Comparative Performance of Machine Learning Algorithms for Fake News Detection

Arvinder Pal Singh Bali(✉), Mexson Fernandes, Sourabh Choubey, and Mahima Goel

Asia Pacific Institute of Information Technology SD India, Panipat 132103, Haryana, India
hgnis.nivra@gmail.com, mexsonfernandes@outlook.com, sourabhchoubey010@outlook.com, mahima240189@gmail.com

Abstract. Automatic detection of fake news, which could negatively affect individuals and the society, is an emerging research area attracting global attention. The problem has been approached in this paper from Natural Language Processing and Machine Learning perspectives. The evaluation is carried out for three standard datasets with a novel set of features extracted from the headlines and the contents. Performances of seven machine learning algorithms in terms of accuracies and F1 scores are compared. Gradient Boosting outperformed other classifiers with mean accuracy of 88% and F1-Score of 0.91.

Keywords: Fake news · Natural Language Processing · Text classification · Machine learning algorithms · Gradient boosting

1 Introduction

The proliferation of internet and social websites have led to the exponential growth in opinion spams and fake news, in recent times.

The Fake news which diffuse faster than the real news on social websites like Google Plus, Facebook, Twitter etc. [2] have been defined as 'fabricated information' which resemble news media content 'in form but not in organizational process or intent' [1]. Ahmed et al. [3] further subdivided fake news into three distinct categories viz. false news, fake satire news and poorly written news articles.

Fake news is a growing menace in the society. Its detection is a complex task, believed to be much harder than detection of fake product reviews [3]. Fake news poses a grave threat not only to the reliability of certain media outlets but to the government and the society as well. It is reported that more than 62% of U.S. adults get their news from social media. Moreover, the volume of disseminated information and the rapidity in which it is spread in social network sites make it extremely difficult to assess its reliability in a timely manner. Identification of fake contents in the online/offline sources is an important research problem.

The objective of the paper is to compare performances of seven machine learning (ML) algorithms on three standard datasets using a novel set of features and statistically validate the results using accuracies and F1 scores.

M. Singh et al. (Eds.): ICACDS 2019, CCIS 1046, pp. 420–430, 2019.
https://doi.org/10.1007/978-981-13-9942-8_40

The remaining of the paper is organized as follows: A short review of selected papers on fake news identification is presented in Sect. 2. The datasets, the features, the experiments as well as the results are summarized in Sect. 3. In conclusion it has been argued that with an appropriate set of features extracted from the texts and the headlines, Gradient Boosting Algorithm (XGB) can effectively classify fake news with very high accuracy and F1 score.

2 Related Work

Ahmed et al. [3] introduced a new n-gram model to detect automatically fake contents, particularly focusing reviews and news. Results of two different feature extraction techniques viz. tf, tf-idf and six machine learning classification techniques were reported by the authors. Linear classifiers viz. linear Support Vector Machine (SVM), Stochastic Gradient Descent (SDG) and Logistic Regression (LR) achieved better results than nonlinear ones for both fake reviews and news. Shu et al. [4] extensively reviewed the detection of fake news on social media, from a data mining perspective, evaluation metrics and representative datasets. Horne and Adali [5] used stylistic features from Python Natural Language Toolkit, complexity and psychology features, carried out their study on three data sets viz. Buzzfeed election data set, dataset related to political news and Burfoot and Baldwin data set. The authors using SVM classifier concluded that fake news is more akin to satire than to real news. Mean accuracies achieved by the authors were for 78% for the title and 71% for the contents, for a political news data set specifically designed by the authors. Horne et al. [6] presented the NELA2017 dataset, which contains articles from 92 news sources over 7 months, as well as, 130 content-based features that have been used throughout the news literature.

Baly et al. [7] attempted to predict the factuality of reporting of a news, using features that were earlier proposed by Horne et al. for detecting “fake news” articles [6]. These features were used to analyze the characteristics of the article viz. Structure, Sentiment, Topic, Complexity, Bias and Morality. The authors used the features in a SVM classifier, training a separate model for factuality and for bias and reported the results for 5-fold cross-validation. The paper, based on a project sponsored by AI laboratory MIT, demonstrated a new system using ML to determine the accuracy/political bias of a source [11]. Pérez-Rosas et al. [8] introduced two novel datasets for the task of fake news detection, covering seven different news domains. From a set of learning experiments to detect fake news the authors concluded that accuracies of up to 76% could be achieved.

