Fake News Classification Using Random Forest and Decision Tree (J48)

Article · December 2020

CITATIONS
0

READS
1,253

2 authors:

Reham Jehad Al-Shammari
Al-Nahrain University
2 PUBLICATIONS **0** CITATIONS
SEE PROFILE

Suhad A. Yousif
Al-Nahrain University
14 PUBLICATIONS **64** CITATIONS
SEE PROFILE

Al-Nahrain Journal of Science
ANJS, Vol.23 (4), December, 2020, pp. 49–55

Fake News Classification Using Random Forest and Decision Tree (J48)

Reham Jehad[1] and Suhad A. Yousif[2,*]

[1,2]Department of Computer Science, College of Science, Al-Nahrain University, Baghdad, Iraq
[1]rehamflower95@gmail.com

Articles Information

Received:
28.08.2020
Accepted:
23.11.2020
Published:
01.12.2020

Keywords:
Machine learning
Text classification
Natural language processing
Decision Tree
Random Forest

Abstract

Fake News is one of the most popular phenomena that have considerable effects on our social life, especially in the political domain. Nowadays, creating fake news becomes very easy because of users' widespread using the internet and social media. Therefore, the detection of elusiveness news is a crucial problem that needs to be considerable mainly because of its challenges like the limited amount of the benchmark datasets and the amount of the published news every second. This research proposed utilizing two different machine learning algorithms (random forest and decision tree (J48)) to detect the fake news. In this paper, the full dataset size equals 20,761 samples, while the testing sample size equals 4,345 samples. The preprocessing steps start with cleaning data by removing unnecessary special characters, numbers, English letters, and white spaces, and finally, removing stop words is implemented. After that, the most popular feature extraction method (TF-IDF) is used before applying the two suggested classification algorithms. The results show that the best accuracy achieved equals 89.11% using the decision tree model while using the random forest; the accuracy achieved equals 84.97 %.

DOI: 10.22401/ANJS.23.4.09

*Corresponding author: say@sc.nahrainuniv.edu.iq

1. Introduction

In 2016, the notoriety of misinformation in the American political thesis received primary attention, especially after electing Donald Trump. The term "Fake News" has become a common language in this field, especially for describing the misleading and false articles printed initially to make money from the web page's visits.

Nowadays, most researchers focus on creating a model that is capable of accurate prediction to distinguish whether a particular article is classified as real or false news. It is necessary to determine what makes the new site "legitimate" and define it objectively [1]. The harmful effects of inaccurate information will make people believe that Hillary Clinton has a foreign child, trying to convince readers that President Trump is trying to cancel the first amendment to kill India's crowds because false rumors spread in WhatsApp application.

Today's, we believe in what we see on the websites or social media and do not pursue to check if the provided information is true or false [2]. It is difficult to distinguish between the fake and real news manually because people need to spend a long time checking the references of news and making sure of their truthfulness. Therefore an automatic and intelligent model for the detection of fake news becomes substantial demand [3]. Thus, the detection of fake news takes considerable attention from the researcher's community worldwide. In Singapore, Google and Facebook object to introducing new laws to combat fake news, claiming that existing legislation is sufficient to address the problem and that an effective way of fighting fake news is by coaching people on how to differentiate from fake news vs. real news. Despite all these efforts done by the existing society, people, technology, and processes, fake news still occurs in some shape or form every day [4].

Technologies like Artificial Intelligence (AI) and Natural Language Processing techniques (NLP), and machine learning promise great human beings for researchers to build systems capable of automatically detecting fake news. On the other hand, discovering fake news is a complicated process because it needs models to summarize news and compare them with real news to classify them as fake [5].

For the remainder of the paper, Section 2 focuses on the contribution of the work. In contrast, the related work is exemplified in Section 3. Section 4 describes the workflow design. Section 5 shows the experimental results, while section 6 offers conclusions and directions for future work.

2. Contribution

The contribution of this research is trying to improve the accuracy results of the fake news classification in using TF-IDF feature extraction to extract the vital word from fake news articles using two different classifiers (Random Forest and Decision Tree) and then compare between their accuracy results and the related works accuracy results.

