a headline “Trump declared war against China” it’s nothing impossible. Just a few lines of texts and an intriguing image can engage millions of readers across the globe to drool over it without even knowing that it’s true or not.

So what can be the consequences of this? What are the aftermaths of fake news getting shared and spread? Well, the sequel is indeed in cases very remorsing and meanwhile devastating. It can within a few minutes give rise to a propaganda leading to national massacres or even social demoralization havoc. The end result can be counted in ways that are countless to even take into account. This is why fake news detection techniques in this era are given the utmost priority because news with false contents are counted evil.

The following paper focuses on how different machine learning techniques can be used to scrutinize information that can be inherited for fake news detection [1]; How data mining approaches can be used to figure out the necessary clustering for differentiating keywords [2] and predicting early warnings to help sorting out hoaxes [3]. As the technology is pioneering the unreachable mountain tops with the help of social networking, news sites and similar blog forums the more the chance of getting conned are in birth queues too. In this era, writing anything and posting before the world has turned out to be free, independent and fast right at finger tips [1]. Though, this advantage of having faster access to information is weighed in the side of good, the drawbacks are quite misleading too. This is because coming across anti-social, vulnerable, harassing, duplicate, fake or propaganda-based news or information aided by tampered or post-processed texts can get readily picked into the readers mind and quickly manipulate or disassemble the healthy psychological or philosophical perspectives, opinions.

Adoption of appropriate fake news detection techniques has turned out to be an inevitable source of help in such scenarios where information is spread with illicit or ill intentions. The most recent U.S. Presidential election that was in 2016 can be considered in such areas because Americans consistently faced spreading of fake news from acquainted sources continuously [4]. That in turns of time affected the people’s view and diverted it through fake manipulations termed as propaganda. So, detection of fake news over online has been increasingly in the stack of demands over the past few years [5]-[8]. These days, texts are in maximum association to the posts made over online. Image attached posts are nearly 11 times larger than that has text only [6]. This is becoming oddly satisfying to the creators behind the malicious or infectious intentions and they reproduce visual illustrations to trick the end readers in a more convincing way. Not only that they also take help of real texts with heinously cloned information to impose heavy effects over people's minds. The data mining approaches embedded in here of the second paper [5] allows one entity to categorically divide the lexical feature, text features and a few other in order to make the process flow smoother so that peripherally actions can be applied. The third paper [3] also deals with the data mining approach for the picking out process of the fake news objects.

[5]. [17] have performed a retrospective analysis on 5 rumors and have noted that rumor detection seems to have a correlation with users-based features. Chen et al. [18] have observed that a convolutional neural network is more appropriate than a recurrent neural network for rumor detection tasks and obtain accuracy near 0.7. Zhao et al.

[20] have shown that the decision tree outperforms SVM with accuracy above 0.7. Poddar et al.[17] have proposed a neural approach to detect rumor veracity.

1.2 Problem Statement

The research focused on basically two defined scenarios with a problem staged with application of machine learning techniques of supervised learning. It states,

Fake news: In the context of microblog, a piece of fake news is a news post that is intentionally and verifiably false

The Problem Stated is as follows:

Given a set of news posts $X = \{x1; x2; : : : ; xm\}$, corresponding texts

$I = \{i1; i2; : : : ; im\}$,and labels $Y = \{y1; y2; : : : ; ym\}$, learn a classifier f that can utilize the corresponding image to classify whether a given post is fake news (yt = 1) or real news (yt = 0) , i.e. yt = f(it).

Fake News Detection on social media can be collected through textual features which in the research [1] are shown. They collected and analyzed the data by creating layers and sub-layers.

Their problem stated as follows:

Given the social news engagements E among n users for news article a, the task of fake news detection is to predict whether the news article a is a fake news piece or not, i.e., F : E ! f0; 1g such that,

F(a) =1; if a is a piece of fake news,

0; otherwise, where F is the prediction functions we want to learn.

Prediction of hoaxes, spams can also be done using suitable data mining and machine learning techniques which was mainly done in the research paper, third one

[3]. That involved data exploration and generate predictive model exploration in order to generate outcomes using content and characteristics base

1.3 Objective

-This research focuses on finding news that can be false posted on social platforms using real-life dataset

-The research also emphasized on improving the accuracy of the result obtained that had previously been found using similar algorithms.

1.4 Project Outline

Chapter 1: This chapter represents the introduction of our work, problem statement and the objectives that we have made.

Chapter 2: This chapter is about related work which has been done before.

Chapter 3: This chapter represents the proposed model of our project and how we have implemented it based on some features and the result and the analysis of our project.

Chapter 4: This chapter summarizes our work and also shows the conclusion of future work.

Chapter 2

Literature Review

Fake news has been there since before the advent of the Internet. Social media and news outlets publish fake news to increase readership or as part of psychological warfare.

