Fake News Detection

A Project Report submitted in partial fulfillment of the requirements for the award of the degree of

Bachelor of Technology

in

Computer Science and Engineering

by

Pratibha 16MI508

Akanksha Gahalot 16MI511

Suraina Dhiman 16MI514

Uprant 16MI555

Under the guidance of

Dr. Lokesh Chouhan

Department of Computer Science and Engineering

National Institute of Technology Hamirpur

Hamirpur(H.P), India-177005

December 2019

Department of Computer Science and Engineering
National Institute of Technology Hamirpur
Hamirpur(H.P), India-177005

We hereby certify the work being presented in the report entitled **"Fake News Detection"** in partial fulfillment of the requirements for the award of the Degree of **Bachelor of Technology** and submitted in the Department of Computer Science and Engineering at National Institute of Technology Hamirpur(H.P.) under the guidance of **Dr. Lokesh Chouhan,** Assistant Professor, Department of Computer Science and Engineering, National Institute of Technology Hamirpur.

The matter presented in this report has not been submitted by us in any other university/institute for any award of any degree.

Pratibha(16MI508)

Akanksha Gahalot(16MI511)

Suraina Dhiman(16MI514)

Uprant(16MI555)

This is to certify that the above statement made by the candidate is true to the best of my knowledge and belief.

(Dr. Lokesh Chouhan)
Assistant Professor
NIT Hamirpur

Place: Hamirpur
Date:

The B.Tech Project Viva Examination has been held on .....

Signature of Supervisor **Signature of DBPC Convenor**

Abstract

Spreading of fake news is a social phenomenon that is pervasive at the social level between individuals, and also through social media such as Facebook and Twitter. The process of obtaining news from social media is like double edged weapon. On one hand, it is easy to access, less time consuming, user friendly, easily conveyable socially relevant news, possibility for obtaining various perspective of a single news and is being updated in every minute. On other hand, news is being manipulated by various networking sites based on private opinions or interest. Fake news is misinformation or manipulated news that is spread across the social media with an intention to damage a person, agency and organization. Due to the dissemination of fake news, there is need for computational methods to detect them. Machine learning has played a vital role in classification of the information although with some limitations. Here we will use various Machine learning approaches like naive bayes, SVM, Neural networks, CNN, LSTM in detection of fake and fabricated news and finally compare their results.

Acknowledgement

We have immense pleasure in expressing our sincerest and deepest sense of gratitude towards our guide Dr. Lokesh Chouhan for the assistance, valuable guidance and co-operation in carrying out this project successfully. We also take this opportunity to thank Head of the Department Dr. T.P. Sharma for providing the required facilities in completing this project. We are greatly thankful to our parents, friends and faculty members for their motivation, guidance and help whenever needed.

Pratibha (16MI508)

Akanksha Gahalot (16MI511)

Suraina Dhiman (16MI514)

Uprant(16MI555)

List of Figures

List of Tables

Contents

Chapter 1

Introduction

1.1 Introduction to the study

Modern life has become quite convenient and the people of the world have to thank the immense contribution of the internet technology for communication and information sharing. There is no doubt that internet has made our lives easier and access to surplus information viable [1]. But this information can be generated and manipulated by common folks in bulk and the spread of such data is reckless due to the presence of social media. Platforms like Facebook and Twitter have allowed all kinds of questionable and inaccurate "news" content to reach wide audiences without proper monitoring. Social media users bias toward believing what their friends share and what they read regardless of authenticity allow these fake stories to propagate widely through and across multiple platforms and increase their credibility[2].

Fake news is increasingly becoming a menace to our society. It is typically generated for commercial interests to attract viewers and collect advertising revenue. However, people and groups with potentially malicious agendas have been known to initiate fake news in order to influence events and policies around the world. It is also believed that circulation of fake news had material impact on the outcome of the 2016 US Presidential Election [2]. Classification of any news item /post/ blog into fake or real one has generated great interest from researchers around the globe. Several research studies have been carried out to find effect of falsified and fabricated news on masses and reactions of people upon coming through such news items. There is a reason for that: recently artificial intelligence algorithms have started to work much better on lots of classification problems

(image recognition, voice detection and so on) because hardware is cheaper and bigger datasets are available. Methodolgies that will be used for fake news detection are – naïve Bayes classifier, SVM, CNN, Decision trees, CNN-LSTM. .