3. Related Work

Most previous research was dedicated to using machine learning and deep learning algorithms to distinguish between fake and real ones. In 2017 Shlok Gilda [6] explored the applications related to NLP techniques to detect 'fake news', which is the deceptive news stories obtained from non-reputable sources-utilizing a dataset acquired from Signal Media as well as a list of sources from the Open Sources. Co, they used TF-IDF regarding the bigrams and the detection of the probabilistic context-free grammar (PCFG) to the corpus of approximately 11000 articles. Besides, test the dataset on various algorithms of classification SVMs, Random Forests, Gradient Boosting, Stochastic Gradient Descent, and Bounded Decision Trees. The study identified that TF-IDF related to bigrams fed to a model of Stochastic Gradient Descent and identifying non-credible sources with an accuracy which is equal to 77.2%.

In 2018 Chandra Mouli Madhav Kotteti [7], they have effectively managed missing values utilizing data imputation for numerical and categorical features. Concerning the categorical features, they study imputed the missing values with frequent central value in columns, while concerning numerical features, a column's mean value is utilized for imputing the missing numerical values. Also, the vectorization of TF-IDF is used in the feature extraction to filter out the irrelevant features. The experimental results show that the MLP classifier with the suggested data pre-processing method is outperforming baselines and improving prediction accuracy by over 15% than the SGD classifier's accuracy 43.23%.

In 2018 Arjun Roy, Kingshuk Basak [8], Asif Ekbal developed different deep learning models to detect fake news and classify them into fine-grained and pre-defined categories. Initially, they developed models based on CNN as well as Bi-LSTM networks. Also, the representations acquired from the two models were fed into MLP for final classification. Furthermore, their experimentations on the benchmark dataset show many results with 44.87% as overall accuracy, outperforming modern models.

In 2019 Arvinder Pal Singh Bali, Maxson Fernandes [9] showed ML and NLP perspectives. The estimated was conducted for three standard datasets with a new group of features that have been extracted from contents and headlines. Besides, the performances regarding seven algorithms of ML concerning F1 scores and accuracies were compared. Furthermore, Gradient Boosting is outperforming classifiers with an accuracy of 88%.

In 2019 Ahlem Drif, Zineb Ferhat Hamida [10] suggested a model of (CNN) as well as Long Short Term Memory (LSTM) recurrent NN architecture, benefiting from the coarse-grained local features which are taken from CNN as well as the long-distance dependencies which are learned through LSTM where the dataset used was the articles news of fake news when the size of dataset was (20,761). Compared with the CNN and SVM baseline, the results show that the best accuracy is 0.725 in CNN-LSTM.

4. Workflow Model

This work has been completed through five steps. The general discussion of these steps is illustrated here. The first step is choosing the appropriate fake news dataset from kaggel.com and preprocessing the dataset. After that, TF-IDF for extracting word features after splitting the dataset using cross-validation (10-Fold) is applied. The next step is to classify the dataset using (Decision Tree, Random Forest) classifiers and evaluate model performance using different metrics like (accuracy, recall, and precision) as described in Figure 1.

Figure 1. Work design step of fake news detection.

4.1 Dataset

Dataset for fake news can be gathered from more than one source like news agency webpages, different social media websites as Twitter, Facebook, Instagram, and others. Nevertheless, it is not easy to distinguish the variety of news manually. Therefore, an annotator who expertizes analyzing the claims, evidence, and context from trustworthy sources is required. In general, the news data can be collected in different ways, through expert journalists, fact-checking websites, and crowd source workers. Till now, there is no concurrent upon benchmark datasets for fake news discovering problems.

In this paper, the dataset (fake news articles .CSV file) collected from kaggel.com is used. This dataset has about 20,800 records from various articles found on the internet, and their attributes are (text, author, title, and label). After applying the preprocessing step, the size of the dataset became 20,761 records. This data divided into two classes 10,423 of real news and 10,432 of fake news. Only two features (text, label) are used to detect fake news classifiers in this work. Label zero is assigned to represent unreliable news (or fake), while one is assigned to real news, as shown in Figure 2.

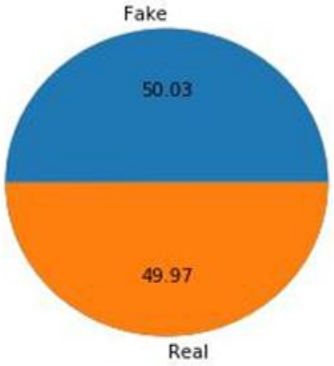

Figure 2. Count Label.