We analyzed some research papers written on different methods for fake news detection which helps us to compare our result's accuracy with them.

2.1 Fake news detection using Deep Learning Method

Machine learning has played a vital role in classification of the information although with some limitations for fake news detection. Some articles reviewed various Machine learning approaches in detection of fake and fabricated news. The limitation of such approaches and improvisation by way of implementing deep learning was also reviewed. With the implementation of deep learning research and applications in the recent past, lots of research work was going to implement deep learning methods [21].

Some researchers used Natural Language Processing, Machine learning and deep learning techniques to classify the datasets. They had yielded a comprehensive audit of detecting fake news by including fake news categorization, existing algorithms from machine learning techniques.

They had explored different Machine learning models and also had explored the benefit of feature extraction, features like n-gram, TF-IDF features were extracted and used in their model. They also had explored the effectiveness of word embeddings and word2vec features in Deep Neural networks [23].

A new Korean fake news detection system using fact DB which was built and updated by human's direct judgement after collecting obvious facts. That system had received a proposition, and searched the semantically related articles from Fact DB in order to verify whether the given proposition was true or not by comparing the proposition with the related articles in fact DB. Researchers had utilized the BiMPM model which was a deep learning model for sentences matching [24].

A model named TI-CNN (Text and Image information based Convolutional Neural Network) was proposed by some researchers. By projecting the explicit and latent features into a unified feature space, TI-CNN was trained with both the text and image information simultaneously. Extensive experiments had carried on the real-world fake news datasets and also had demonstrated the effectiveness of TI-CNN in solving the fake new detection problem. The experimental results had shown that the TI-CNN could successfully identify the fake news based on the explicit features and the latent features had learned from the convolutional neurons [28].

In The University of California, some researchers had investigated automatic fake news detection based on surface-level linguistic patterns. They had designed a novel, hybrid convolutional neural network to integrate metadata with text. It had shown that, that hybrid approach could improve a text-only deep learning model. They had introduced LIAR, a new dataset for automatic fake news detection which had shown that when combining meta-data with text, significant improvements could be achieved for fine-grained fake news detection [30].

2.2 Fake News detection using traditional machine learning classifiers

There are researches that used sentiment analysis for fake news detection. It incorporated sentiment as an important feature to improve the accuracy. It also investigated the performance of the proposed method using three different data sets. This paper analyzed different text preprocessing techniques and selected tf-idf with similarity score as the best approach using accuracy as an evaluation metric. Also it had enriched the merged data set using sentiment to increase the accuracy of fake news detection. The result showed that the proposed solution was performing well [22].

The results of a fake news identification study that documents the performance of a fake news classifier were presented by some researchers. The resultant process precision was 63.333% effective at assessing the likelihood that an article with quotes was fake. That process was called influence mining and that novel technique was presented as a method that could be used to enable fake news and even propaganda detection. That paper had presented the results of a study that produced a limited fake news detection system [25].

Some researchers proposed Hierarchical Discourse-level Structure for Fake news detection. HDSF had learned and constructed a discourse-level structure for fake/real news articles in an automated and data-driven manner. they had identified insightful structure-related properties, which could explain the discovered structures and boosted our understating of fake news. by doing that, it had highlighted noticeable differences between structures of fake and real news documents. These had differentiate and also indicated less coherency in the fake news documents [26].

Some researchers had presented a comprehensive review of detecting fake news on social media, including fake news characterizations on psychology and social theories, existing algorithms from a data mining perspective, evaluation metrics and representative datasets. They also discussed related research areas, open problems, and future research directions for fake news detection on social media. In the detection phase, the researchers had reviewed existing fake news detection approaches from a data mining perspective, including feature extraction and model construction [27].

Some researchers had conducted a benchmark study to assess the performance of different applicable approaches on three different datasets where the largest and most diversified one was developed by them. They had performed a topic-based analysis that exposed the difficulty to correctly detect political, health and research related deceptive news [29].

Chapter 3

Proposed Methodology

3.1 Introduction

Choosing a suitable method to go on with the process bears great importance because an unsuitable method applied over the following procedure results in a whole different and non-familiar result. So, the following methods were introduced in association of how the research aims to find results on the basis of the data collected, sampled and refined for new and successful outcomes.