1.2 Problem statement

To develop efficient machine learning algorithms to identify fake/unreliable news based on content acquired.

1.3 Goals

- Predicting that whether a news is fake or not by using various ML algorithms which will help in reducing the spread of fake news.
- Improving the trustworthiness of information in online social networks by identifying the fake news timely.

Chapter 2

Literature Review

2.1 Motivation

The easy access and exponential growth of the information available on social media networks has made it difficult to distinguish between false and true information. The easy dissemination of information by way of sharing has added to exponential growth of its falsification. Fake news can even be used to manipulate voters during election season and can affect the democracy of the entire nation, so detecting them is very important. A large body of recent works has focused on understanding and detecting fake news stories that are disseminated on social media. We try to propose a method for "fake news" detection and ways to apply it on data set available. It is also highlighted how fake news detection approaches can be used in the practice.

2.2 Related Work

Many researches have been done which address this problem of detecting fake news. In a paper by Mykhailo Granik, Volodymyr Mesyura[1] on 'Fake News Detection Using Naive Bayes Classifier' they an Achieved 74% accuracy on facebook news post dataset using naive bayes classifier but only limitation was they used only naive bayes classifier. In a paper[2] by Julio C.S.Reis, et al. on 'Supervised Learning for Fake News Detection' they achieved 86% accuracy for Random Forest which performed better than KNN and naive bayes but the limitation was they used very less data points(2000). In a paper[3] by Rohit Kumar, Kaliyar on 'Fake News Detection Using a Deep Neural Network' they

used methods like cnn, lstm, knn, naive bayes were used and best accuracy was obtained from CNN but the limitation was very less data points were used and combination of cnn-lstm was not used. In a paper[4] by Prannay S Reddy, et al. on "A Study on Fake News Detection Using Naïve Bayes, SVM, Neural Networks and LSTM" they found that best accuracy was obtained using neural network 90.62% but the limitation was they did not removed stop words before classification. In a paper by Oluwaseun Ajao, Shahrzad Zargari et al. on "Fake News Identification on Twitter with Hybrid CNN and RNN Models" they used a hybrid of CNN and LSTM achieved an accuracy of 82% but the limitation was they did not compared it with other classification method and they did not used stemming. In a paper by Supanya Aphiwongsophon, Prabhas Chongstitvatana on "Detecting Fake News with Machine Learning Method" they used SVM, naive bayes, neural networks and found that naive bayes performed best with a accuracy of 96.08% but limitation was they did not take into consideration the publisher/authors name only news content was used.

Chapter 3

Methodology

3.1 Predictive Modelling

We used Predictive modeling for making predictions since it has the method which is able to build a model and has the capability to make predictions. This method consists of different algorithms of ML that can study properties from the data used for training which is used for producing predictions. It is split in two major classes one is Regression and other is classification of patterns. Regression models are based upon analysis of the relationship that are present between trends and variable in order to make predictions about the continuous variables. Whereas, the job of classification is to assign a particular class labels to a data value as output of the prediction. Division of pattern classification is in two ways i.e., Supervised and Unsupervised learning. It is already known in supervised learning that which class labels are to be used for building classification models. In unsupervised learning, these class labels are not known. Here, we worked with supervised learning methods.

3.2 Data collection and Preprocessing

- Data Collection is a process in which information is gathered from many sources which is later used to develop the machine learning models. The data should be stored in a way that makes sense for the problem. In this project, data is taken from Kaggle site.
- Data's pre-processing basically involves methods clean and scale the data like re-

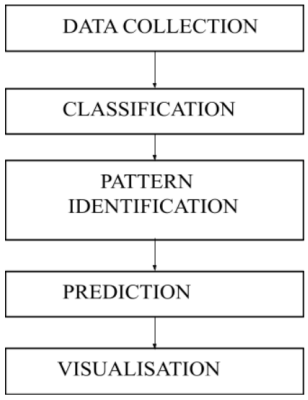

Figure 3.1: Work flow graph

moving the infinite or null values from data which might affect the performance of the model. In this step the data set is converted into a format which can be fed into machine learning models. For textual data it includes two steps.

