Current Applied Science and Technology Vol. 23 No. 2 (March-April 2023)

*Research article*

# Features Extraction Based on Probability Weighting for Fake News Classification on Social Media

**Sherly Valentina and Wararat Songpan***

*Data Science and Artificial Intelligence Program, College of Computing, Khon Kaen University, Khon Kaen, Thailand*

Received: 25 November 2021, Revised: 5 January 2022, Accepted: 15 September 2022

DOI: 10.55003/cast.2022.02.23.014

## Abstract

**Keywords**

fake news;

machine learning;

sentiment analysis;

probability weighting;

data visualization

Fake news is a massive problem globally, especially on social media. Most people spend a lot of time consuming social media every day, and it is very possible for people as social media users to receive fake news without realizing it. Primarily due to this situation, we developed a machine learning tool to detect fake news that operates with the aid of various algorithms such as Decision Tree, K-Nearest Neighbor, and Naïve Bayes. Our experiement is tested based on machine learning that selected only one technique used to classify the data by finding the model set. In addition, the performance of the set describes the classification of the model and the inconsistency solution for each iteration. This study proposed a model which used the probability weighting of the model in features extraction processing for data classification. The concept is the enhancement of probability weighting features that converge exactly the class labels of classification. Our work was also implemented based on traditional Count Vectorizer and TF-IDF Vectorizer sentiment analysis and combined probability weighting features for fake news articles. The experimental results of the work illustrate that the best accuracy achieved by a proposed model used probability weighting features to find out the impact of classifiers models. In addition, the results of experimental information is represented by enhancing the overall performance of Decision Tree, K-Nearest Neighbor, and Naïve Bayes with various datasets. In addition, the measures of precision, recall, F1-measure, AUC, and accuracy for each class and deep in each class were achieved and reached the highest performance of the proposed model.

*Corresponding author: E-mail: wararat@kku.ac.th

## 1. Introduction

Fake news has become a major issue in a field of technology. Fake news contains false or misleading information that is contrary to real news. Fake news is created by irresponsible individuals or organizations for a variety of purposes, including product promotion, entertainment, health, and political and social issues. The fast speed of information movement and people's lack of awareness means that people are unable to check the veracity of the news and may read and believe them. Many of the platforms that spread fake news are social media like Facebook, WhatsApp, Instagram, YouTube, and others. There has been an incredible change as social media users share fake information without figuring it out. Social media is an interactive platform that allows users to express themselves through virtual networks by creating and sharing information. An interesting topic has become the detection of fake news. The detection of misleading and untrue news can help people in the world, especially social media users, know the truth and recognize what is real and fake information on social media.

Recently, fake news domains such as of PolitiFact domain have been set up to combat fake news and provide users with a look at the reality of their information [1]. Machine learning, sentiment analysis, and human-computer interaction offer good ways to solve fake news. One type of machine learning is supervised learning which involves classification and prediction. In sentiment analysis, the information is classified in a positive or negative way. In human-computer interaction, the cognitive aspects and human factor issues are defined. Furthermore, data visualization additionally performs an important role in supporting people to deal with the fake news. Data visualization is represented by means of assignment taxonomy; a word cloud is used in the qualitative evaluation of news, and they can help people to determine the content of articles. In data visualization, a network graph is used in the qualitative assessment of fake news, and a mapping of the articles [2]. It needs to be remembered that data visualization is an important part to display the results of an experiment. According to Yan *et al*. [3], data is visualized using nodes of images and graphics, text, table, or icon. It can significantly increase data processing and interpretation capabilities, and data visualization has emerged as an important tool for test data processing.

Aspect of sentiment analysis, a branch of Natural Language Processing (NLP), helps us to establish whether the object in social media is real news or fake news [4]. Count vectorizer and TF-IDF Vectorizer, which are sentiment analysis tools, were used in text pre-processing to extract news articles before testing them with a classifier model. Various models can be used to solve the problem of fake news data. However, problems that are to be evaluated deeply in classes require a high degree of accuracy of the classes. Differently, this paper improved the vector from the text in the fake news article by using text pre-processing extracted and used techniques that had huge vectors and filtered vectors i ordering by frequently words.

One of the limitations of current research is that most studies of fake news have a multidisciplinary methodology that brings in theories from linguistics, sociology, psychology, and other humanistic sciences. The challenge is then to research on classification models and convert the results of the models with data visualization without automatically relying on formerly used disciplines. However, research with classification models can achieve fake news classification but it is difficult to identify the most robust model with imbalance datasets.