3.2 The Methodology of Proposed Approaches

3.2.1 Data Preprocessing

Raw texts of news required some preprocessing, before feeding into the models. We first checked whether the dataset contained any missing value or not. As our dataset had no missing value, so we have not to remove any rows. We have performed a few techniques to transfer text from human language to machine-readable format for data processing. For this, we have converted our class level text data into numeric values. After that we have performed text normalization. Text may contain unwanted spaces, single words, special characters, and numbers and depending upon the problem we face, we have to remove those numbers and special characters from text. Our text normalization includes:

- Removing all the special characters
- Removing all single characters
- Removing single characters from the start
- Substituting multiple spaces with single space
- Converting to Lowercase
- Removing prefixed 'b'
- Lemmatization

After that we have applied the Tf-idf Vectorizer. Tf-idfvectorizer converts a collection of raw documents to a matrix of TF-IDF features. The purpose of using tf-idf is to reduce the impact of tokens that appear very frequently in a given corpus and that are hence empirically less informative than features that occur in a small fraction of the training corpus.

Formula of Tfidf:

$$w_{i,j} = tf_{i,j} \times \log\left(\frac{N}{df_i}\right)$$

Where

$$tf_{i,j} = \text{number of occurrences of } i \text{ in } j$$
$$df_i = \text{number of documents containing } i$$
$$N = \text{total number of documents}$$

In our code we have written TfidfVectorizer (max_features=1500, min_df=5, max_df=0.7, stop_words='english'). Where

- **max_features:** Limit the amount of features (vocabulary) that the vectorizer will learn. For processing we have taken 1500 most occurring words as features.
- **min_df:** It is used for removing terms that appear too infrequently. We have used min_df = 5 which means “minimum number of rows that should contain this feature and ignore terms that appear in less than 5 documents”.
- **max_df:** this is used for removing terms that appear too frequently. We have used max_df=0.7, which means we should include only those words that occur in a maximum of 70% of all the rows
- **stop_words:** stop word is a commonly used word (such as "the"). We have used stop_words='english', which means that it will remove all stop words which contain in English language.

Our next step was to split training and test sets. We have divided our dataset into 20% test set and 80% training set. Finally, we have applied feature scaling by Standard Scaler library for data preprocessing, which transforms the data in such a manner that it has mean as 0 and standard deviation as 1. In short, it standardizes the data. It is useful for data which has negative values. It arranges the data in a standard normal distribution. It is more useful in classification than regression

3.2.2 Classifier Approach:

In this section, we describe the experimental setup of different models. Here we also provide some implementation details of our approaches in fake news detection.

Random Forest: We have used Random Forest as our 1st classifier. It is a supervised learning algorithm. Random forest is used for both regression as well as classification. Although it is mainly used for classification problems, this algorithm creates decision trees on data samples and then gets the prediction from each of them and finally selects the best solution by means of voting. Random Forest is a combination of decision trees. Each tree will build a random subset of a training dataset. Random forest is better than a single decision tree because by averaging the result it reduces the over-fitting. In our code we have used n_estimators=1000, here n_estimators means “the number of trees in the forest”. And we have written random_state=0, where random_state is the seed used by the random number generator.

```python
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score
from sklearn import metrics
import matplotlib.pyplot as plt
import matplotlib.cm as cm

classifier = RandomForestClassifier(n_estimators=1000, random_state=0)
classifier.fit(x_train, y_train)

#Predict the response for test dataset
y_pred = classifier.predict(x_test)

print(classification_report(y_test, y_pred))

# Model Accuracy, how often is the classifier correct?
print("Accuracy:",metrics.accuracy_score(y_test, y_pred))
```

```
              precision    recall  f1-score   support

           0       0.86      0.91      0.88       434
           1       0.89      0.82      0.85       368

    accuracy                           0.87       802
   macro avg       0.87      0.87      0.87       802
weighted avg       0.87      0.87      0.87       802

Accuracy: 0.8690773067331671
```

Fig 1: Random Forest Algorithm

KNN: Our 2nd classifier is K-nearest neighbor algorithm (k-NN). It is also used in regression and classification problems. In both cases, the input consists of the k number of neighbors (closest training examples in the feature space). KNN classifier is an instant based learning which is done by approximation. This classifier searches through the entire train dataset for the most K similar instances and the data with the most similar instances finally returns as prediction. In this algorithm we have to set the neighbors value (value of K), where 'k' neighbors is the most occurring feature in observation.

```python
from sklearn.neighbors import KNeighborsClassifier
classifier = KNeighborsClassifier(n_neighbors=5)
#k=5
classifier.fit(x_train, y_train)

#Predict the response for test dataset
y_pred = classifier.predict(x_test)

print(confusion_matrix(y_test, y_pred))
print(classification_report(y_test, y_pred))

# Model Accuracy, how often is the classifier correct?
print("Accuracy:",metrics.accuracy_score(y_test, y_pred))
```

```
[[355  79]
 [151 217]]
              precision    recall  f1-score   support

           0       0.70      0.82      0.76       434
           1       0.73      0.59      0.65       368

    accuracy                           0.71       802
   macro avg       0.72      0.70      0.70       802
weighted avg       0.72      0.71      0.71       802

Accuracy: 0.713216957605985
```

Fig 2: KNN

SVM Kernels: After applying KNN, we have applied Support Vector Machine (SVM) as our classifier. SVM is a supervised learning method that looks at data and sorts it into one of two categories. The purpose of implementing SVMs is to find the best line in two dimensions or the best hyperplane in more than two dimensions to separate our space into classes. In machine learning, kernel methods are a class of algorithms for pattern analysis, whose best known member is the support vector machine (SVM). The general task of pattern analysis is to find and study general types of relations (for example clusters, rankings, principal components, correlations, classifications) in datasets. Kernel methods require only a user-specified kernel.