4.2 Preprocessing step

Before representing the data and the extraction of the features using (TF-IDF), the data is needed to be exposed to specific filtering and cleaning processes, such as removing stop words, punctuation, removing case-sensitivity characters, and removing the special character, numbers, and white space [11]. By eliminating the immaterial information found in the data, this will reduce the dataset size, and in turn, just the valuable information remains in the dataset [12, 13]. Table 1 shows a sample of the dataset used, representing the collected raw data without any preprocessing step, while Table 2 shows the data after the preprocessing step.

Table 1. Before preprocessing step.

	Text	Label	Length
0	House Dem Aide: We Did not Even See Comey's Let...	REAL	4930
1	Ever get the feeling your life circles the rou...	FAKE	4160
2	Why the Truth Might Get You Fired October 29, ...	REAL	7692
3	Videos 15 Civilians Killed In Single US Airstr...	REAL	3237
4	Print \nAn Iranian woman has been sentenced to...	REAL	938

Table 2. After preprocessing step.

	Text	Label	Length
0	house dem aide' even see comey' letter jason...	REAL	3338
1	ever get feeling life circles roundabout rathe...	FAKE	2857
2	truth might get fired October 29 2016 tension	REAL	5328
3	videos 15 civilians killed single us airstrike...	REAL	2268
4	print iranian woman sentenced six years prison...	REAL	688

4.3 Features extraction

Term Frequency-Inverse Document Frequency (TF-IDF)

TF-IDF can be described as a well-defined standard method used to manipulate natural language processing and creating a vector space model for extracted features [14]. In the text, the meaning of the term is evaluated. The importance of the word is evaluated in the document. Significance is increased in proportion to the number of times the word has appeared in that document. In comparison with the inverse of the word itself in the entire set of documents. Essentially, the TF-IDF measurement is related to the term *t* that takes:

- A lower value in the case where a term *t* appears fewer times in the document or appears in several documents;
- A higher value in the case where a term *t* occurs multiple times in a small number of documents;
- A lower value in the case where a term *t* occurs in nearly wholly documents. More formally:

Let D = $\{d_1, d_2, \ldots, d_n\}$ be an entire group of documents, and *t* represents a term in that group. The term frequency-inverse document size is calculated as follows:

$$TF\text{-}IDF(t, d, D) = TF(t, d) \times IDF(t, D) \quad (1)$$

Mainly, TF(*t*, *d*) characterizes the term *t* frequency in document *d* (in other words, the number of times a term appears in the document), which is represented as:

$$TF\ (t, d) = F\ (t, d)\ |d| \quad (2)$$

F(*t*,*d*) represents how often a term *t* has occurred in document *d*, and the denominator is the length of *d*, which is represented as its own terms' cardinality. The inverse document frequency IDF (*t*, *D*) can be defined below:

$$IDF(t, D) = log\ |D|\ |\ \{d|t \in d\}| \quad (3)$$

The denominator is responsible for characterizing the number of documents in which a term *t* has occurred [14].

4.4. Classifier

a) Decision Tree (J48) classifier:

One of the most common algorithms used in classification is the J48 algorithm. It is based on the C4.5 algorithm in which all the data to be studies must of the type numeric and categorical kind. Therefore, a continuous type of data will not be examined [15,16,17]. J48 utilizes two different pruning ways. The first method, named subtree replacement, which denotes the possibility of replacement nodes in a decision tree with its leaves to minimize the number of tests in the convinced path. Usually, the subtree raising is of a modest impact on the models of the decision tree. Typically, there is no exact way to predict an option's utility, although it can be advisable to turn it off when the induction procedure takes longer because of the subtree's raising being relatively computationally complicated.

Algorithm 1: Decision Tree.

Input: Predefined classes
Output: Built decision tree
Num of features =17000
Max –depth =2
Begin
Step1: Create a root node for the tree
Step 2: If all examples are positive, return leaf node 'positive.'
Else if all examples are negative, return leaf node 'negative.'
Step 3: Calculate the entropy of current state H(S)
Step 4: For each attribute, calculate the entropy concerning the attribute 'x' denoted by H(S, x)
Step 5: Select the attribute which has a maximum value of IG(S, x)
Step 6: Remove the attribute that offers the highest IG from the set of attributes
Step 7: Repeat until we run out of all attributes or the decision tree has all leaf nodes.
End

b) Random Forest

"Bagging or bootstrap aggregation can be defined as a procedure that reduces the variance of an estimated function of prediction". Bagging works efficiently with high variance and low bias techniques like trees in classification. Random forests are a significant innovation of the bagging in which it forms a large group of de-correlated trees, and after that, take an average for them. Random Forest enhanced on bagging through decreasing correlation between trees with no increase in the variance. In many situations, the random forest performance is like boosting in which they are simpler to be trained and tuned. As a result, random forests are widespread algorithms that are applied to various packages [18,19].