Gilda [9] presented work on datasets from Signal Media and OpenSources.co, applied *tf-idf* to a corpus of nearly 11,000 articles. The test conducted using five classification algorithms viz. SVM, Stochastic Gradient Descent, Bounded Decision Trees, Random Forest and GBoost achieved accuracy of 77.2% using stochastic gradient descent.

The problem was approached by Bajaj [10] from a purely NLP perspective using Convolutional Neural Networks (CNN). The project at Stanford University envisaged to build a classifier that can predict whether a piece of news is fake, based only on its contents. Several architectures were explored, including a novel design that

incorporates an attention mechanism in a CNN. However, the results were not promising, compared to conventional ML algorithms.

Liu and Wu argued that the existing ML approaches were inadequate and proposed a model for an early detection of fake news on websites by classifying news propagation paths [12]. Experimental results demonstrate that the proposed model can detect fake news with accuracies 85% and 92% respectively on Twitter and SinaWeibo, the Chinese social website. TriFN proposed by Shu et al. [13] is a tri-relationship embedding framework, based on publisher-news relations and user-news interactions. The authors also reported performance comparison of ML algorithms with TriFN. The algorithms included LR, Naïve Bayes, Decision Tree, Random Forest, XGBoost, AdaBoost, and Gradient Boosting. On BuzzFeed and PolitiFact datasets TriFN recorded 86% and 87% average accuracies. F1 score recorded by TriFN were 0.87 and 0.88 respectively [13]. Eugenio Tacchini et al. in their technical report submitted to UCSC [14] used two classification techniques viz. LR and a novel adaptation of Boolean crowdsourcing algorithm.

3 Experimental Evaluation

3.1 Datasets Statistical Information

The model was tested on three different datasets: (i) OpenSources dataset having 9,408,908 articles out of which 11,161 articles from categories fake and reliable were selected [15] (ii) Kaggle dataset on fake news consisting 20,800 articles [16] and (iii) GitHub repository for fake or real news dataset by George McIntire, having two sections, headlines and text of the news [17]. Description of the labeled datasets is given in Table 1.

Table 1. Labeled dataset description

Dataset	Fake count	Reliable count	Total
Open sources [16]	5385	5776	11161
Kaggle dataset [17]	10413	10387	20800
George McIntire dataset [18]	3164	3171	6335

3.2 Features

For building a Machine Learning model, feature selection is of utmost importance for optimum performance of the system.

Features used in the proposed model are as follows:

3.2.1 n-grams Count Feature

These features are used for counting occurrences of n-grams in the title and body of the news, and various ratios of the unique n-gram and total word count given by Eq. (1).

$$ratio\ of\ unique\ n\text{-}gram = \frac{total\ unique\ n\text{-}gram}{total\ n\text{-}gram} \quad (1)$$

Equation (1) is the ratio of n-gram where n-gram could be unigram, bigram or trigram [3]. Thereafter it uses specific binary refuting words, like 'fake', 'fraud', 'hoax', 'false', in the headline.

3.2.2 tf-idf: Term Frequency- Inverse Document Frequency

It consists of two terms tf and idf. Term Frequency is how many times a word occurs in a given document given by Eq. (2).

$$tf(t,d) = 0.5 + 0.5.\frac{f_{t,d}}{\max\{f_{t',d}:t' \in d)\}} \quad (2)$$

Thus, tf(t, d) is raw count of term in a document $f_{t,d}$ divided by max $\{f_{t',d} : t' \in d\}$, the number of words in the document.

Inverse Document Frequency (idf) is the number of times a word occurs in corpus of documents. This facilitates to understand which words are important [3, 21]. Usually the natural log - normalization of the Inverse Document Frequencies given by Eq. (3) is used:

$$idf(t,D) = log\frac{N}{|\{d \in D:t \in d\}|} \quad (3)$$

Here, 'N' is the total number of news articles present in the corpus 'D', N = |D|, $|\{d \in D : t \in d\}|$ is the number of documents where the term 't' appears. Then *tf-idf* is calculated as: tf (t, d). idf(t, D).

