International Journal of Computer Applications (0975 - 8887)
Volume183-No.37,November2021

Fake News Classification in Machine Learning with Different Word Representations

Elaf Alhazmi
Computer Science Department
Umm Al-Qura University
Al-leith, Saudi Arabia

ABSTRACT

Text classification has been effectively applied in a variety of domains, one of which is the detection of fake news. Working with a classification framework is an important approach for detecting fake news. One of the most significant steps in converting text to numbers in a classification framework is feature extraction. In this paper, we compare the effectiveness of several feature extraction approaches such as bag of words, TF-IDF, and one-hot encoding. For the experiment, we measured the accuracy of the classification and evaluated the best/worst classifier in three techniques using three fake news detection data sets and six machine learning classifiers. Following our tests, we discovered that employing a bag of words, also known as CountVectorizer, and the TF-IDF approach in text classification for selected data outperforms one-hot encoding. Despite the fact that logistic regression and support vector machine both produce valid results by using bag of words and TF-IDF, random forest classifier is the only algorithm that consistently produces accurate results in all three feature extraction methods. The accuracy of support vector machine in one-hot encoding was the lowest even though the algorithm produced substantial results in the other two extraction procedures

General Terms

Natural Language Processing, Machine Learning

Keywords

Fake News Classification, Word Representation, Machine Learning

1. INTRODUCTION

Natural Language Processing -NLP- is a fascinating field of study. Text classification is one of these challenging domains, with studies like as spam e-mail filtering [10], sentiment analysis [13], and, more recently, fake news detection [19]. Preprocessing, feature extraction, feature selection, and classification stages are all part of the framework of text classification [11]. To begin, there is the cleaning stage, which includes operations like as tokenization, removing stop words, lowering letters, and stemming. Vectorization model is commonly used in the feature extraction stage. The process of transforming text to numeric is used in a variety of ways, including bag-of-word, TF-IDF, one-hot encoding, and word embedding. Finally, the training and testing stages of the classification process employ tried-and-true classification methods such as support vector machines, decision trees, artificial neural networks, and the naïve-bayes classifier. Finally, successful classifiers such as naïve-bayes classifier, support vector machines, decision trees, and artificial neural networks, are used in the training and testing stages of the classification process. As has recently been explored in several machine learning [3, 21, 25] and deep neural network [5, 12] experiments, model training accuracy affects text classification performance [22]. The proper approach for feature extraction as well as the selected algorithm for classification have an effect on the accuracy of classification issues. As a result, fake news identification is one of the classification issues that is influenced by the approach and algorithm chosen throughout the text classification process.

Fake news has received considerable attention in recent years, especially following the 2016 US elections. Since 2017, multiple data sets such as Kaggle, and LIAR have been introduced, one of the NLP study areas that has received interest is fake news detection. The method of categorizing fake news is separated into two stages: the first is identifying fake news, and the second is classifying fake news in various forms. The main purpose is to detect the fake news, which is a standard text classification problem with an immediate solution. The model is critical for distinguishing between real and fake news.

This paper focuses on text classification by conducting an experiment with classifying fake news to assess and compare the accuracy results of different feature extractions: bag of words, TF-IDF, and one-hot representation on various classifiers: naïve-bayes, logistic regression, decision tree, random forest, support vector machine, and simple multi-layer perceptron classifier. One of the fundamental contribution of this research is to compare the accuracy results of each algorithm using a variety of feature extraction approaches, as well as to determine the best/worst performance of each algorithm utilizing feature extraction techniques in various data sets. The other sections of this

paper are organized as follows: section 2 examines some relevant works, section 3 describes the technique used in this paper, section 4 offers the experiment that was conducted, and section 5 concludes the paper.

2. RELATED WORKS

The classic example of a text classification challenge is the identification of fake news. Fake news is one of the text classification domains that has received an amount of publicity in 2016, especially following united States presidential election. Due to the necessity of detecting fake news, a number of research studies have been conducted to compare the effectiveness of machine and deep learning algorithms when utilizing various feature engineering methodologies. When it comes to detecting fake news, machine learning techniques for supervised classification are commonly used. In [18], the author used a support vector machine to test 360 news articles, and the model performed well with 90 percent precision, 84 percent recall, and with F1-score of 87 percent. In addition, the authors in [3] conducted an empirical investigation on the LIAR [24] data set using four algorithms that are widely used in machine learning: decision trees, naïve bayes, random forest, and neural networks, the findings demonstrate that the naïve bayes classifier performs better than the other techniques.