There are many types of SVM Kernels. In our dataset we have applied Polynomial Kernel, Gaussian Kernel, and Sigmoid Kernel.

Naive Bayes: Naive Bayes algorithms perform well in most of the complex real world problems. This classifier assumes that the existence of a specific feature in a class is unrelated to the existence of any other feature Naive Bayes classifiers are a collection of classification algorithms based on Bayes' Theorem. They deal with probability distribution of variables in the dataset and predicting the response variable of value. There are three types of Naïve-Bayes algorithm.

- Multinomial Naive Bayes ,
- Gaussian Naive Bayes,
- And Bernoulli Naive Bayes.

```
from sklearn.naive_bayes import BernoulliNB

clf = BernoulliNB()
clf.fit(x_train, y_train)
#Predict the response for test dataset
y_pred = clf.predict(x_test)

print(confusion_matrix(y_test, y_pred))
print(classification_report(y_test, y_pred))
# Model Accuracy, how often is the classifier correct?
print("Accuracy:",metrics.accuracy_score(y_test, y_pred))
```

```
[[315 119]
 [ 45 323]]
              precision    recall  f1-score   support

           0       0.88      0.73      0.79       434
           1       0.73      0.88      0.80       368

    accuracy                           0.80       802
   macro avg       0.80      0.80      0.80       802
weighted avg       0.81      0.80      0.80       802

Accuracy: 0.7955112219451371
```

Fig 3: Naive Bayes

Logistic Regression: Like all other regression, Logistic Regression is also a predictive analysis. It is worked well when the dependent variable is binary. This regression is used to describe data and to explain the relationship between one dependent binary variable and one or more nominal, ordinal, interval or ratio-level independent variables.

```
from sklearn.linear_model import LogisticRegression

logreg = LogisticRegression()
logreg.fit(x_train, y_train)

y_pred=logreg.predict(x_test)


print(confusion_matrix(y_test, y_pred))
print(classification_report(y_test, y_pred))
# Model Accuracy, how often is the classifier correct?
print("Accuracy:",metrics.accuracy_score(y_test, y_pred))
```

```
C:\Users\Mobin\Anaconda3\lib\site-packages\sklearn\linear_m
ged to 'lbfgs' in 0.22. Specify a solver to silence this wa
  FutureWarning)
```

```
[[373  61]
 [ 83 285]]
              precision    recall  f1-score   support

           0       0.82      0.86      0.84       434
           1       0.82      0.77      0.80       368

    accuracy                           0.82       802
   macro avg       0.82      0.82      0.82       802
weighted avg       0.82      0.82      0.82       802

Accuracy: 0.8204488778054863
```

Fig 4: Logistic Regression

Linear Discriminant Analysis (LDA): Linear Discriminant Analysis (LDA) works as a classifier. It is a most commonly used dimensionality reduction technique in supervised learning.
It means that it reduces the number of dimensions (i.e. variables) from a dataset.

```
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis

lda = LinearDiscriminantAnalysis()
lda.fit(x_train, y_train)


y_pred=lda.predict(x_test)


print(confusion_matrix(y_test, y_pred))
print(classification_report(y_test, y_pred))
# Model Accuracy, how often is the classifier correct?
print("Accuracy:",metrics.accuracy_score(y_test, y_pred))
```

```
[[363  71]
 [ 98 270]]
              precision    recall  f1-score   support

           0       0.79      0.84      0.81       434
           1       0.79      0.73      0.76       368

    accuracy                           0.79       802
   macro avg       0.79      0.79      0.79       802
weighted avg       0.79      0.79      0.79       802

Accuracy: 0.7892768079800498
```

Fig 5: LDA

```
from sklearn.linear_model import PassiveAggressiveClassifier

pac=PassiveAggressiveClassifier(max_iter=50)
pac.fit(x_train,y_train)
#Predict on the test set and calculate accuracy
y_pred=pac.predict(x_test)


print(confusion_matrix(y_test, y_pred))
print(classification_report(y_test, y_pred))
# Model Accuracy, how often is the classifier correct?
print("Accuracy:",metrics.accuracy_score(y_test, y_pred))
```

```
[[353  81]
 [ 71 297]]
              precision    recall  f1-score   support

           0       0.83      0.81      0.82       434
           1       0.79      0.81      0.80       368

    accuracy                           0.81       802
   macro avg       0.81      0.81      0.81       802
weighted avg       0.81      0.81      0.81       802

Accuracy: 0.8104738154613467
```

Fig 6: Passive Aggressive Classifier

Passive Aggressive Classifier: Passive Aggressive classifier remains passive for a correct classification outcome, and turns aggressive in the event of a miscalculation, updating and adjusting. It does not converge, unlike most other classifiers. Aim of this algorithm is to make updates that correct the loss, causing very little change in the norm of the weight vector. In our code we have set max_iter=50, this means "maximum number of iterations".