- Text Prepocessing: The entire process of cleaning and standardization of text, making it noise-free and ready for analysis is known as text preprocessing.
- Text to Features: As the machine learning models do not understand the textual data. So they are needed to be converted into a format which is understandable by models i.e. into vectors. All of this work is done in this step.

3.3 Algorithms Used

- **Naive Bayes**

  It uses probabilistic approaches and are based on Bayes theorem. Bayes' Theorem is stated as:

  $$P(h|d) = (P(d|h) * P(h))/P(d)$$

  Where P(h—d) is the probability of hypothesis h given the data d. This is called the posterior probability, P(d—h) is the probability of data d given that the hypothesis h was true, P(h) is the probability of hypothesis h being true (regardless of the data). This is called the prior probability of h, P(d) is the probability of the data (regardless of the hypothesis). This deals with probability distribution of variables in the dataset and predicting the response variable of value. They are generally

used for text classification and used in medical diagnosis. Naive Bayes classifier model have worked well in many complicated real-world situations. An advantage of naïve Bayes classifier is that only requires less bulk of training data to access the parameters necessary for classification.

- **Support Vector Machine**

  Support Vector Machine or SVM is one of the Supervised Learning algorithms, which is used for Classification as well as Regression problems but more often it is used for Classification problems more than regression in Machine Learning. SVM algorithm have a common goal to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane, so in simpler terms. A hyperplane in an n-dimensional Euclidean space is a flat, n-1 dimensional subset of that space that divides the space into two disconnected parts. SVM aims to find the best separating hyperplane given by the equation

$$Y = Wx + b$$

  that maximizes the distance between the two classes.

- **Neural Network**

  Artificial Neural network is developed after taking inspiration from biological neuron and try simulate decision making process of human brain. It comprises of huge number of constituents which works cooperatively to process and resolve problems. It is based on prediction by analysing trends in an already existing large amount of historical data. It displays a link between an input neuron and an output neuron. Neurons have some specified weights. Output is calculated by multiplying the input with the specific neuron weight and then comparing it with the threshold value. If its above given threshold then it is contemplated as the output.

- **CNN**

  It is similar to the basic neural network. CNN also have learnable parameter like

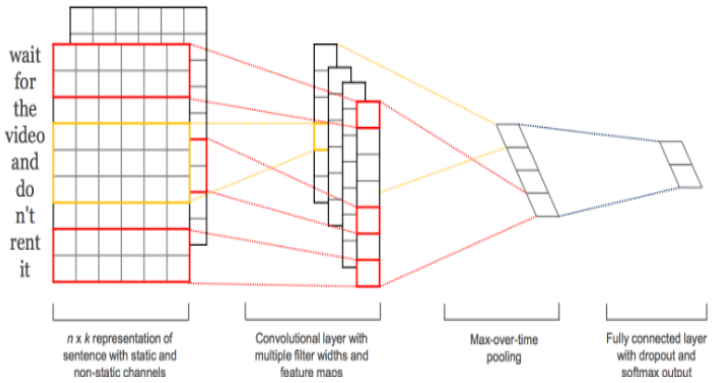

Figure 3.2: Depiction of Convolutional neural network for text Classification.

neural network i.e, weights, biases etc. It is also successfully applied to recommender systems, natural language processing and more. The main advantage of CNN compared to its predecessors is that it automatically detects the important features without any human supervision. For example, given many pictures of cats and dogs it learns distinctive features for each class by itself.

CNN is also computationally efficient. It uses special convolution and pooling operations and performs parameter sharing. This enables CNN models to run on any device, making them universally attractive.

- **LSTM**

  Long Short-Term Memory (LSTM) networks are a modified version of recurrent neural networks, which makes it easier to remember past data in memory. The vanishing gradient problem of RNN is resolved here. LSTM is well-suited to classify, process and predict time series given time lags of unknown duration. It trains the model by using back-propagation.