Finally, Cosine similarity of these normalized tf-idf vectors are calculated for headlines and the contents. This gives the measure of how correlated the headlines and their corresponding article contents are. Since cosine similarity considers only those vectors which have non-zero dimension, its calculation is quite fast.

$$Similarity = Cos\,\theta = \frac{\boldsymbol{A}.\boldsymbol{B}}{\|\boldsymbol{A}\|\|\boldsymbol{B}\|} \quad (3)$$

3.2.3 Word Embedding

To get the vector space representation of the words, Word Embeddings are used. Word Embedding replaces each word with real valued vector. Global vector for word representation [18] is used for this task. It is trained on aggregated word-word co-occurrence counts on a given corpus. The pre-trained word vectors contained 6 billion tokens having a vocabulary of 400,000 words and represent each word as a 50-dimensional vector.

3.2.4 Sentiment Polarity Score

It is a basic task in sentiment analysis which could be used as an idea of what tone different articles follow. During dataset exploration, using IBM Watson tone analyzer [19], the fake and reliable news showed clear difference in the Anger analysis.

Therefore, making it a good choice as a feature. Using open source library Natural Language ToolKit (NLTK) [20], sentiment intensities were analyzed for positive, negative, neutral and compound sentiments.

3.2.5 Linguistic

Linguistic features [21] such as readability ease and lexical diversity, represent the texts statistically and also represent context of the sentences in terms of ease of reading it. Readability standard gives an approximation of years of education required to understand a sentence on single reading. By using Flesch-Kincaid, Gunning fog, and other features, the readability standards of the news articles are determined. Also lexical diversity of articles are calculated and used as features.

Feature Matrix: Features and their counts are summarized in Table 2 and explained in the followings:

- The tf-idf vectors and the tf-idf Cosine similarity between headlines and the contents are concatenated in a single feature vector.
- Sentiment analysis for the headings and the contents generate 8 features in total.
- There are 12 Readability features along with 41 count features.
- Total features are 163 with word embedding alone accounting to 101 features.

Table 2. Features and their counts.

Features		Number of feature vectors
Sentiment	Headline	4
	Content	4
Readability		12
Count		41
Cosine Similarity of Normalized *tf-idf* vectors between headline and content		1
Word Embedding	Headline	50
	Content	50
	Cosine similarity between Headline and Content	1
Total number of features		**163**

3.3 Dataset Preprocessing and Model Implementation

Dataset preprocessing and cleaning is one of the important tasks in Natural Language Processing [3]. The removal of extraneous information is crucial. Articles that are used, contained links, numbers, and other symbolic contents that are not required for feature analysis. The statistics of occurrence of word in a corpus is the main source of information for any NLP task. Regular Expressions are used to replace symbolic characters by words, digits by ‘number’, and dates by ‘date’. Source URLs if any are

removed from the article content. The text of headline and body are then tokenized and stemmed. Finally unigrams, bigrams and trigrams are created out of the list of tokens. These grams and the original text are used by the different feature extractor modules. The implementation and sequential flow of the classification process is shown in Fig. 1.