Empirical studies on fake news used a variety of feature extraction approaches as well as word representation techniques. The authors in [8] applied a classification model for detecting fake news, that depends on Doc2vec and Word2vec embedding as feature extraction techniques. The proposed approaches were tested using a variety of methods, including artificial neural network, long short-term memory, logistic regression, support vector machine, and random forest. The suggested Doc2vec model outperformed a TF-IDF based model that used support vector machine and logistic regression classifiers on the same data set.

In several studies, word embedding was compared to TF-IDF. To detect fake news articles, the author in [21] applied and compared different feature engineering methods such as bag of word, TF-IDF, and word embedding. With 94 percent accuracy, the linear support vector machine with TF-IDF achieves the maximum accuracy. Despite the fact that the accuracy of a neural network with bag of word remains the same, neural networks take longer to train and are more sophisticated. The efficacy of employing word embedding and TF-IDF in text categorization was compared in the two research studies [4] and [23]. The former finding concludes that TF-IDF outperformed word embedding (FastText), with the ANN achieving the greatest results, with an F1-score of 97 percent in all experiments. The latter found that word embedding (Word2Vec) performed worse than bag of words (CountVectorizer) and TF-IDF. Furthermore, among the five selected strategies: random forest, logistic regression, support vector machine, artificial neural network (ANN), and long short term memory networks (LSTM), the last two mentioned algorithms (ANN and LSTM) produced the best performance results.

Not just machine learning, but also deep learning techniques such as the recurrent neural network RNN are commonly used to identify fake news [2]. The author in [5] uses LSTM and Bi-directional LSTM to evaluate the performance of two publicly available unstructured news article datasets. The accuracy of Bi-directional LSTM model outperformed alternative methods such as convolutional neural network and unidirectional LSTM for detecting fake news.

3. METHODOLOGY

The suggested experiment is divided into five phases: data acquisition, data preprossing, feature extraction, classification, and metrics evaluation. The data sets selected from Kaggle public data sets. The preprocessing stage usually consists of a series of operations that begin with tokenizing words and end with stemming words. One of the following approaches, bag of word, TF-IDF, and one-hot encoding, will be used to vectorize text data features. Naïve bayes (NB), logistic regression (LR), decision tree (DT), random forest classifier (RFC), support vector machine (SVM), and multi-layer perceptron classifiers (MLP) were among the classification algorithms utilized in the experiment. Finally, metrics equations were used to evaluate the classifiers.

3.1 Data Set

We used three Kaggle public data sets that were divided between fake and real labels. The first data set, known as Fake News [1] data set, is made up of 20800 human-labeled brief sentences that have been reduced to 18285 labels after null values have been removed. Each label in a news article has its own id, title, author, text, and label. The label shows if the article is fake or real, with 1 denoting unreliability and 0 denoting reliability. The second data set, known as Fake News detection [2] data set, contains 4009 fake-real news articles that have been reduced to 3988 news items after null values were eliminated. The following features are included in each news article: url, headline, body, and label. The label has a 1 for real and a 0 for fake. The final data collection, known as Real or Fake [3] data set, has 6335 news article labels. The data set label has a unique id, title, text, and a label with 'REAL' or 'FAKE' text.

3.2 Data Preprocessing

Data preparation involves a number of key stages. The data corpus was tokenized first, then special characters were eliminated, and stop words were deleted before lowering and stemming word. Removing Stop Words is a method for removing all meaningless words from a corpus. In machine learning classifiers, nonsensical words such as conjunctions, and pronouns cause noise in classification performance. Stop-words, or nonsensical words, have been eliminated from selected data sets. Stemming is the operation that converts the word to the original format, such as changing the words 'likes', 'liked', 'likely', and 'liking' to the original word 'like'. We also utilized Porter stemmer because it is the most extensively used stemming algorithm in text classification.

3.3 Feature Extraction

The process of converting text data into numerical values is known as feature extraction. We used three different ways to convert text into numerical representation: bag of words, TF-IDF, and one-hot representation.

3.3.1 Bag of Word. is a text representation that depicts the frequency of words in a document. It entails two components: a known-word vocabulary and a measure of the presence of known

[1] https://www.kaggle.com/c/fake-news/data
[2] https://www.kaggle.com/jruvika/fake-news-detection
[3] https://www.kaggle.com/rchitic17/real-or-fake

words. With unigram, bigram, and trigram, the $CountVectorizer$ from the scikit-learn module was utilized [6].