For the reasons above, our proposed model will solve the problem of finding out the probability weight from a model to enhance the features and thus it can be used to test articles that contain fake news. The probability weights gathered as the features in the training data help us to create accurate classes. Therefore, our work can be divided into 2 main issues as follows: 1) Gathering probability weight features has impacted to increase the value of accuracy is based machine learning classifiers, and 2) Using the probability weight feature such as recall, precision, F1 score, AUC, and accuracy for each class to consistently and accurately measured the performance in real news and fake news.

## 2. Materials and Methods

### 2.1 Materials

Several papers we investigated had problems in detecting fake news. Dey *et al*. [5] and Raza and Ding [6] categorized fake news using linguistic analysis. The aim of linguistic analysis is to discover the patterns of fake news which focus on grammar. The authors of both studies agreed that linguistic analysis started with data set acquisition, data exploration or pre-processing, including Bag of Words data, syntax analysis,semantic analysis, and word vector. Bedi *et al*. [7] and Traylor *et al*. [8] tried to tackle the problem by first focusing on the characteristics of fake news on social media platforms. Bedi *et al*. [7] explained that the spread of fake news depended on psychological and social foundations, whereas Traylor *et al*. [8] explained that after recognizing the characteristic of fake news, the user must check the facts of that news. For this purpose, they used a fact checking platform. The author additionally produced a corpus for fake news identification to assist users to identify the patterns of fake news in a technical way. Ghinadya and Suyanto [9] and Thakur *et al*. [10] experimented using stance detection to detect fake news. In stance detection, the headlines and text of an article are used to describe and automatically detect the extraction components of news articles. The literature reviewthat follows is divided into three categories.

#### 2.1.1 Machine learning classifier

Base models in machine learning incorporate many algorithms such as Decision Tree, Naive Bayes, K-NN and so on. Kesarwani *et al*. [11] collected a dataset from the BuzzFeed News Organization and divided the dataset up into a training set and a testing set. Moreover, a confusion matrix was used to evaluate the performance of KNN. At the end of the experiment, the KNN classifier finds the best solution for detecting fake news for the research. Thakur *et al*. [10] presented a hybrid of Convolutional Neural Network (CNN) and Gradient Boosted Decision Tree for detecting fake news. The dataset was collected from Kaggle. For the extractor, Kaliyar [12] used Keyword Extractor and News Extractor to refine the keyword, and used a combination of machine learning models and deep learning models. For machine learning models, the author used Naïve Bayes, Decision Tree, Random Forest, K-Nearest Neighbor. And for deep learning models, the author used Convolutional Neural Network and Long Short-Term Memory. In this paper, sentiment analysis features were also used to arrange the articles from the dataset before testing them in the machine learning and deep learning models. For sentiment analysis, Term Frequency-Inverse Document Frequency (TF-IDF) was used in this study. The result showed the accuracy for Naïve Bayes, Decision Tree, Random Forest, K- Nearest Neighbor, and the combination of CNN and LSTM. Poddar *et al*. [13] presented a comparison of different machine learning models in training and testing the dataset collected by Kaggle. Before training the dataset, text preprocessing was done using Count Vectorizer and TF-IDF Vectorizer to tokenize the documents. For classification, the author used five machine learning classifiers such as Decision Tree, Naïve Bayes, Logistic Regression, Support Vector Machine and Artificial Neural Network. Support Vector Machine with TF-IDF Vectorizer achieved the highest accuracy followed by Logistic Regression with Count Vectorizer, Naïve Bayes with Count Vectorizer, Decision Tree with Count Vectorizer, and lastly is Artificial Neural Network. Bhutani *et al*. [14] collected the Kaggle dataset George McIntire and the ISOT dataset. Sentiment analysis consisted of two types of TF-IDF vectorizers such as TF-IDF with cosine similarity and TF-IDF without cosine similarity. For the machine learning classifier, the authors used Random Forest and Naïve Bayes. The results showed that using TF-IDF with cosine similarity improved accuracy. The feature extraction method showed that using TF-IDF with cosine similarity improved the accuracy of the classifier. The papers mostly compared machine learning models and focused on the method of text analysis and preprocessing with Count Vectorizer or TF-IDF. In addition, machine learning was applied to various models on fake news. Song *et al.* [15] explained about a new benchmark for detecting fake news, namely TGNF for temporal news propagation graphs. This paper discussed DAGA-NN (Domain-Adversarial & Graph-Attention Neural Network) development to identify fake news using domain discriminators [16]. The analysis above is summarized in Table 1.