Gradient Boosting Classifier: Our last machine learning classifier is Gradient Boosting. To create a strong predictive model, it combines many weak learning models together. This classifier is becoming popular because of its effectiveness at classifying complex datasets.

```
from sklearn.ensemble import GradientBoostingClassifier


clf = GradientBoostingClassifier()
clf.fit(x_train,y_train)
y_pred=clf.predict(x_test)

print(confusion_matrix(y_test, y_pred))
print(classification_report(y_test, y_pred))
# Model Accuracy, how often is the classifier correct?
print("Accuracy:",metrics.accuracy_score(y_test, y_pred))
```

```
[[405  29]
 [181 187]]
              precision    recall  f1-score   support

           0       0.69      0.93      0.79       434
           1       0.87      0.51      0.64       368

    accuracy                           0.74       802
   macro avg       0.78      0.72      0.72       802
weighted avg       0.77      0.74      0.72       802

Accuracy: 0.7381546134663342
```

Fig 7: Gradient Boosting Classifier

Chapter 4

Experimental Setup and Result analysis

Results Analysis and Discussion

In this section, we are going to describe performance of our machine learning models. We have applied thirteen classifiers to analyze best performance of our datasets. We have calculated precision, recall, f1-score, accuracy for real and fake class and find their weighted average and macro average by their support. We have marked the best result in bold text. Following table is the average score of precision, recall, f1-score and accuracy of two datasets.

Model	Precision	Recall	F1-Score	Accuracy
Random Forest	0.80	0.80	0.80	0.80
KNN	0.65	0.63	0.61	0.63
SVM (Linear)	0.73	0.73	0.73	0.73
SVM (Polynomial)	0.71	0.54	0.42	0.54
SVM (Gaussian)	0.78	0.78	0.78	0.78
SVM (Sigmoid)	0.78	0.78	0.78	0.78
Naive Bayes	**0.82**	**0.82**	**0.82**	**0.82**
Logistic Regression	0.74	0.74	0.74	0.74
Linear Discriminant Analysis	0.77	0.77	0.77	0.77
Passive Aggressive Classifier	0.73	0.73	0.73	0.73
Gradient Boosting Classifier	0.77	0.75	0.74	0.75

Table 1: Performance of Supervised learning model from news.csv

Model	Precision	Recall	F1-Score	Accuracy
Random Forest	**0.87**	**0.87**	**0.87**	**0.87**
KNN	0.72	0.71	0.71	0.71
SVM (Linear)	0.81	0.81	0.81	0.81
SVM (Polynomial)	0.80	0.73	0.72	0.73
SVM (Gaussian)	0.85	0.84	0.84	0.84
SVM (Sigmoid)	0.81	0.81	0.81	0.81
Naive Bayes	0.81	0.80	0.80	0.80
Logistic Regression	0.82	0.82	0.82	0.82
Linear Discriminant Analysis	0.79	0.79	0.79	0.79
Passive Aggressive Classifier	0.80	0.80	0.80	0.80
Gradient Boosting Classifier	0.77	0.74	0.72	0.74

Table 2: Performance of Supervised learning model from data.csv

Above tables shows that Bernoulli Naive Bayes performs well in the news dataset. Its precision, recall, f1-score and accuracy are almost 82%. On the other hand Random Forest performs better than all other classifiers in data.csv dataset. Precision, recall, f1-score and accuracy are almost 87%.

4.1 Evaluation

4.1.1 Cross-Validation

Cross-validation is a technique for asserting how the statistical model generalizes to an independent data set. It allows us to compare different machine learning methods and get a sense of how well they will work in practice. In an ideal situation, cross-validation will produce optimum results. But in case of inconsistent data, the result may be varying drastically. We do not have to implement cross-validation manually. Python provides a simple implementation in Scikit-Learn library, which will split the data accordingly.

4.1.2 Confusion Matrix

Confusion matrix also known as “matching matrix” or “error matrix”. It is a simple way to lay out how many predicted categories or classes were correctly predicted and how many were not. It is used to evaluate the results of a predictive model with a class outcome to see the number of classes that were correctly predicted as their true class. In the confusion matrix, each column represents instances of a real class and each row represents immediate prediction.