3.3.2 TF-IDF. is a numerical statistic called term frequency–inverse document frequency that measures how a word is significant in a corpus. As indicated in Equation (1), TF-IDF is calculated as the multiplication of word frequency -tf- and inverse of document frequency -idf, respectively [14].

$$TF - IDF = TF(\mathbf{t}, \mathbf{d}) \times IDF(\mathbf{t}) \tag{1}$$

As shown in Equation (2), word frequency is a metric that reflects how frequently a term, t, appears in a document, d, and inverse document frequency is a metric for the importance of a phrase, as shown in Equation (3).

$$TF(t, d) = \frac{Number\ of\ \mathbf{t}\ in\ \mathbf{d}}{Total\ number\ of\ \mathbf{t}\ in\ \mathbf{d}} \tag{2}$$

$$IDF(t) = log\frac{N}{DF(\mathbf{t})} \tag{3}$$

$$DF(t) = occurence\ of\ \mathbf{t}\ in\ documents \tag{4}$$

Inverse document frequency - IDF - is determined by document frequency - DF - which is calculated by counting the times of term t occurred in a documents as indicated in Equation (4). DF calculates the incidence of a word existing in total documents without taking into account the number of times a word appears in a document.

3.3.3 One-hot Encoding. is a technique for converting textual data into numerical vectors that either containing zero or one [7]. Because of the high-dimension and big sparse data set, one-hot encoding has some performance issues. When the vocabulary size of the words in the data set is enormous, each word will have one-hot vector, resulting in a very high dimension data set that is sparse, with many zeros and only a few indexes with value one. To build a one-hot representation in this experiment, we employed two primary methods: one_hot, which converts a text into a collection of dictionary n word indexes, and $pad\_sequences$, which adds zeros to all sentences to make them the same size. It should also be mentioned that word embedding was not used in this study.

3.4 Classification Algorithms

In the experiment, classifiers such as naïve bayes, logistic regression, decision tree, random forest, support vector machine, and multi-layered perceptron neural network were applied to distinguish between fake and real news.

3.4.1 Naïve Bayes. as demonstrated in Equation (5), where A and B denote two conditions. The foundation of naïve bayes is the Bayes Theorem, which relies on probability. The classifier -Naïve Bayes- built on the notion that all classified attributes are independent; in other words, the existence of one feature in a class does not imply the presence of another [14]. In the conducted experiment, the selected algorithm for naïve bayes model in Sci-Kit learn is $BernoulliNB$ [20].

$$P(A|B) = \frac{P(B|A) * P(A))}{P(B))} \tag{5}$$

3.4.2 Logistic Regression. is among the most frequently applied algorithms in classification [15]. By adding a sigmoid function at the end, logistic regression yields a logistic graph that is restricted between 0 and 1 values, as illustrated in Equation (6), where x, w and b denotes input features, weight, bias value, respectively.

$$P = \frac{1}{1 + e^{-z}} \quad with\ \ z = b + w_1x_1 + ... + w_nx_n \tag{6}$$

3.4.3 Decision Tree. not only have numerous nodes, but it also has leaf nodes. The decision nodes assess feature values, whereas the leaf nodes assign labels. The initial decision node is the root node of a decision tree. This node has a condition that looks at one of the input value's properties and selects a branch based on its value. The procedure continues down the branch determined by each node's condition until we reach a leaf node that assigns a label to the input value [17].

3.4.4 Random Forest. simply combines multiple decision trees to produce more accurate and consistent forecasts. Each node represents a categorization class in a decision tree, which is made up of parents with distinct conditions branches. The random forest classifier is a method for creating a large number of decision trees that is used in ensemble classification [1].

3.4.5 Support Vector Machine. creates a hyperplane that divides the training data as evenly as feasible in N-dimensional space, where N is the number of distinguishing qualities between data points. A hyperplane is a line that divides a plane into two portions in two dimensions, with each class on one side [16]. In textual data categorization, support vector machine performs well with a high set of features. It requires not just a large amount of memory but also a huge amount of adjusting [9].

3.4.6 Artificial Neural Network. is a sort of computing system that mimics the way the human brain processes and analyzes data. It is made up of numerous layers of nodes called neurons. A layer for input ($x1, x2, ..., xn$), an optional hidden layer, and a layer for output (y) are all included in a simple ANN . The weights (also known as parameters) of a layer store the definition of what it does to the input data. The activation function then takes these inputs and decides what to do with them. In the experiment, a multi-layered perceptron (MLP)-feedforward artificial neural network was used to assess fake news detection capability [8].

