Fake News Recognition using Machine Learning and Natural Language Processing

Abdus Sattar, Probir Bhowmik, S. R. Sakib Ahmod, Rishad Amin Pulok, Shamsuzzaman Miah, Raihana Zannat, Ohidujjaman

[1,2,3,4,5,7]*Department of Computer Science and Engineering,*
[6]*Department of Software Engineering,*
Daffodil International University, Dhaka-1207

Abstract— Fake news is a real menace as it quickly spreads panic among the publics. Massive spread of fake news makes negative impact on individuals and society. In this internet era people spend more time in online and for more exact in social sites over the internet. Moreover with excessive use of social sites, individuals are habituated of getting news from random social platform, online resources, news agency homepages and search engines. The most terrifying things are some people and organizations make rumors and fake news for their own interest, and social platform makes it easy to spread. Top most social sites such as twitter, facebook and instagram have huge amount of users worldwide, and using these channel's fake news not only spread but also it is laterally blowout. However users of these sites mostly believe this news since they have no prior knowledge about that topic. This study provides a computational tool to tackle this problem of quick and accurate classification of news while fake or authentic. The proposed system is to develop a model using machine learning and NLP techniques to determine the news fake or real.

Index Terms— Deep Learning, LSTM, Machine Learning, News Recognition, NLP, SVM.

1 Introduction

Fake news which is created intentionally to misguide the readers. It is a type of propaganda which is published in the form of genuine news. Through the social-platform and conventional news-media, fake news is spread all over the world. Fake news had been a problem from a long time. With the introduction of various social-media platform, the extent of false news is increased and it became difficult to differentiate among actual and made-up newscast. And spread of false news is a matter of concern as it manipulates the public opinions. During the American Presidential elections of 2016, more than one million influential false news posted in social media like tweeter, Facebook and so many other online platform, that has been construct huge impact on national election. On individuals and society, massive spread of fake news makes negative impact. Related to our research has been done on autonomous recognition of misleading contents, that has been explored area of fundraising website, customer review site, online marketing, blog and forum, dating websites and so many other related platform. The lingual trail like autonomous or negative & positive individual sentence and word has been used to determine truthful and storyteller linguistic clues. Analysis of self-mentions, amount of word, effect, longitudinal and chronological infomation related to fake contents. In many application, accumulating and arrangement of text has vital role-play, for example filtering spam, data recovery, web scraping. Machine learning algorithms like K-means and logistic regression is the core of these application [7]. To represent as a fixed-length vector, algorithms frequently need input text. Apparently mostly frequent representation of fixed-length vector for texts are naïve bayes, SVM, Keras, η-grams and LSTM (Herris, nineteen fifty-four) because it is effective, simple and frequently give more accurate result. There are three datasets from Kaggle for fake news recognition, to train the models [2]. Along with an additional data set with the flood of news rising from online content creators, as well as several formats and categories, it is impossible to verify news using traditional fact checkers and vetting. To tackle this problem of quick and accurate classification of news as fake or authentic, we provide a computational tool. The proposed system is to develop a model using machine learning and NLP techniques to determine the news is real or fake.

2 Data Processing

Preprocessing of data is prepared to transform the raw and unstructured data into a required format. Data pre-processing can be done by numerous approaches like data cleaning, data reduction, data integration etc. In this study, the datasets are collected from different resources which have different formats and attributes. Hence, the data can be identical and they may contain some attributes which are not useful. However, it transforms the data into the required format with essential attributes which are used to train our model. For the purpose of training, we use data from kaggle's pre-trained dataset. There is putted collected data into algorithm to train it, and then it passes through the model. Moreover there input some data through pre-trained algorithm, it can assume whatever the data is real or fake. After this process, it needs to evaluate the result for finest prediction as shown in Fig. 1.

2.1 Generating Feature Vector

The most important part of detecting a given news is fake or not; it needs to convert the news article into a news vector which contains the important features are used to determine the nature of the news. There are several ways to generate this feature vector [3]. There are different approaches for the same to determine which method gives the best accuracy. Some of the methods such as Bag-of-words, TF-IDF are discussed for the further discussion.