As can be seen in Table 1, many researchers experimented with Fake News data sets and tried to use many classification models. The proposed model compares 2 main issues; text-processing and the classification models. Text-processing was performed using Count Vectorizer and TF-IDF methods, and these methods were used in combination with other models. In aspect of classification models, deep learning models are currently of interest; however, deep learning has the nature of randomization in a model and involves complex measuring. That makes such models time consuming and unstable to use. Therefore, the models such as Naïve Bayes, Decision Tree, Random Forest, K-Nearest Neighbor, Support Vector Machine, and Artificial Neural Network will focus on purposed feature exaction and feature extraction for purposed method development.

### 2.1.2 Sentiment analysis

Alonso *et al*. [4] pointed out that sentiment analysis is an essential feature of text analysis. Many researchers believe that sentiment analysis is a technique suitable for analyzing sentences or human statements when human beings are confronted with fake news. In detecting fake news, sentiment analysis divides the news into two categories which are the content and context of the news. Bhutani *et al*. [14] accrued the Kaggle dataset and the George McIntire and ISOT datasets. Sentiment analysis consists of two sorts of TF-IDF Vectorizer, TF-IDF with cosine similarity and TF-IDF without cosine similarity. The authors used Random Forest and Naïve Bayes as machine learning classifiers. The result shows that using TF-IDF with cosine similarity can enhance the accuracy.

### 2.1.3 The probability weighting

This section presents studies related to the probability weighting of classification models. The probability weight for each classification model illustrates the efficiency of the model.

A decision tree model is related to learning through classifying the test data into categories using the criterion of feature value. The tree structure is like a tree with branches and deals with the conditions. A decision tree needs the conditions to construct the decision tree model. A feature with high measure to the class is selected as the root node of the tree. Therefore, using the relationships between the features, the information gain values are employed. By selecting the feature with the highest information gain that can be calculated, and the probability weight of the decision tree measures the information required to classify any recording as equations. Kaliyar [12] mentioned that decision trees are good for supporting final decision and decision trees are usually used in machine learning processing.

First, equation (1) calculates the information required to classify any arbitrary instance X.

$$I(s_1, s_2, \ldots, s_m) = -\sum_{i=1}^{m} \frac{si}{s} log_2 \frac{si}{s}. \quad (1)$$

**Table 1.** Summarize of previous work in machine learning classifier

| Year | Authors | Domain Dataset | Text processing | Classification Model |
|---|---|---|---|---|
| 2018 | Kaliyar [12] | Fake or RealNews | Hashing Vectorizer | Machine Learning: Naïve Bayes, Decision Tree, Random Forest, K-Nearest Neighbor<br>Deep Learning: Convolutional Neural Network, Long Short-Term Memory |
| 2019 | Poddar *et al.* [13] | BS DetectorChrome Extension | Count VectorizerTF-IDF Vectorizer | Machine Learning: DecisionTree, Naïve Bayes, LogisticRegression, Support Vector Machine, Artificial Neural Network |
| | Bhutani *et al*. [14] | George McIntire | Bag of Words | Machine Learning: Random Forest and Naïve Bayes |
| 2020 | Kesarwani *et al*. [11] | BuzzFeed News Organization | TF-IDF | Machine Learning: K-NearestNeighbor |
| | Thakur *et al*. [10] | Fake NewsDetection | Count VectorizerTF-IDF | Machine Learning: GradientBoosted Decision Tree<br>Deep Learning: ConvolutionalNeural Network |
| 2021 | Song *et al*. [15] | Weibo Twitter API FakeNewsNet | Bert-wordvector | Machine Learning: DecisionTree, Support Vector Machine<br>Deep Learning: Recursive NeuralNetwork |
| | Yuan *et al*. [16] | TwitterWeibo | Vectorization | Machine Learning: Support Vector Machine, XGBoost<br>Deep Learning: Long Short-TermMemory, Domain-Adversarial & Graph-Attention Neural Network |

where S contains Si-instances of class $C_i$ for i = {1, …, m}

The entropy of feature F with values $\{F_1, F_2, \ldots, F_n\}$is:

$$E(F) = \sum_{j=1}^{v} \frac{s1j+\ldots+smj}{s} I(s_{1j}, \ldots, s_{mj}) \tag{2}$$

Where information gained by branching on feature F is:

$$Gain(F) = I(s_1, s_2, \ldots, s_m) - E(F) \tag{3}$$

K-Nearest Neighbors(K-NN) is a model used for classifying classes to predict which class could replace the condition or new instance by determining numbers of neighbors that were the same or closest to the data. The K-NN model compares data of interest with other data to establish similarity of distance. The equation below was used for calculating the distance weight. For example, from $(x_i, y_i)$ to $(x_j, y_j)$, when d was the distance as follows.