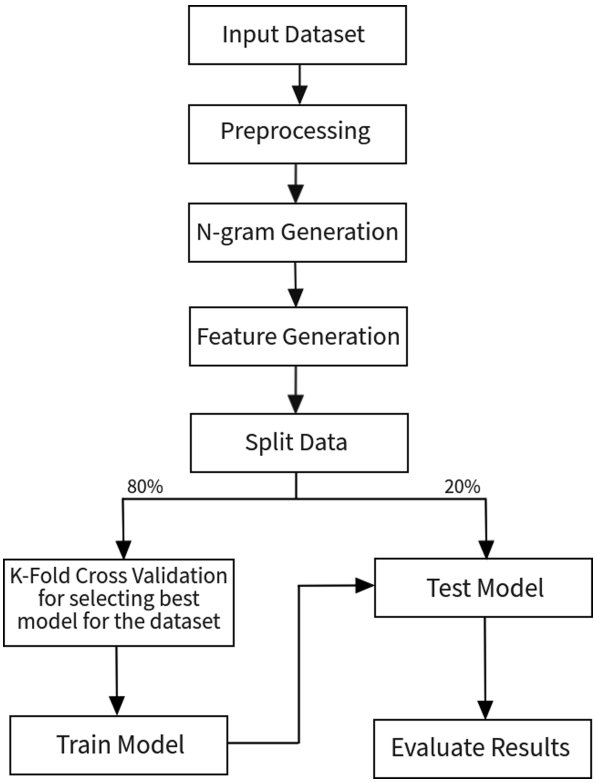

Fig. 1. Flowchart of the classification process

After feature generation, as discussed in Sect. 2, cross validation is used to find the best performing classifier. The best algorithm is then used for training on 80% dataset and rest 20% is used for testing.

3.4 Model Evaluation

Cross validation is a standard technique for assessing how the results of a statistical analysis generalize to an independent data set. To evaluate the models, stratified 10-fold cross-validation [24] was used on the complete dataset.

For model evaluation, seven Machine Learning algorithms viz. Random Forest (RF), Support Vector Classifier (SVC), Gaussian Naïve Bayes (GNB), AdaBoost (AB), K-Nearest Neighbor (KNN), Multi-Layer Perceptron (MLP) and Gradient Boosting

(XGB) [23] were selected. Accuracy and standard deviation are the evaluation metrics for comparison. Accuracy vs n-fold cross validation for the ML algorithms are shown in Figs. 2a, b and c for the three datasets.

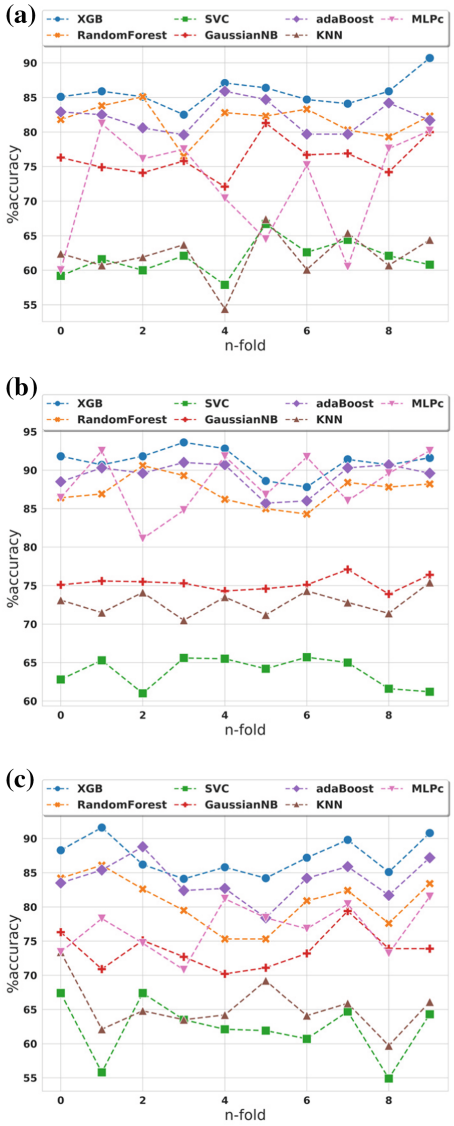

Fig. 2. (a) Accuracy vs. Cross validation (Dataset 1) (b) Accuracy vs. Cross validation (Dataset 2) (c) Accuracy vs. Cross validation (Dataset 3)

It can be readily inferred from Figs. 2a, b and c that the boosting algorithms viz. Gradient Boosting (XGB) and AdaBoost are performing better compared to other classifiers in terms of average accuracies and standard deviations.

The lowest average accuracies recorded are 62.5%, 72.54% and 64.8% when using KNN for Opensource [15], Kaggle [16] and George McIntire [17] datasets. Whereas Gradient Boost (XGB) achieved 87.2%, 92.0% and 87.3% average accuracies for these datasets respectively.

The complete model comparison is shown in Table 3.