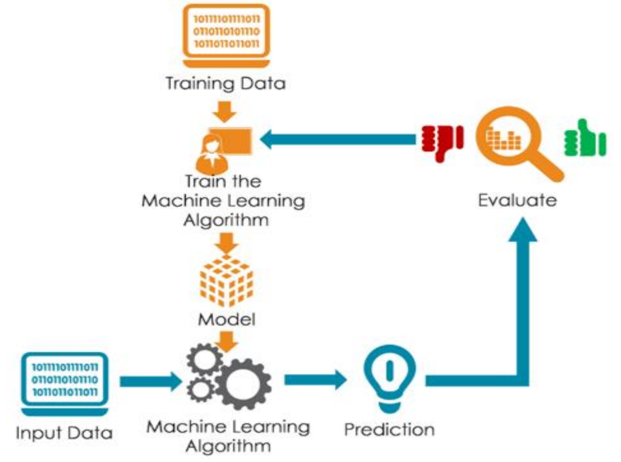

Fig. 1. Training model

2.2 Bag of Words

A method named Bag-of-words [8] represent text in a format which can be easily processed by the machine learning algorithms [9]. BoW is one of the ways of extracting the features from existing text. In this type of text representation mainly two things are involved such as known words vocabulary and presence measure of known words.

2.3 TF-IDF

The short form of term-frequency and inverse document frequency is TF-IDF [10]. Term frequency-inverse document is a method applied for represent text in a format which is easily processed by the machine learning algorithms. A statistic of numerical data demonstrates the importance of a significant word in the documents for word principal. The amount of time a words exist in a certain article is corresponding to the dominance of that word; however inversely proportional to number of time the term seems in principal.

TF: Term-frequency is distinct as the frequency of a word in the documents. TF is calculated as:

TF(w) = (No. of time word ‘w’ appear in the documents) / (Total no of words in document).

IDF: This determines the importance of words contains in the documents. For example, words such as and, of, the, appears lot of times but they are less significant. Thus most repeated terms are given less weights and less frequent terms are given more weights. IDF is calculated as: IDF (w) = $\log_e$(total amount of papers / amount of papers with word ‘w’ in it). The TF-IDF weight is given to each word by calculating TF*IDF values. For generating the news vector, there is analyzed the TF and IDF values of the bigrams and represent the TF-IDF vector of that bigrams. It is chosen bigrams over unigrams because it gives the context η –grams. N-gram is connecting arrangement of η terms from a given text. N-gram of extent one is devoted as uni-gram; dimension two is a bi-gram, size three as a tri-gram and so on. With greater n a model can store extra perspective.

$$w_{i,j} = tf_{i,j} * log(N/df_i) \text{----------------------------}(1)$$

$tf_{i,j}$= number of occurences of I in j
df_i= number of documents containing i
N=total number of documents

2.4 Combining Features to Form Final News Vector

There are considered 3 methods for generating feature vectors:

- ✓ TF-IDF bigram vector of the news article.
- ✓ Feature vector generated by semantic analysis of news article.
- ✓ Feature vector created by syntax analysis of the news article.

Afterward producing these features and generating their individual feature vector, there have to combine these features to form the final news vector on which classification is performed. The method that approached for combining the feature vectors is

- ✓ Take the most important features for the 3 feature vectors.
- ✓ Assign weights to each vector and then take the weighted combination of the 3 feature vectors to generate the final feature vector. If x is the weight corresponding to the first feature vector, y for the second, and 1-x-y for the third. The final feature vector will be the linear combination of these feature vectors multiplied by their corresponding weights.

3 Classification

After generating the news feature vector, it classifes the vector to whether it is fake or real. There is an aim to use the following classification algorithms such as Naïve Bayes and Support vector machines for the purpose of classification.

3.1 Naive Bayes

Naïve bayes is an algorithm for supervised learning which is used for the classification [5]. This is constructed on bayes theorem assuming that features are autonomous of each other. It calculates the probability of every classes, and the classes with maximum possibility is chosen as output.

Text classification: For text classification, classifiers named Naive Bayes is mostly used. By comparison with other algorithms, Naive Bayes provides a more accurate result because of success rate, it is beastly used in many field like sentiment inquiry and spam filtering (spam e-mail identify). Naive Bayes model is appropriate for working with very vast data sets, and it is not that complex to build. However for filtered calcifications it performs outstanding along with simplicity. For calculating subsequent probability $P(c|x)$ from $P(c)$, $P(x)$ and $P(x|c)$, bayes theorem provides a way by the formula given below:

$P(c/x) = P(x/c)*P(c)/P(x)$ -------------------------(2)

P(c/x) =Posterior probability
P(x/c) =Likelihood
P(x) =Predictor prior probability
P(c) =Class prior probability

$P(c/x) = P(x_1/c) * P(x_2/c) * P(x_3/c) * \text{-----} * P(x_n/c) * P(c)$ --------(3)

3.2 Support Vector Machine (SVM)

Support vector machine is a model for controlling learning with algorithms of associated study, which applied for the application of analyzing data for cataloging purpose and enquiry of regression [3]. This model cannot perform as expected way and its can perform in line sorting. Barnhard Bosarc, M. Gaynor and N. Vepnik later invent kernel terms, the able Support vector machine to perform in non linear classification. With support of this model SVM acquired more power.