Algorithm 2: Random Forest Algorithm.

Input: Predefined classes
Output: Built Forest trees
Num of features =17000
Num of estimators (num of tree in the forest) = 100
Begin
Step 1: extract features from texts (X_1, X_2, …, X_n: float number)
Step 2: Compute the best splinter point between the n features For the node d.
Step 3: Utilize the optimal splinter point to split the node into two child nodes.
Step 4: Repeat steps 1, 2 to n number of nodes was reached.
Step 5: Build the forest through the repetition of steps 2-4 for D time
End

4.5. Evaluation metrics

A variety of evaluation measures were utilized to evaluate the algorithm's classification accuracy in detecting fake news. In this section, the most frequently utilized measure metric (Confusion Matrix) to detect fake news has been used. Through the formulation of this as a task of classification, it is possible to define the measures that the confusion matrix has as below [18]:

$$Precision = \frac{TP}{(TP+FP)} \quad (4)$$

$$Recall = \frac{TP}{(TP+FN)} \quad (5)$$

$$Accuracy = \frac{TP+TN}{(TP+TN+FP+FN)} \quad (6)$$

where TP represents (True Positive) and TN represent (True Negative)

Moreover, FP is False positive, and FN is False-negative, as discussed in Table 3.

Table 3. Parameters of evaluation metrics.

Parameters	Description
True Positive TP	Number of records which correctly classified
True Negative TN	Number of the correct rejection of records which have been classified
False Positive FP	The number of records incorrectly classified
False Negative FN	Number of the incorrect rejection of records which have been classified

Those measurements are usually utilized in the series of machine learning algorithms and enable evaluating the efficiency of a classifier from diverse estimation. Especially the accuracy metric that represents the likeness between predicted fake news and real fake news. Precision performs the measuring of the portion of the found fake news, which has been labeled as fake, addressing the significant issue of the fake news classification. However, due to the dataset of fake news, it is usually skewed; high precision may be accomplished by creating a smaller number of optimistic predictions, so recall is utilized to measure sensitivity or the portion of the annotated fake articles projected as fake. It should be noted that higher values mean better performances for the Recall, Precision, and Accuracy [20].

5. Results and Discussion

The classification results showed that the accuracy of the decision tree and random forest classifier is 89.11% and 84.97%, respectively. Figure 3, Figure 4, Figure 5 and Figure 6 represent the resulted confusion matrix with TP, TN, FP, and FN values. Our experimental results without preprocessing steps are 78.13% for the decision tree and 73.07 % for the random forest. Table 3 and Table 4 illustrate all results of used evaluation metrics applied to classify the fake news accurately.

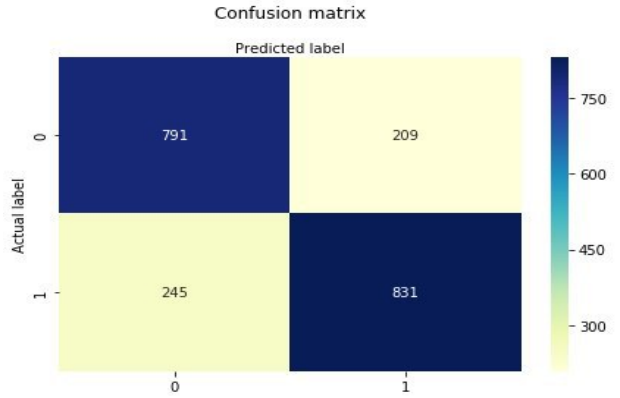

Figure 3. Confusion matrix of Decision Tree before preprocessing steps.

Table 4. Results of decision tree before preprocessing steps.