3.4 Training and Testing

In this step, after validating the assumptions the algorithm that we have chosen. Model is trained on the basis of given training sample. After traning, the performance of the model is checked on the basis of error and accuracy, At last, the trained model is tested with some unseen data and the model performance is checked on the basis of various performance parameters depending on the problem.

Chapter 4

Implementation

4.1 Data Collection

The datasets used for this project were drawn from Kaggle [7]. The training dataset has about 16600 rows of data from various articles on the internet. We had to do quite a bit of pre-processing of the data, as is evident from our source code [4], in order to train our models. A full training dataset has the following attributes:

- id: unique id for a news article
- title: the title of a news article
- author: author of the news article
- text: the text of the article; incomplete in some cases
- label: a label that marks the article as potentially unreliable
  - 1: unreliable
  - 0: reliable

A	B	C	D	E
id	title	author	text	label
1	FLYNN: Hillary Clinton, Big	Daniel J. F	Ever get the feeling your life circles t	0
2	Why the Truth Might Get Y	Consortiur	Why the Truth Might Get You Fired	1
3	15 Civilians Killed In Single	Jessica Pu	Videos 15 Civilians Killed In Single	1

Figure 4.1: data set used

4.2 Data Preprocessing

This basically involves two steps:

1. **Text Preprocessing:** Since, text is the most unstructured form of all the available data, various types of noise are present in it and the data is not readily analyzable without any pre-processing. Following steps has been performed to clean data.

```
def textClean(text):
    text = re.sub(r"[^A-Za-z0-9^,!.\/'+-=]", " ", str(text))
    text = text.lower().split()
    stops = set(stopwords.words("english"))
    text = [w for w in text if not w in stops]
    text = " ".join(text)
    return (text)
```

Figure 4.2: data cleaning steps

   - Removing the stopwords. A stop word is a commonly used word (such as “the”, “a”, “an”, “in”).We would not want these words to take up space in our database, or taking up valuable processing time. For this, we can remove them easily, by storing a list of words that you consider to stop words.
   - Removing empty/null cell. Data sets are not perfect. Sometimes they end up with invalid, corrupt, or missing values. Therefore it is advisable to remove them.
   - Deleting special characters and punctuation. We may want the words, but without the punctuation like commas and quotes.Converting all text to lower-case.

```
text_model = Doc2Vec(min_count=1, window=5, vector_size=vector_dimension, sample=1e-4, negative=5, workers=7, epochs=10,
                     seed=1)
text_model.build_vocab(x)
text_model.train(x, total_examples=text_model.corpus_count, epochs=text_model.iter)
```

Figure 4.3: Text to feature conversion with Doc2vec

2. **Text to features:** To analyse a preprocessed data, it needs to be converted into features i.e. into numeric feature vectors as the models do not understand the text

data. So for this purpose Doc2vec is used. The goal of doc2vec is to create a numeric representation of a document, regardless of its length. Text Preprocessing produces a comma-separated list of words, which can be input into the Doc2Vec algorithm to produce an 300-length embedding vector for each article. Preservation of word order information makes Doc2Vec useful for our application, as we are aiming to detect subtle differences between text documents.

4.3 Training Models

For training the data splitted in the ratio of 80 percent for training and 20 percent for testing using sklearn library. As a result we train size of around 114000 data points and test size of around 28000 data points. After trying different combinations of the parameters for each classification algorithm used, the optimal value of accuracy.