4.1.3 Accuracy

Accuracy is a measurement of a set which describes anything that is close to its true value or state of being precise. In confusion matrix accuracy is a proportion of correct classification (true positives and negatives) from the overall number of cases.

$$Accuracy=(TP+TN)/(TP+TN+FP+FN)$$

Where,

- ·TP: True positive
- ·TN: True negative
- ·FP: False positive
- ·FN: False negative

4.1.4 Precision

Precision is a proportion of correct positive classifications (true positives) from cases that are predicted as positive. It uses positives and negatives, to measure models accuracy when making predictions.

$$Precision=TP/(TP+FP)$$

4.1.5 Recall

Recall is a proportion of correct positive classifications (true positives) from cases that are actually positive. It also uses’ positives and negatives, to measure models accuracy when making predictions.

$$Recall=TP/(TP+FN)$$

4.1.6 F1-Score

F1 score is also known as F-score or F-measure or Sørensen–Dice coefficient or Dice similarity coefficient (DSC). It is also a way to measure the accuracy for binary classification of a model. To get F1-Score we have to combine precision and recall to get an overall score to see how well our model is performing

$$F1\text{-}Score = 2*(Precision*Recall)/(Precision+Recall)$$

4.2 Dataset

A dataset is a number of related sets of data or information. We can collect news from many different sources like search engines, news agencies and social media websites. It will be a challenging task, if we collect data manually to make a dataset. That’s why we have downloaded the dataset from online. In our project we have used two datasets. One is “news.csv” and another is “data.csv”. Dataset link:

1.news.csv
https://s3.amazonaws.com/assets.datacamp.com/blog_assets/fake_or_real_news.csv
Alternate Link:
https://drive.google.com/file/d/1er9NJTLUA3qnRuyhfzuN0XUsoIC4a-_q/view
2. data.csv https://www.kaggle.com/jruvika/fake-news-detection

	Unnamed: 0	title	text	label
0	8476	You Can Smell Hillary's Fear	Daniel Greenfield, a Shillman Journalism Fello...	FAKE
1	10294	Watch The Exact Moment Paul Ryan Committed Pol...	Google Pinterest Digg Linkedin Reddit Stumbleu...	FAKE
2	3608	Kerry to go to Paris in gesture of sympathy	U.S. Secretary of State John F. Kerry said Mon...	REAL
3	10142	Bernie supporters on Twitter erupt in anger ag...	— Kaydee King (@KaydeeKing) November 9, 2016 T...	FAKE
4	875	The Battle of New York: Why This Primary Matters	It's primary day in New York and front-runners...	REAL

Fig 1: Snapshot of news.csv Dataset

	URLs	Headline	Body	Label
0	http://www.bbc.com/news/world-us-canada-414191...	Four ways Bob Corker skewered Donald Trump	Image copyright Getty Images\nOn Sunday mornin...	1
1	https://www.reuters.com/article/us-filmfestiva...	Linklater's war veteran comedy speaks to moder...	LONDON (Reuters) - "Last Flag Flying", a comed...	1
2	https://www.nytimes.com/2017/10/09/us/politics...	Trump's Fight With Corker Jeopardizes His Legi...	The feud broke into public view last week when...	1
3	https://www.reuters.com/article/us-mexico-oil-...	Egypt's Cheiron wins tie-up with Pemex for Mex...	MEXICO CITY (Reuters) - Egypt's Cheiron Holdin...	1
4	http://www.cnn.com/videos/cnnmoney/2017/10/08/...	Jason Aldean opens 'SNL' with Vegas tribute	Country singer Jason Aldean, who was performin...	1

Fig 2: Snapshot of data.csv Dataset

The news.csv dataset contains 6335 records, where 3164 are fake news and 3171 are real news. The dataset contains multiple information like: unnamed id, title, text and label. In data.csv, it contains 4009 records, where 2137 are fake news and 1872 are true news. This dataset contains URLs, Headline, Body, and Label. To detect fake news we have used the title and label of the news.csv dataset and used the Headline and Label of the data.csv dataset.

4.3 Conclusion of results and its analyses

Investigations can lead to a lot of further future work for the researchers. In the papers [2],[3] they only evaluate the proposed model on Weibo data due to the shortage of desired texts in the existing Twitter having multimedia dataset and less textual features. In future work, it is aimed to carry out the research with more exquisite dataset from the different social platforms and news sites and explore the generalization capacity of the proposed model on different datasets. In Addition to that, it is also aimed to compare the similarities and differences between the visual contents of Weibo and Twitter data. As there are already many studies focusing on fusing multi-modal information for fake news detection, it is still a challenging problem that requires further investigations. At last, it is stated that how to explain the decisions made by existing models based on multi-modal information is worth considering, since it can help them in deeper understandings of such processes and help defend fake news from spreading.