Pointer	Result
Correctly classified as 1	831
Incorrectly classified as 1	209
Correctly classified as 0	791
Incorrectly classified as 0	245
Precision	79.1%
Recall	76.35%
Accuracy	78.13%

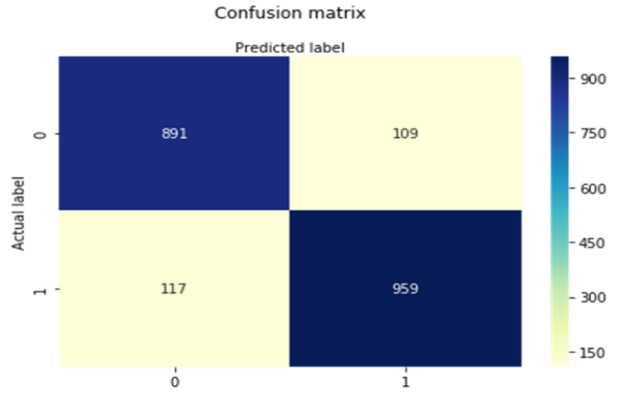

Figure 4. Confusion matrix of Decision Tree after preprocessing steps.

Table 5. Results of Decision Tree after preprocessing steps.

Pointer	Result
Correctly classified as 1	959
Incorrectly classified as 1	109
Correctly classified as 0	891
Incorrectly classified as 0	117
Precision	89.10%
Recall	88.39%
Accuracy	89.11%

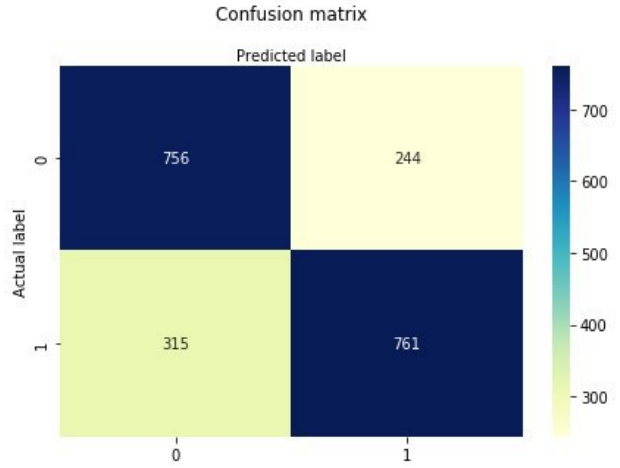

Figure 5. Confusion Matrix Random Forest before preprocessing steps.

Table 6. Results of Random Forest before preprocessing steps.

Pointer	Result
Correctly classified as 1	761
Incorrectly classified as 1	244
Correctly classified as 0	756
Incorrectly classified as 0	315
Precision	75.6%
Recall	70.58%
Accuracy	73.07%

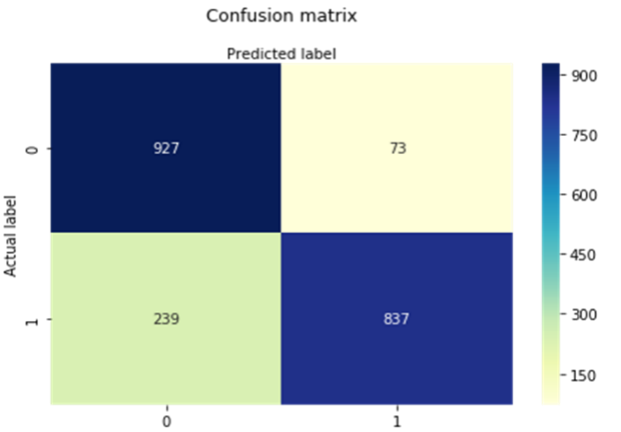

Figure 6. Confusion Matrix Random Forest after preprocessing steps.

Table 7. Results of Random Forest after preprocessing steps.

Pointer	Result
Correctly classified as 1	**837**
Incorrectly classified as 1	**73**
Correctly classified as 0	**927**
Incorrectly classified as 0	**239**
Precision	**92.1%**
Recall	**79.50%**
Accuracy	**84.97%**

From the results shown above, it appears that the decision tree outperforms better than a random forest in terms of accuracy, where the accuracy of the decision tree equals 89.11% while in random forest equals 84.97%. This is due to the characteristics and behavior of each algorithm and its effect on the dataset used. Based on our dataset, the features used' impotence plays an essential role in classification accuracy since the decision tree algorithm gives high importance to some features more than others. While in the random forest, the features are chosen randomly during the training phase, and it does not rely on specifics groups of features. It is also much more difficult and time-consuming to build their architecture in the random forest than in decision trees. It also requires more computing resources and is less intuitive when you have a large group of decision trees, and it is not easy to have an intuitive understanding of the relationship in the input dataset used.