3.5 Evaluation Metrics

Accuracy, precision, recall, and F1 measurements are the four primary assessment metrics used to assess the results of the conducted experiments. The ratio of the quantity of predictions that are correct - true positives and true negatives - to the total quantity of input labels is known as accuracy.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Precision is defined as the ratio of true positive outcomes to positive expected results indicated by the model, and Recall is the ratio of right positive outcomes to the total set - the sum of true positive and false negative sets - as shown in the following Equation:

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

Table 1. : Data Set - Statistics

Number of Labels	Data set (1)	Data set (2)	Data Set (3)
Data Size	18285	3988	6335
Training Size	14628	3190	5068
Test Size	3657	798	1267

Data_Set: (1),(2), and (3) are Fake News, Fake News detection, and Real or Fake data sets, respectively

The F1 score is also known as F-measure. To show balancing with both recall and precision, F1 score may be preferable to be selected, and it is calculated as follow:

$$F1 = 2 * \frac{Recall * Precision}{Recall + Precision}$$

The abbreviations TP, TN, FP, and FN stand for true positives, true negatives, false positives, and false negatives, respectively.

4. EXPERIMENT

In this section, we show how different classification algorithms performed with different feature extraction methods for text data. The experiment will be evaluated using classification methods for each sort of feature selection in the experiment. The purpose of the experiment is to examine the performance of algorithms for classification with various methods of word representation

4.1 Experimental Setting

We used Python 3.8.3 as the language to test the performance of algorithms for different word representations. The proposed models were implemented using Sci-kit Learn, TensorFlow, and Keras libraries. For measurements, the following environment and platform operating system were used: Version of Windows 10 with 2.50 GHz Intel Core i5 processor and 8 GB RAM.

4.2 Experimental Implementation

Three public data sets were used in the experiment [4]. Data training actually took 80 percent of the data set, while testing took up the remaining 20 percent. Table (1) shows the statistics for the sets of data. In classification, we employed bag of words, TF-IDF, and one-hot encoding. Nine files, each with six models, are being implemented to see how well selected machine learning techniques recognize fake news.

In the first approach, we used the $CountVectorizer$ module which is based on a bag of words for feature extraction, and $TfidfVectorizer$ for the TF-IDF technique. Both algorithms are limited to five-thousand maximum features and used for word grams with unigram, bigram, and trigram. One-hot encoding is the third strategy, and it is one of the most common approaches for feature extraction. In keras and tensorflow, one-hot encoding is implemented using the one_hot method and $pad\_sequences$, respectively. After transforming the sparse matrix to a dense matrix, one-hot encoding is often utilized in deep learning algorithms as word embedding but it must be mentioned that word embedding were not implemented in this experiment.

[4] https://github.com/E-Alhazmi/-FakeNewsClassification_Project.git

Table 2. : Parameters of Algorithms

Task	**Bag of Words**	**TF-IDF**	**One-hot**
Naive Bayes	alpha = 0.01		alpha = 0.01
Logistic Regression	C = 2 solver = newton-cg		C = 2 solver = newton-cg
Decision Tree	max_depth = None max_leaf_nodes = None min_samples_split = 2 min_samples_leaf = 1		max_depth = None min_samples_split = 2 min_samples_leaf = 1
Random Forest	n_estimators = 100 max_depth = None min_samples_split = 2 min_samples_leaf = 1 max_leaf_nodes = None		max_depth = None min_samples_split = 2 min_samples_leaf = 1 n_estimators = 100
Support Vector Machine	Kernel = linear		Kernel = sigmoid
Multi-layered Perceptron Neural Network	Activation = logistic solver = lbfgs alpha = 1 hidden_layer_sizes = 100		Activation = logistic solver = lbfgs alpha = 1 hidden_layer_sizes = 100

4.3 Experimental Training

The quantity of news statements in each data set varies, thus we adjust the parameters of each algorithm on each data set to get different results. After some poor results in training , grid search tool was applied to find the best settings for each algorithm, as shown in Table (2). We used the trained models of each algorithm for testing after fine-tuning the parameters of each algorithm.

4.4 Experimental Result

The accuracy, precision, recall, and F1-score outcomes of each algorithm are shown in Table (3) using three different word representations (bag of words, TF-IDF, and one-hot encoding) and three different data sets. Figure (1) shows the summary accuracy results of the conducted experiment. The confusion matrix with one of the highest performance in each data set with varied word representations is shown in Figure (2). In several algorithms, the experiment findings indicate that bag of words and TF-IDF handle fake news classification better than one-hot encoding. Despite the fact that the best algorithm performance differed in three feature extractions, the Random Forest Classifier -RFC- achieves the accurate results in all three word representations.