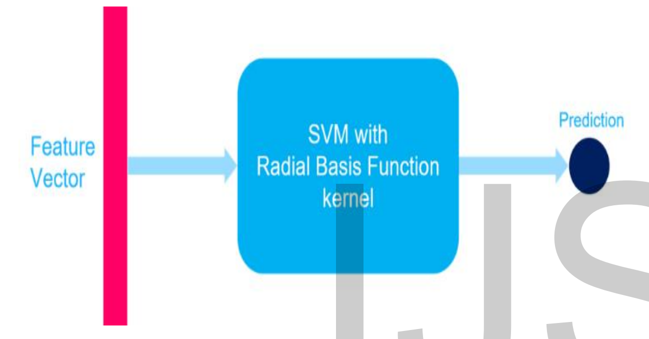

Fig. 2. Diagram of Support Vector Machine

Neural Network NN is bunch of algorithms, modeled like human brain and able to work loosely, which is planned to propose of recognize outlines [4]. There construe sensory facts over a kind of mechanism insight, tagging or gathering fresh response. For purpose of the data processing there use Tensorflow and Keras [8]. Tensorflow has hidden layer structure (300, 300, 300, 300, 300) and (256, 256, 80) for Keras [11]. Learning rate of Tensorflow is 0.001 and learning rate of Keras is 0.01. Training steps of Tensorflow is 20000 and 10000 for keras.

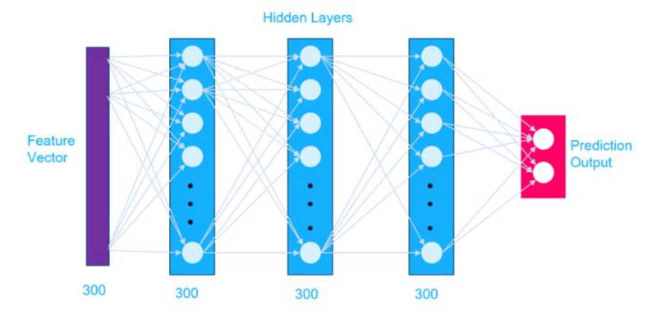

Fig. 3. Neural Network

4 Experimental Result

There are numerous experimentations with diverse groupings of feature groups to discover the analytical distinctly and together. There is used a rectilinear classifier named SVM and bearing our assessments by 5-fold cross authentication with accurateness, exactness, recollection as presentation metrics. The machine learning algorithms are used in execution accessible as open-source with the defaulting factors. After generating the feature vectors, there were combined with different weights. The results are compiled in the following table 1:

TABLE 1
Weights of the Feature Vectors and Corresponding Results

Bigrams Vector	Syntax Vector	Semantic Vector	Multinomial Naïve Bayes	Random Forests	Gradient Boosting
1	0	0	82.9%	84.7%	86.5%
0	1	0	66.5%	86.6%	88.3%
0.33	0.33	0.33	91.2%	86.0%	91.4%
0.5	0.5	0	92.2%	88.4%	91.4%
0.5	0	0.5	90.5%	80.3%	85.8%
0	0.5	0.5	89.4%	79.9%	88.3%
0.2	0.2	0.6	87.8%	86.2%	90.8%
0.2	0.4	0.4	89.8%	85.5%	90.8%
0.2	0.6	0.2	90.9%	85.6%	90.9%
0.35	0.5	0.15	**92.7%**	89.7%	91.5%
0.4	0.2	0.4	91.1%	85.0%	91.3%
0.4	0.4	0.2	92.1%	86.3%	91.4%
0.6	0.2	0.2	91.6%	86.5%	91.3%
0.4	0.5	0.1	92.3%	87.3%	91.4%

matrix corresponding to the weights are equal weightage such as 0.333, 0.333, and 0.333.

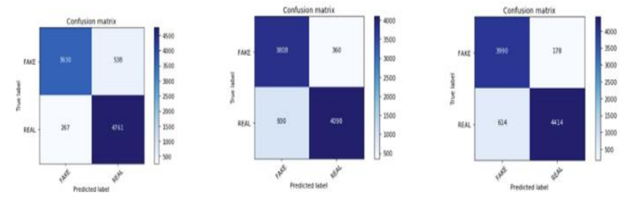

Fig. 4. Confusion matrices for Naive Bayes, Random Forests and Gradient Boosting respectively for equal accuracy

Confusion matrix corresponding to the weights (0.35, 0.5, and 0.15) which give maximum accuracy as shown in figure 5.