For these reasons, the decision tree with this type of fake news dataset gives a better result than the random forest in the classifying process.

Additionally, in our results, the random forest prediction takes a longer time than the decision tree, where the time of running random forest is (20 min), while the decision tree is (10 min). Besides, Internal processes can be checked and thus allow the reproduction of work. After that, we compared our classification method's accuracy with the accuracy of other related works. Our results were given better accuracy than previous works, especially [9] that used the same dataset.

6. Conclusion and Future Work

In 2016, in the last US presidential elections. The problem of fake news received enormous attention. As new statistics and research emerged, those who spread such news on social media in the United States are about 62% adults. In the present research, we offered a detection model for the fake news using the TF-IDF features extraction technique. Additionally, we are using two different methods of ML algorithms. The realized model has achieved maximum accuracy in the case of using the decision tree classifier. The maximum accuracy score was 89.11%. The achieved result is better than the listed related work, so using this algorithm enhances the classification accuracy. From the results above, we conclude that:

1. The decision tree is better than a random forest in the classification accuracy of fake news dataset.
2. Random forest is more suitable for large datasets because more than decision trees are generated randomly and depend on voting between results to choose the best result.
3. The preprocessing steps using our dataset give better results. These steps had a significant impact on increasing the accuracy of the classification.
4. The type of dataset collected (Fake news articles of dataset) also has a significant impact on the classification accuracy of this work.

In the future, we suggested using other powerful classification algorithms like deep learning DNN such as LSTM, GRU, or CNN and using word embedding as feature extraction or classify Arabic dataset news.

References

[1] Gilda S., "Evaluating machine learning algorithms for fake news detection", 2017.

[2] Yang Y., "TI-CNN: Convolutiona Neural Networks for Fake News Detection", 2018.

[3] Mykhailo G. V. M., "Fake News Detection Using Naive Bayes Classifier", 2017.

[4] Priyanka S, "Detection of Fake Profiles on Twitter using Random Forest & Deep Convolutional Neural Network", 2019.

[5] Gentzkow H., "Social media and fake news in the 2016 election", 2017.

[6] Gilda S., "Evaluating Machine Learning Algorithms for the Detection of the Fake News", 2017.

[7] Chandra Mouli Madhav Kotteti, Na Li, "Fake News Detection Enhancement with Data Imputation", 2018.

[8] Arjun R., Asif Ekbal, Pushpak Bhattacharyya, "A Deep Ensemble Framework for Fake News Detection and Classification", 2018.

[9] Arvinder P.B., Sourabh C., and Mahima G. "Comparative Performance of Machine Learning Algorithms for Fake News Detection", (2019).

[10] Ahlem Drif, " Fake News Detection Method Based on Text-Features", 2019.

[11] Hadeer Ahmed, Sherif Saad, "Detection of Online Fake News Using N-Gram Analysis and Machine Learning Techniques", 2017.

[12] Suhad A. Yousif, Islam Elkabani, "Arabic Text Classification: The Effect of the AWN Relations Weighting Scheme". 2017.

[13] Suhad A. Yousif, Venus W. Samawi, "Utilizing Arabic WordNet Relations in ArabicText Classification:NewFeature SelectionMethods", 2019.

[14] Ugo Erra, FernandoM., "Approximate TF-IDF based on Topic Extraction from Massive Message Stream using the GPU", 2014.

[15] Suhad A. Yousif, Islam Elkabani, "The Effect of Combining Different SemanticRelations on Arabic Text Classification", Vol. 5, No. 6, 112-118, 2015.

[16] Suhad A. Yousif, Hussam Y. Abdul-Wahed, Nadia M. G. Al-Saidi, "Extracting a new fractal and semi-variance attributes for texture images", 2019.

[17] Sam F., "Decision Tree Classification with Differential Privacy: A Survey", 2019.

[18] Jehad Ali, R. K., Nasir Ahmad, Imran Maqsood, "Random Forests and Decision Trees", 2019.

[19] Suhad A.Yousif, Venus W. Samawi, Islam Elkaban, Rached Zantout, "Enhancement of Arabic text

classification using semantic relations of Arabic WordNet", 2015.

[20] Kai Shuy, Suhang Wangy, Jiliang Tang, Huan Liuy, "Fake News Detection on Social Media: A Data Mining Perspective", 2019.