4.5 Experimental Evaluation

Standard metrics that quantify overall performance were used to evaluate each algorithm's classification process. In various methods, the overall performance of bag of words and TF-IDF outperformed one-hot encoding. The best performance in each methodology, as well as the greatest/worst performance in all word representation methods, will be discussed in this section. In terms of bag of words, the performance of logistic regression algorithm exhibits the best performance for all of the examined data sets. In data sets (1), (2), and (3), Logistic regression -LR- was the most accurate, with accuracy of 94 percent, 86 percent, and 84 percent, respectively. In three data sets, the support vector machine -SVM- also displays good accuracy results. SVM is able to predict fake real news with 93 percent, 84 percent, and 82

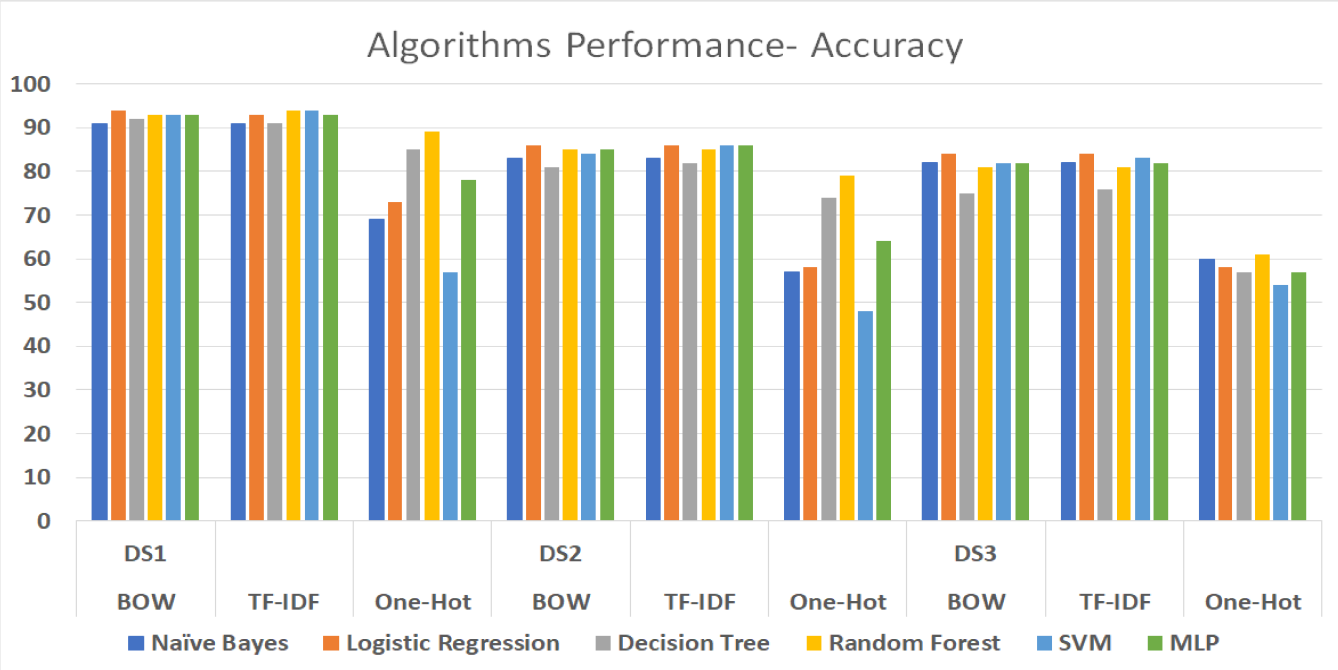


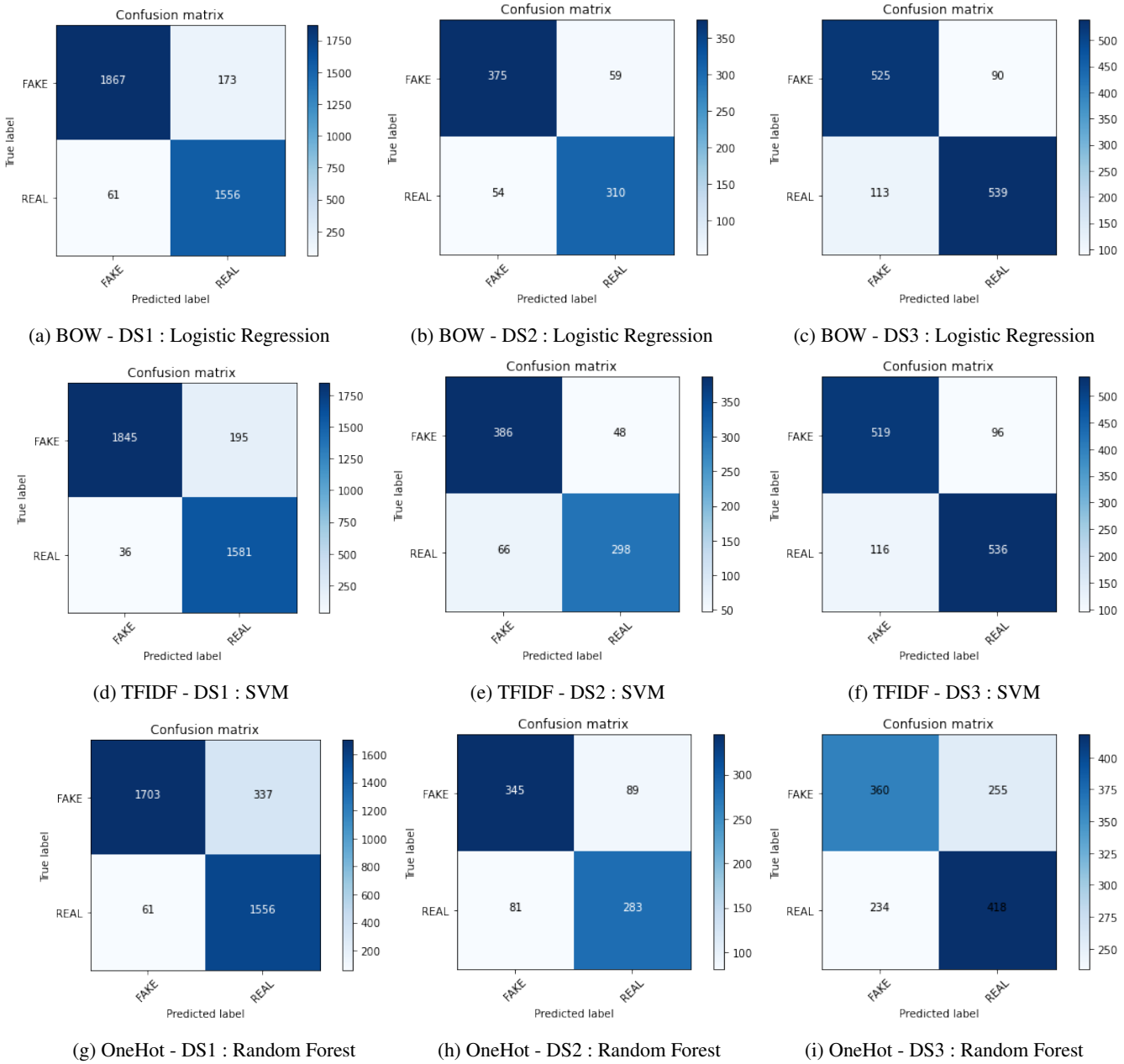
Fig. 1: Accuracy- Different Word Representations

Table 3. : Algorithms Performance with Feature Extraction Methods

	Data (1) - Fake News				Data (2) - Fake News detection				Data (3) - Real or Fake			
Bag of Words												
	Accuracy	Precision	Recall	F1	Accuracy	Precision	Recall	F1	Accuracy	Precision	Recall	F1