1. Neural Network: Neural network is implemented using Keras Frame work. Best results were obtained after using 3 dense layers and dropout rate of 0.5. The activation function used in the hidden layers is relu and softmax in the output layer. Maximum accuracy i.e. 92 % is obtained after using combination of sgd optimizer and categorical cross entropy loss function.

```
'''Neural network with 3 hidden layers'''
model = Sequential()
model.add(Dense(256, input_dim=300, activation='relu', kernel_initializer='normal'))
model.add(Dropout(0.3))
model.add(Dense(256, activation='relu', kernel_initializer='normal'))
model.add(Dropout(0.5))
model.add(Dense(80, activation='relu', kernel_initializer='normal'))
model.add(Dense(2, activation="softmax", kernel_initializer='normal'))

# gradient descent
sgd = SGD(lr=0.01, decay=1e-6, momentum=0.9, nesterov=True)

# configure the learning process of the model
model.compile(loss='categorical_crossentropy', optimizer=sgd, metrics=['accuracy'])
return model
```

Figure 4.4: Training of Neural Network

2. SVM: This estimator implements regularized linear models with stochastic gradient descent (SGD) learning: the gradient of the loss is estimated each sample at a time and the model is updated along the way with a decreasing strength schedule (aka learning rate). SGD allows minibatch (online/out-of-core) learning via the partial fit method. For best results using the default learning rate. Here best results were obtained after using default parameters.

```
clf = SVC()
clf.fit(xtr, ytr)
y_pred = clf.predict(xte)
m = yte.shape[0]
n = (yte != y_pred).sum()
print("Accuracy = " + format((m-n)/m*100, '.2f') + "%")
```

Figure 4.5: Training of SVM

3. CNN: As described earlier CNN includes three parting convolution layer, pooling layer and dense layers. Here best results were obtained using one convolution, one pooling layer and 4 dense layers and softmax activation. However the accuracy obtained by this method is less i.e. 72.70 %. THere is a scope of improvment if we can tune hyper parameters more efficiently.Figure 4.6 shows its implementation.

```
    layers.Conv1D(128, 5, activation='relu',input_shape=(300,1)),
    layers.GlobalMaxPooling1D(),

    # part 2: classification
    layers.Dense(128, activation='relu'),
    layers.Dense(128, activation='relu'),
    layers.Dense(100, activation='relu'),
    layers.Dense(2, activation='softmax')

])
model.compile(loss=j[n],optimizer=a[i],
```

Figure 4.6: Training of CNN

4. Naives Bayes: Naive Bayes methods are a set of supervised learning algorithms based on applying Bayes' theorem with the "naive" assumption of conditional independence between every pair of features given the value of the class variable. GaussianNB implements the Gaussian Naive Bayes algorithm for classification. Figure 4.7 shows its implementation.

5. LSTM : Our first layer will be an Embedding layer. We feed our embedding matrix which we've created earlier to the embedding layer and set trainable to false. This is because we are using the pre-trained weights vector and we don't want to train it again. After creating our embedding sequences, it's time to define the LSTM layer which returns the sequence . Our final layer is a dense layer which is of length 1 as is of one length.

```
gnb = GaussianNB()
gnb.fit(xtr,ytr)
y_pred = gnb.predict(xte)
m = yte.shape[0]
n = (yte != y_pred).sum()
print("Accuracy = " + format((m-n)/m*100, '.2f') + "%")   # 72.94%
```

Figure 4.7: Training of Naive Bayes

```
embedding_vecor_length = 32
model = Sequential()
model.add(Embedding(top_words+2, embedding_vecor_length, input_length=max_review_length))
model.add(LSTM(100))
model.add(Dense(1, activation='sigmoid'))
model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])
print(model.summary())
model.fit(X_train, ytr, validation_data=(X_test, yte), epochs=epoch_num, batch_size=batch_size)
```

Figure 4.8: Training of LSTM

4.4 Model Evaluation and Metrics

This section shows the results that we have obtained after implementing Naive Bayes, Neural Networks, SVM, CNN, LSTM which includes its test accuracy and confusion matrix.