Chapter 5

Limitations and Conclusion

5.1 Limitations

From the overall analysis of our paper, we have identified that our model sustains problems because of the ambiguities or uncertainty because some text or word had both meaning in several languages as Char. If we presented it in English then it represents a character type. Another problem is also faced which occurs because of erroneous words. It happens because of the spelling mistake for example HAE (HA vs. HAI) which represents a positive statement. Alphanumeric representation of words also creates problems because it is complex to extract this type of words.

5.3 Conclusion

Fake News Detection analyses the data that resembles social interactions to pick out if it is real or fake. Here we have explored multiple Machine learning models like Naïve Bayes, K nearest neighbors, Decision tree, Random forest and Deep Learning networks. We applied different deep learning methods as well as traditional classifier algorithms of machine learning to help the whole process of finding out results from two different datasets. We also reflected the benefits of feature extraction by exploring features like n-gram, TF-IDF to extract the desired characteristics and were used in the model. We also did advent the effectiveness of word embeddings and word2vec features in Deep Neural networks.

We presented comprehensive performance reasoning for different approaches using two different datasets. We presented that the Naive Bayes with n-gram can attain analogous results to neural network-based models on a dataset with less than 100k news articles.

The negative impacts of fake news and infiltrating information carries a vital role in our everyday life. This is happening due to the high end technologies and sources of digital and social communications facilitated the spreading of any information too easily and within minutes and flashes of second travelling from one pole of earth to another. This is a reason why researchers around the world buckled up to come up with technologies in association to fraud detection algorithms that can help the humankind to prevent falling into deep wells of filthy information. Though in the million jam of online portals for news it is nearly impossible to control these single handedly because every second hundreds of people are posting things online on their own will, being the platforms free for anyone to use it. In this study we attempted to scrutinize the news articles authenticity credibilities depending on their textual features. We did compile an algorithm comprising various classification methods for text models. The performance was satisfactory because of the results being accurate to its end. In the near future, we are aiming to study the correlation between different feature extraction methods and the classifiers keeping in mind using the best classifiers that can help in availing the best outcomes. However, in order to acquire more accuracy we will be in need of applying more sophisticated algorithms which use data mining techniques for handling big data. This is because incorporating a big dataset including more sophisticated and wide range of different types of news with a higher number of labels is going to aid the chances of acquiring higher accuracies.

References

[1] P. Qi, J. Cao, T. Yang, J. Guo, and J. Li, “Exploiting multi-domain visual information for fake news detection,” *Proc. - IEEE Int. Conf. Data Mining, ICDM* , vol. 2019-Novem, no. Icdm, pp. 518– 527, 2019, doi: 10.1109/ICDM.2019.00062.

[2] K. Shu, A. Sliva, S. Wang, J. Tang, and H. Liu, “Fake News Detection on Social Media,” *ACM SIGKDD Explor. Newsl.*, vol. 19, no. 1, pp. 22– 36, 2017, doi: 10.1145/3137597.3137600.

[3] D. Henry and E. Stattner, “Predictive models for early detection of hoax spread in twitter,” *IEEE Int. Conf. Data Min. Work. ICDMW*, vol. 2019-Novem, pp. 61–64, 2019, doi: 10.1109/ICDMW.2019.00018.

[4] H. Allcott and M. Gentzkow, “Social media and fake news in the 2016 election,” Journal of Economic Perspectives, vol. 31, no. 2, pp. 211–36, 2017.

[5] K. Shu, A. Sliva, S. Wang, J. Tang, and H. Liu, “Fake news detection on social media: A data mining perspective,” ACM SIGKDD Explorations Newsletter, vol. 19, no. 1, pp. 22–36, 2017.

[6] S. Kumar and N. Shah, “False information on web and social media: A survey,” arXiv preprint arXiv:1804.08559, 2018.

[7] A. Zubiaga, A. Aker, K. Bontcheva, M. Liakata, and R. Procter, “Detection and resolution of rumours in social media: A survey,” ACM Computing Surveys (CSUR), vol. 51, no. 2, p. 32, 2018.

[8] L. Wu, F. Morstatter, X. Hu, and H. Liu, “Mining misinformation in social media,” Big Data in Complex and Social Networks, pp. 123–152, 2016.

[9] Z. Jin, J. Cao, Y. Zhang, J. Zhou, and Q. Tian, “Novel visual and statistical image features for microblogs news verification,” IEEE Transactions on Multimedia, vol. 19, no. 3, pp. 598–608, 2017.

[10] Z. Jin, J. Cao, H. Guo, Y. Zhang, and J. Luo, “Multimodal fusion with recurrent neural networks for rumor detection on microblogs,” in Proceedings of the 2017 ACM on Multimedia Conference. ACM, 2017, pp. 795–816.