NB	0.91	0.91	0.91	0.91	0.83	0.83	0.83	0.83	0.82	0.82	0.82	0.82
LR	0.94	0.93	0.94	0.94	0.86	0.86	0.86	0.86	0.84	0.84	0.84	0.84
DT	0.92	0.92	0.92	0.92	0.81	0.82	0.80	0.80	0.75	0.75	0.75	0.75
RF	0.93	0.93	0.94	0.93	0.85	0.86	0.85	0.85	0.81	0.81	0.81	0.81
SVM	0.93	0.93	0.93	0.93	0.84	0.83	0.83	0.83	0.82	0.82	0.82	0.82
MLP	0.93	0.93	0.93	0.93	0.85	0.85	0.84	0.84	0.82	0.82	0.82	0.82
TF-IDF												
NB	0.91	0.91	0.91	0.91.	0.83	0.84	0.83	0.83	0.82	0.82	0.82	0.82
LR	0.93	0.93	0.94	0.93	0.86	0.86	0.85	0.85	0.84	0.84	0.84	0.84
DT	0.91	0.91	0.91	0.91	0.82	0.82	0.81	0.81	0.76	0.76	0.76	0.76
RF	0.94	0.94	0.94	0.94	0.85	0.85	0.84	0.84	0.81	0.81	0.81	0.81
SVM	0.94	0.94	0.94	0.94	0.86	0.86	0.85	0.86	0.83	0.83	0.83	0.83
MLP	0.93	0.93	0.93	0.93	0.86	0.86	0.85	0.85	0.82	0.82	0.82	0.82
One-Hot Encoding												
NB	0.69	0.69	0.67	0.67	0.57	0.61	0.59	0.55	0.60	0.61	0.59	0.58
LR	0.73	0.72	0.72	0.72	0.58	0.58	0.58	0.58	0.58	0.59	0.58	0.56
DT	0.85	0.85	0.85	0.85	0.74	0.74	0.73	0.73	0.57	0.57	0.57	0.57
RF	0.89	0.89	0.90	0.89	0.79	0.79	0.79	0.79	0.61	0.61	0.61	0.61
SVM	0.57	0.56	0.56	0.56	0.48	0.48	0.48	0.48	0.51	0.51	0.51	0.51
MLP	0.78	0.78	0.78	0.78	0.63	0.63	0.63	0.63	0.58	0.58	0.57	0.57