Models	**Accuracy**
Naive Bayes	72.04 %
Neural Network	92.66 %
SVM	88.42 %
CNN	72.70 %
LSTM	51.89 %

Table 4.1: Comparing Accuracies of Models

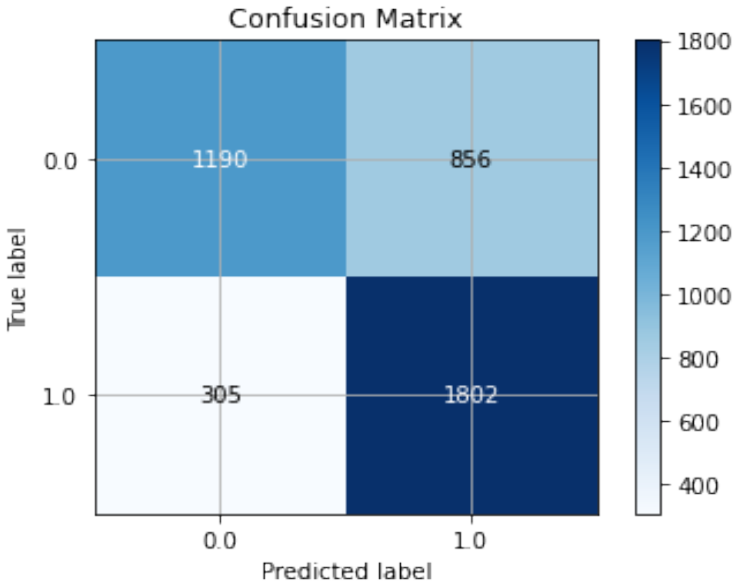

Figure 4.9: Confusion Matrix for Naive Bayes

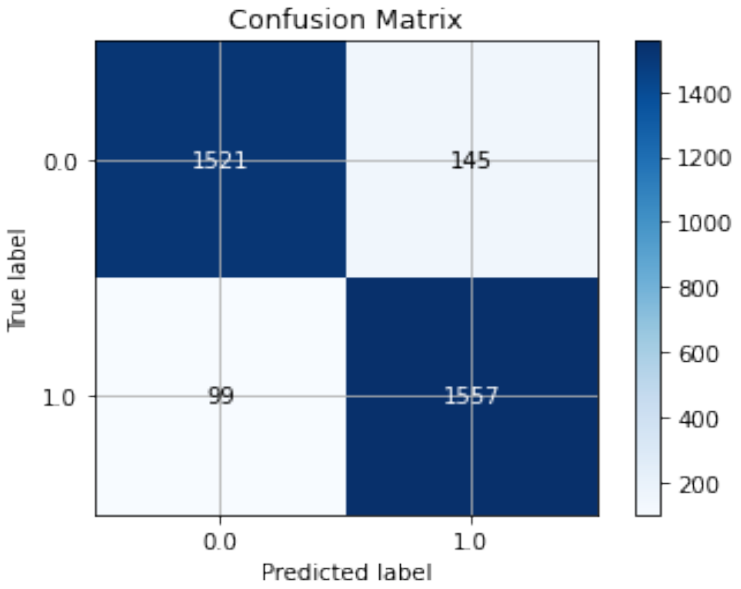

Figure 4.10: Confusion Matrix for Neural Network

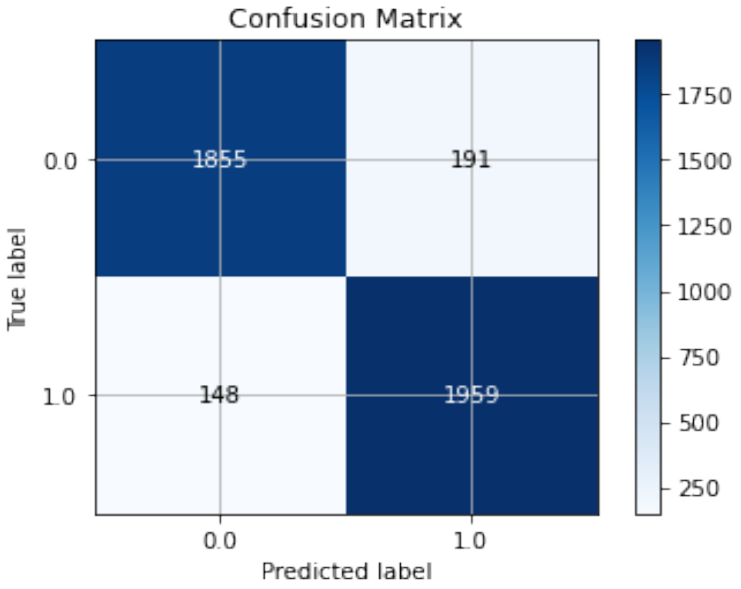

Figure 4.11: Confusion Matrix for SVM

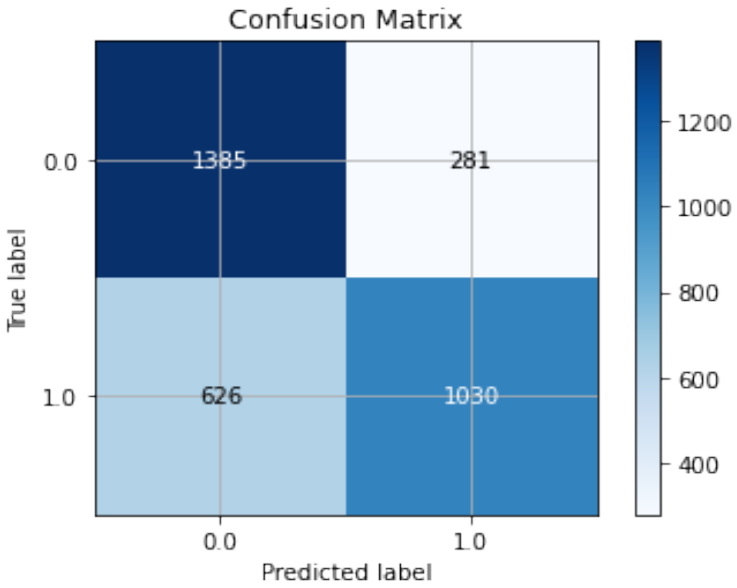

Figure 4.12: Confusion Matrix for CNN

Chapter 5

Discussion and Conclusion

In recent years, deception detection in online reviews fake news has an important role in business, law enforcement, national security, political due to the potential impact fake reviews can have on consumer behavior and purchasing decisions. After doing literature survey we found that various classification algorithms can be used for classifying fake news from the real one like Naive bayes, Decision trees, SVM, Neural networks, LSTM, CNN and we saw that in different cases different algorithms performed better. But from our study we found out that each of them had some drawback. Some of them used very less data, some did not removed stop words, some did not compare their algorithm with other classification algorithms. So in our work, we tried to remove all these drawbacks and compare the results obtained after implementating different ML algorithm and the drawbacks and which performed better. From the results it is clear that best results were obtained with help of Neural Network with Keras. Hence, should be used for classification. However which algorithm will work best is also dependent on the dataset that we are using.