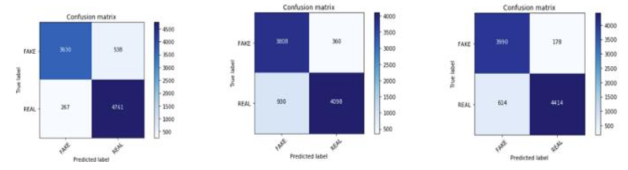

Fig. 5. Confusion matrices for Naive Bayes, Random Forests and Gradient Boosting respectively for maximum accuracy

To calculate the Precision, the F1 scores and Recall, it matched the model using confusion matrices results are shown in table 2.

TABLE 2
MODEL PERFORMANCE ON THE TEST SETS

Name	Precision	Recall	F1
Naive Bayes	0.68	0.86	0.76
SVM	0.85	0.93	0.89
Tensorflow	0.77	0.92	0.84
Keras	0.92	0.93	0.92
LSTM	0.94	0.94	0.94

The average accuracy rate of each models as shown in the table 3. LSTM and Keras gives more accuracy with comparison to other models.

TABLE 3
COMPARISON OF MODELS

Model	Accuracy
Naive Bayes	72.94%
SVM	88.42%
Neural Network using Tensor flow	81.42%
Neural Network using Keras	92.62%
LSTM	94.53%

5 CONCLUSION

For classifying uncertain and confirmed news tweets and assume exact forms of suspicious news there make lingual imparted NN model which equally acquire after specific social network connections and contents. There is observed that all the 3 features are paramount in detecting fake news when combined together. It is achieved the best result with an accuracy of 92.7% by using the weights (0.35, 0.5, 0.15) for feature vectors derived by bigrams, syntax and semantic analysis. Thus it concludes that the linguistic features are pivotal in detecting whether a news is real or fictitious. Using advanced dissertation and pragmatic structures, and concluding aspect of reliability is on focused for future works.

REFERENCES

[1] Kai Shu and Huan Liu, ”Detecting Fake News on Social Media”, Synthesis Lectures on Data Mining and Knowledge Discovery, July 2019, 129 pages.

[2] K. Shu, A. Sliva, S. Wang, J. Tang, and H. Liu, “Fake news detection on social media: A data mining perspective,” CoRR, vol. abs/1708.01967, 2017.

[3] S. Feng, R. Banerjee, and Y. Choi, “Syntactic stylometry for deception detection,” Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers - Volume 2, ACL ’12, (Stroudsburg, PA, USA), pp. 171–175, Association for Computational Linguistics, 2012.

[4] N. J. Conroy, V. L. Rubin, and Y. Chen, “Automatic deception detection: Methods for finding fake news,” Proceedings of the 78th ASIS&T Annual Meeting: Information Science with Impact: Research in and for the Community, ASIST ’15, (Silver Springs, MD, USA), pp. 82:1–82:4, American Society for Information Science, 2015.

[5] V. Pérez-Rosas, B. Kleinberg, A. Lefevre, and R. Mihalcea, “Automatic detection of fake news,” CoRR, vol. abs/1708.07104, 2017.

[6] S. Gilda, “Evaluating machine learning algorithms for fake news detection,” IEEE 15th Student Conference on Research and Development (SCOReD), pp. 110–115, Dec 2017.

[7] M. Yancheva and F. Rudzicz, “Automatic detection of deception in child-produced speech using syntactic complexity features,” ACL, 2013.

[8] Richard Socher and Christopher D. Manning “Global Vectors for Word Representation Jeffrey Pennington”, Computer Science Department, Stanford University, Stanford, CA 94305.

[9] Svitlana Volkova, Kyle Shaffer, Jin Yea Jang and Nathan Hodas, “Separating facts from fiction: Linguistic models to classify suspicious and trusted news posts on twitter”, Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, Volume 2, P17-2102,July 2017, Association for Computational Linguistics

[10] Data Sciences and Analytics Group, National Security Directorate Pacific Northwest National Laboratory 902 Battelle Blvd, Richland, WA.

[11] S Sen and A Raghunathan, “Approximate Computing for Long Short Term Memory (LSTM) Neural Networks,” IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2018, 1-1.