Naïve Bayes: NB , Logistic Regression: LR, Decision Tree: DT, Random Forest: RF, Support Vector Machine: SVM, Multi-layer Perceptron neural network: MLP

percent accuracy in data sets (1), (2), and (3), respectively. Random forest classifier -RFC- is also regarded as important classification algorithm. With 93 percent and 85 percent in data set (1) and data set (2), respectively. RFC is consider as second best method in the experiment. Although the bag of words algorithm revealed variations between SVM, RFC, and MLP, another technique that shows promise in fake news classification. Multi-layer perceptron neural classifier -MLP- has accuracy results of 93 percent, 85



(a) BOW - DS1 : Logistic Regression (b) BOW - DS2 : Logistic Regression (c) BOW - DS3 : Logistic Regression

(d) TFIDF - DS1 : SVM (e) TFIDF - DS2 : SVM (f) TFIDF - DS3 : SVM

(g) OneHot - DS1 : Random Forest (h) OneHot - DS2 : Random Forest (i) OneHot - DS3 : Random Forest

Fig. 2: Highest Accuracy Performance

percent, and 82 percent from the first to the third data sets. In terms of TF-IDF, two methods, SVM and LR, yield considerable classification results. SVM accuracy results in data sets (1), (2), and (3) are 94 percent, 86 percent, and 83 percent, respectively, while Logistic regression accuracy results are 93 percent, 86 percent, and 84 percent, respectively. It should also be highlighted that RFC is still considered an algorithm with considerable accuracy results of 94 percent, 85 percent, and 81 percent in data sets (1), (2), and (3), respectively.

On the other hand, One-hot encoding showed remarkable overall weak performance compared to bag of words and TF-IDF. One-hot encoding is one of the basic approach used to convert text to numbers. In the conducted experiment, one-hot encoding is consider as naive approach and the best accuracy results showed by training the data sets is by using random forest classifier. RFC accuracy results for data sets (1), (2), and (3) are 89 percent, 79 percent, and 61 percent, respectively. When one-hot encoding is employed for word representation, SVM has the lowest performance in all three data sets.

5. CONCLUSION

In this paper, multiple word representations were used to compare the accuracy of different methods in machine learning and artificial neural networks on fake news data sets, including bag of words, TF-IDF, and one-hot encoding. To evaluate the best/worst algorithm in fake news classification, three data sets were selected: Fake News, Fake News detection, and Real-or-Fake data sets. The algorithms being used are naïve bayes, logistic regression, decision tree, random forest, support vector machine, and multi-layer perceptron neural classifiers. It was found that bag of words and TF-IDF handled text classification better than one-hot encoding, and that random forest classifiers accuracy results were noticeably accurate in all three words representation methods, while Logistic regression outperformed accuracy results in bag of words and TF-IDF word representation. There are also considerable significant accuracy result by using support vector machine, and multi-layer perceptron neural algorithm in bag of words and TF-IDF but the worst result of support vector machine noticed in one-hot encoding.

6. REFERENCES

[1] Rahul Agrawal, Archit Gupta, Yashoteja Prabhu, and Manik Varma. Multi-label learning with millions of labels: Recommending advertiser bid phrases for web pages. In *Proceedings of the 22nd international conference on World Wide Web*, pages 13–24, 2013.

[2] Oluwaseun Ajao, Deepayan Bhowmik, and Shahrzad Zargari. Fake news identification on twitter with hybrid cnn and rnn models. In *Proceedings of the 9th international conference on social media and society*, pages 226–230, 2018.

[3] Abdulaziz Albahr and Marwan Albahar. An empirical comparisonof fake news detection using different machine learning algorithms. *IJACSA*, 2020.