References

[1] Granik, Mykhailo, and Volodymyr Mesyura. "Fake news detection using naive Bayes classifier." Electrical and Computer Engineering (UKRCON), 2017 IEEE First Ukraine Conference on. IEEE, 2017

[2] Julio C. S. Reis, André Correia, Fabrício Murai, Adriano Veloso, and Fabrício Benevenuto. 2019. Supervised Learning for Fake News Detection. IEEE Intelligent Systems, Vol. 34, 2 (2019)

[3] 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.

[4] Prannay S Reddy, Diana Elizabeth Roy, P. Manoj, M. Keerthana and Poonam V Tijare,"A Study on Fake News Detection Using Naïve Bayes, SVM, Neural Networks and LSTM",2018 IEEE International Students' Conference on Electrical, Electronics and Computer Science (SCEECS), Bhopal, 2018, pp. 1-5.

[5] Oluwaseun Ajao, Deepayan Bhowmik, Shahrzad Zargari, "Fake News Identification on Twitter with Hybrid CNN and RNN Models",2017 IEEE 15th Student Conference on Research and Development (SCOReD), Putrajaya, 2017, pp. 110-115

[6] S. Aphiwongsophon and P. Chongstitvatana, "Detecting Fake News with Machine Learning Method," 2018 15th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), Chiang Rai, Thailand, 2018, pp. 528-531.

[7] Datasets, Kaggle, https://www.kaggle.com/c/fake-news/data, February 2018.

[8] https://miro.medium.com/max/1052/1*VHDtVaDPNepRglIAv72BFg.jpeg