Table 3. Model comparison

Classifier	Dataset 1		Dataset 2		Dataset 3	
	A*	SD**	A*	SD**	A*	SD**
XGB	**86.2**	**2.21**	**91.05**	1.67	**87.3**	**2.59**
RF	81.2	2.32	86.63	1.8	82.6	3.1
SVC	62.9	2.31	63.55	1.65	62	3.96
GNB	76.4	2.67	75.24	1.02	73.2	2.73
AB	81.7	2.25	89.25	1.87	83.7	2.75
KNN	62.5	3.28	72.54	1.43	64.8	3.6
MLP	67.1	6.18	88.36	9.93	72.8	6.14

*A: Average Accuracy
**SD: Standard Deviation
Note: All values are in percentage

From n-fold cross validation, the best performing ML algorithm XGB is used for training a model.

3.5 Results and Discussions

The results of the trained models on the test data are tabulated in Table 4. F1 Score defined as the harmonic mean of precision P and recall R is a measure of the accuracy of the test, in statistical analysis for binary classification. It is evident that the models performed remarkably well with XGB as the classifier.

Table 4. Classification report

Dataset	Precision	Recall	F1-score
Dataset 1	0.92	0.92	0.92
Dataset 2	0.93	0.94	0.94
Dataset 3	0.89	0.87	0.89

Finally, to analyze relative significance of the features in the proposed model, F1 Score was calculated for each dataset with exclusion of one the features.

The Loss is calculated as:

$$LOSS = Original\ F1\ Score - New\ F1\ Score \tag{5}$$

Table 5. Relevance of selected features

Feature excluded		Dataset 1	Dataset 2	Dataset 3
tf-idf	F1-score	0.9	0.94	0.89
	Loss	0.02	0	0
Count	F1-score	0.89	0.92	0.88
	Loss	0.03	0.02	0.01
Word embedding	F1-score	0.8	0.9	0.85
	Loss	0.12	0.04	0.04
Sentiment	F1-score	0.9	0.94	0.89
	Loss	0.02	0	0
Readability	F1-score	0.9	0.93	0.88
	Loss	0.02	0.01	0.01

Table 5, indicates the significance of 'word embedding' as a feature in the proposed scheme for model evaluation. The Original F1 Scores are shown in the Table 4.

While analyzing the results, it may be recalled that Ahmed, Traore and Saad [3] used tf and tf-idf as features, Baly et al. [7] used 141 features earlier used by Horne et al. [6], whereas 163 features proposed in the model viz. Cosine similarity of normalized *tf-idf* vetors, word embedding, sentiment polarity and readability are more generalized and the results Tables 3 and 4 confirm the significance of the features extracted from the news headlines and the contents in this context.

Following is the configuration of the computing environment for carrying out the experiment: Intel Processor - Quad core i7 @ 2.8 GHz, RAM - DDR4 8 GB, 2400 MHz, Nvidia GTX 1050 GDDR5 4 GB, Operating System – 64 bit Ubuntu 18.04.1 LTS, Compiler: Python ver. 3.6.6.

4 Conclusion and Future Scope

Early detection of fake news is of primary importance for public and the society, for redesigning the 'information ecosystem in the 21st century' which would eventually lead to the creation of a system and culture having values that promote truth [1]. The paper demonstrates with extensive experimentation that with a novel set of features extracted from the heading and the text, specifically the XGB classifier can efficiently detect fake news with 88% mean accuracy and 0.91 F1 score, outperforming other ML classifiers. A number of features in addition to the vectors corresponding to the words in the text as well as other linguistic features not explored in this paper, could be added to the feature matrix in future for better accuracies.

Acknowledgements. Comments on the paper by the anonymous reviewers were immensely helpful in revising the paper.