[4] Eman Alsagour, Lubna Alhenki, and Mohammed Al-Dhelaan. Different word representation for text classification: A comparative study. In *2019 IEEE/ACS 16th International Conference on Computer Systems and Applications (AICCSA)*, pages 1–2. IEEE, 2019.

[5] Pritika Bahad, Preeti Saxena, and Raj Kamal. Fake news detection using bi-directional lstm-recurrent neural network. *Procedia Computer Science*, 165:74–82, 2019.

[6] Jason Brownlee. *Deep learning for natural language processing: develop deep learning models for your natural language problems*. Machine Learning Mastery, 2017.

[7] Mwamba Kasongo Dahouda and Inwhee Joe. A deep-learned embedding technique for categorical features encoding. *IEEE Access*, 9:114381–114391, 2021.

[8] Ibrahim Eldesoky, Desouky Fattoh, Ali Farid, and Mousa. Fake news detection based on word and document embedding using machine learning classifiers. *Journal of Theoretical and Applied Information Technology*, 99, 04 2021.

[9] Babacar Gaye, Dezheng Zhang, and Aziguli Wulamu. Improvement of support vector machine algorithm in big data background. *Mathematical Problems in Engineering*, 2021, 2021.

[10] Serkan Günal, Semih Ergin, M Bilginer Gülmezoğlu, and Ö Nezih Gerek. On feature extraction for spam e-mail detection. In *International Workshop on Multimedia Content Representation, Classification and Security*, pages 635–642. Springer, 2006.

[11] Ammar Ismael Kadhim. Survey on supervised machine learning techniques for automatic text classification. *Artificial Intelligence Review*, 52(1):273–292, 2019.

[12] Sheng How Kong, Li Mei Tan, Keng Hoon Gan, and Nur Hana Samsudin. Fake news detection using deep learning. In *2020 IEEE 10th Symposium on Computer Applications & Industrial Electronics (ISCAIE)*, pages 102–107. IEEE, 2020.

[13] Isa Maks and Piek Vossen. A lexicon model for deep sentiment analysis and opinion mining applications. *Decision Support Systems*, 53(4):680–688, 2012.

[14] CD Manning, P Raghavan, and H Schutze. Introduction to information retrieval (vol. 1). cambridge: Cambridge university press. 2008.

[15] Chao-Ying Joanne Peng, Kuk Lida Lee, and Gary M Ingersoll. An introduction to logistic regression analysis and reporting. *The journal of educational research*, 96(1):3–14, 2002.

[16] Karishnu Poddar, KS Umadevi, et al. Comparison of various machine learning models for accurate detection of fake news. In *2019 Innovations in Power and Advanced Computing Technologies (i-PACT)*, volume 1, pages 1–5. IEEE, 2019.

[17] Lior Rokach and Oded Maimon. Top-down induction of decision trees classifiers-a survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 35(4):476–487, 2005.

[18] Victoria L Rubin, Niall Conroy, Yimin Chen, and Sarah Cornwell. Fake news or truth? using satirical cues to detect potentially misleading news. In *Proceedings of the second workshop on computational approaches to deception detection*, pages 7–17, 2016.

[19] Kai Shu, Amy Sliva, Suhang Wang, Jiliang Tang, and Huan Liu. Fake news detection on social media: A data mining perspective. *ACM SIGKDD explorations newsletter*, 19(1):22–36, 2017.

[20] Mandeep Singh, Mohammed Wasim Bhatt, Harpreet Singh Bedi, and Umang Mishra. Performance of bernoulli's naive bayes classifier in the detection of fake news. *Materials Today: Proceedings*, 2020.

[21] N Smitha and R Bharath. Performance comparison of machine learning classifiers for fake news detection. In *2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA)*, pages 696–700. IEEE, 2020.

[22] Alper Kursat Uysal and Serkan Gunal. The impact of preprocessing on text classification. *Information processing & management*, 50(1):104–112, 2014.

[23] Sairamvinay Vijayaraghavan, Ye Wang, Zhiyuan Guo, John Voong, Wenda Xu, Armand Nasseri, Jiaru Cai, Linda Li, Kevin Vuong, and Eshan Wadhwa. Fake news detection with different models. *arXiv preprint arXiv:2003.04978*, 2020.

[24] William Yang Wang. " liar, liar pants on fire": A new benchmark dataset for fake news detection. *arXiv preprint arXiv:1705.00648*, 2017.

[25] Kasra Majbouri Yazdi, Adel Majbouri Yazdi, Saeid Khodayi, Jingyu Hou, Wanlei Zhou, and Saeed Saedy. Improving fake news detection using k-means and support vector machine approaches. *International Journal of Electronics and Communication Engineering*, 14(2):38–42, 2020.