[11] Y. Wang, F. Ma, Z. Jin, Y. Yuan, G. Xun, K. Jha, L. Su, and J. Gao, “Eann: Event adversarial neural networks for multi-modal fake news detection,” in Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. ACM, 2018, pp. 849–857.

[12] K. Dhruv, G. Jaipal Singh, G. Manish, and V. Vasudeva, “Mvae: Multimodal variational autoencoder for fake news detection,” in Proceedings of the 2019 World Wide Web Conference. ACM, 2019.

[13] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” arXiv preprint arXiv:1409.1556, 2014.

[14] Arkaitz Zubiaga, Ahmet Aker, Kalina Bontcheva, Maria Liakata, and Rob Procter. Detection and resolution of rumours in social media: A survey. arXiv preprint arXiv:1704.00656, 2017.

[15] Yimin Chen, Niall J Conroy, and Victoria L Rubin. Misleading online content: Recognizing clickbait as false news. In Proceedings of the 2015 ACM on Workshopon Multimodal Deception Detection, pages 15 19. ACM, 2015.

[16] Martin Potthast, Sebastian Kopsel, Benno Stein, and Matthias Hagen. Clickbait detection. In European Conference on Information Retrieval, pages 810{817. Springer, 2016.

[17] R. Dayani, N. Chhabra, T. Kadian, and R. Kaushal, “Rumor detection in twitter: An analysis in retrospect,” in *2015 IEEE International*

Conference on Advanced Networks and Telecommunications Systems (ANTS). IEEE, 2015, pp. 1–3.

[18] Y.-C. Chen, Z.-Y. Liu, and H.-Y. Kao, "Ikm at semeval-2017 task 8: Convolutional neural networks for stance detection and rumor verification," in *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017),* 2017, pp. 465–469.

[19] Z. Zhao, P. Resnick, and Q. Mei, "Enquiring minds: Early detection rumors in social media from enquiry posts," in *Proceedings of the 24th International Conference on World Wide Web* .

International World Wide Web Conferences Steering Committee, 2015, pp. 1395–1405.

[20] L. Poddar, W. Hsu, M. L. Lee, and S. Subramaniyam, "Predicting stances in twitter conversations for detecting veracity of rumors: A neural approach," in *2018 IEEE 30th International Conference on Tools with Artificial Intelligence (ICTAI)*. IEEE, 2018, pp. 65–72. rumors in social media from enquiry posts," in *Proceedings of the 24th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 2015, pp. 1395–1405.

[21] S. I. Manzoor, J. Singla and Nikita, "Fake News Detection Using Machine Learning approaches: A systematic Review," 2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI), Tirunelveli, India, 2019, pp. 230-234, doi: 10.1109/ICOEI.2019.8862770.

[22]. B. Bhutani, N. Rastogi, P. Sehgal and A. Purwar, "Fake News Detection Using Sentiment Analysis," 2019 Twelfth International Conference on Contemporary Computing (IC3), Noida, India, 2019, pp. 1-5, doi: 10.1109/IC3.2019.8844880.

[23]. R. K. Kaliyar, "Fake News Detection Using A Deep Neural Network," 2018 4th International Conference on Computing Communication and Automation (ICCCA), Greater Noida, India, 2018, pp. 1-7, doi: 10.1109/CCAA.2018.8777343.

[24]. K. Kim and C. Jeong, "Fake News Detection System using Article Abstraction," 2019 16th International Joint Conference on Computer Science and Software Engineering (JCSSE), Chonburi, Thailand, 2019, pp. 209-212, doi: 10.1109/JCSSE.2019.8864154.

[25]. T. Traylor, J. Straub, Gurmeet and N. Snell, "Classifying Fake News Articles Using Natural Language Processing to Identify In-Article Attribution as a Supervised Learning Estimator,"
2019 IEEE 13th International Conference on Semantic Computing (ICSC), Newport Beach, CA, USA, 2019, pp. 445-449, doi: 10.1109/ICOSC.2019.8665593.

[26]. H. Karimi and J. Tang, "Learning Hierarchical Discourse-level Structure for Fake News Detection," pp. 3432–3442, 2019, doi: 10.18653/v1/n19-1347.

[27]. K. Shu, A. Sliva, S. Wang, J. Tang, and H. Liu, "Fake News Detection on Social Media: A Data Mining Perspective," no. i, 2017.

[28]. Y. Yang, L. Zheng, J. Zhang, Q. Cui, Z. Li, and P. S. Yu, "TI-CNN: Convolutional Neural Networks for Fake News Detection," 2018.

[29]. J. Y. Khan, A. Iqbal, and S. Afroz, "Detection," pp. 1–14, 2019.

[30]. W. Y. Wang, "' Liar , Liar Pants on Fire ':," 2016.