References

1. Lazer, D., et al.: The science of fake news. Science **359**(6380), 1094–1096 (2018)
2. Vosoughi, S., Roy, D., Aral, S.: The spread of true and false news online. Science **359** (6380), 1146–1151 (2018)
3. Ahmed, H., Traore, I., Saad, S.: Detecting opinion spams and fake news using text classification. Secur. Priv. **1**(1) (2017). https://onlinelibrary.wiley.com/doi/full/10.1002/spy2.9
4. Shu, K., Sliva, A., Wang, S., Tang, J., Liu, H.: Fake news detection on social media: a data mining perspective. ACM SIGKDD Explor. Newsl. **19**(1), 22–36 (2017). https://www.kdd.org/exploration_files/19-1-Article2.pdf
5. Horne, B.D., Adali, S.: This just in: fake news packs a lot in title, uses simpler, repetitive content in text body, more similar to satire than real news. Paper Presented at: The 2nd International Workshop on News and Public Opinion at ICWSM; Montreal, Canada (2017). https://arxiv.org/abs/1703.09398
6. Horne, B.D., Khedr, S., Adali, S.: Sampling the news producers: a large news and feature data set for the study of the complex media landscape. In: Proceedings of the Twelfth International Conference on Web and Social Media, ICWSM 2018, Stanford, CA, USA, pp. 518–527 (2018)
7. Baly, R., Karadzhov, G., Alexandrov, D., Glass, J., Nakov, P.: Predicting Factuality of Reporting and Bias of News Media Sources (2018). https://arxiv.org/abs/1810.01765
8. Pérez-Rosas, V., Kleinberg, B., Lefevre, A., Mihalcea, R.: Automatic detection of fake news. In: Proceedings of the 27th International Conference on Computational Linguistics, Santa Fe, New Mexico, USA, 20–26 August, pp. 3391–3401 (2018)
9. Gilda, S.: Evaluating machine learning algorithms for fake news detection. In: 2017 IEEE 15th Student Conference on Research and Development (SCOReD), Putrajaya, pp. 110–115 (2017)
10. Bajaj, S.: The Pope Has a New Baby! Fake News Detection Using Deep Learning. https://web.stanford.edu/class/cs224n/reports/2710385.pdf
11. http://news.mit.edu/2018/mit-csail-machine-learning-system-detects-fake-news-from-source-1004
12. Liu, Y., Wu, Y.-F.B.: Early detection of fake news on social media through propagation path, classification with recurrent and convolutional networks. In: AAAI Publications, Thirty-Second AAAI Conference on Artificial Intelligence (2018). https://aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/16826
13. Shu, K., Wang, S., Liu, H.: Beyond news contents: the role of social context for fake news detection. In: WSDM 2019, 11–15 February (2019). http://www.public.asu.edu/∼skai2/files/wsdm_2019_fake_news.pdf
14. Tacchini, E., Ballarin, G., Della Vedova, M.L., Moret, S., de Alfaro, L.: Some like it hoax: automated fake news detection in social networks. Technical report UCSC-SOE-17-05 School of Engineering, University of California, Santa Cruz (2017). https://www.soe.ucsc.edu/sites/default/files/technical-reports/UCSC-SOE-17-05.pdf
15. Opensource Dataset. http://www.opensources.co/
16. Kaggle Dataset. https://www.kaggle.com/jruvika/fake-news-detection
17. GitHub Repository. https://github.com/GeorgeMcIntire/fake_real_news_dataset

18. Pennington, J., Socher, R., Manning, C.D.: GloVe: Global Vectors for Word Representation (2014). https://nlp.stanford.edu/pubs/glove.pdf
19. https://www.ibm.com/cloud/watson-tone-analyzer
20. https://www.nltk.org/
21. Furnkranz, J., et al.: Case study in using linguistic phrases for text categorization on the WWW. In: AAAI Technical report WS-98: (1998). https://www.aaai.org/Papers/Workshops/1998/WS-98-05/WS98-05-002.pdf
22. Seki, Y.: Sentence extraction by tf-idf and position weighting from newspaper articles. In: Proceedings of the 3rd NTCIR Workshop, Tokyo (2002). http://research.nii.ac.jp/ntcir/workshop/OnlineProceedings3/NTCIR3-TSC-SekiY.pdf
23. Chen, T., Guestrin, C.: XGBoost: a scalable tree boosting system. In: Proceedings of the 22nd ACM SIGKDD, pp. 785–794 (2016)
24. Alpaydın, E.: Introduction to Machine Learning, pp. 487–488, 2nd edn. MIT Press, Cambridge (2010)