$$Distance\ weight(d) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \tag{4}$$

Kaliyar [12] explained that Naïve Bayes is suitable for real-world cases nowadays. Naïve Bayesian learning is a machine learning model based on the principle of probability as seen in Bayes' Theorem and includes a hypothesis that encourages independent features. According to the definition, the classification model generated the classifiers of the Naïve Bayes model as below,

$$\text{Naive Bayes Classifier } = Max(P(Class_i) \prod P(f_j|Class_i)) \tag{5}$$

where each instance F had n features or F = $\{f_1, \ldots, f_n\}$ and $Class_i$ as the class label. The Naïve Bayes classifier results were selected as the class with the maximum probability (MAP class).

From a machine-based learning model, classification models such as Decision Tree, K-NN and Naïve Bayes models, which are robust and have various features, are of interest. The advantage of those machine learning models is that the probability weight before solving and predicting classes is given. Therefore, the proposed model will be applied if the probability weighting was extracted to feature exaction processing in the methods section.

## 2.2 Methods

Our study started with data collection for fake news, i.e. 2 classes (Fake or Real news datasets). After collecting the data, the data went through the text preprocessing phase to transform the raw data into vectors before testing in the model. The next step was adding more features to support the algorithm. In this case we trained and tested the data using Decision Tree, K-Nearest Neighbor, Naïve Bayes and Probability Weighting Feature (PWF) Model. In the last step, we transformed the results into data visualization (word cloud) and also evaluate the experiment. The framework methodology in this study can be seen in Figure 1.

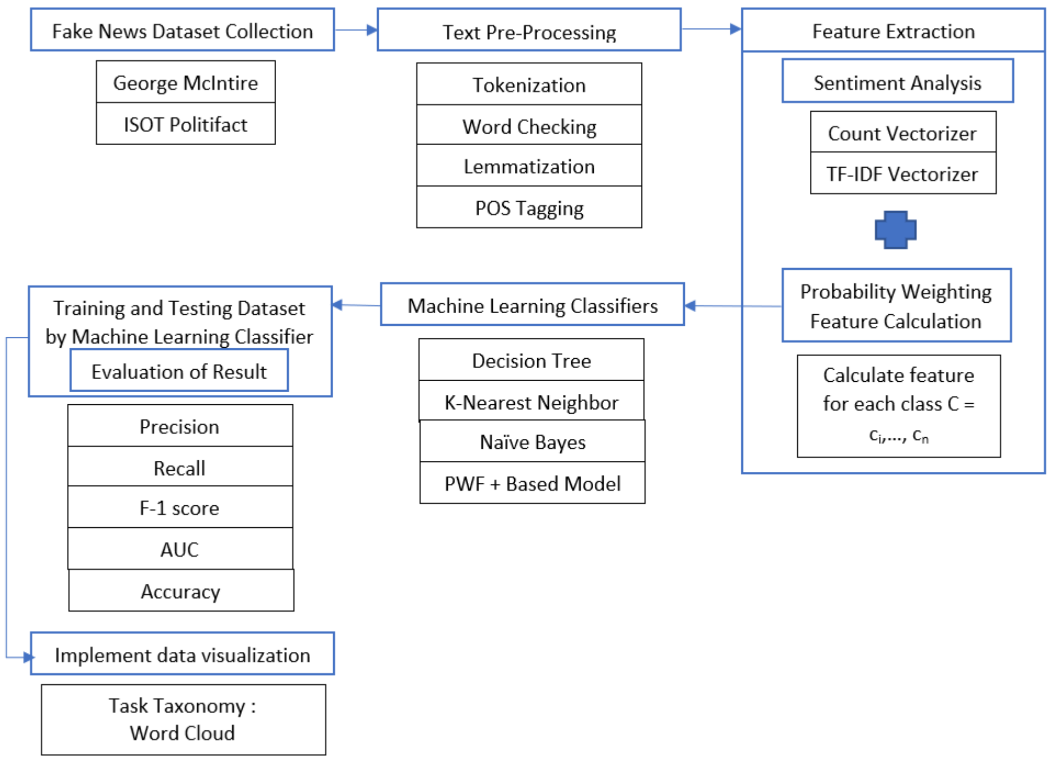



**Figure 1.** Framework of proposed model

### 2.2.1 Data collection

Our methodology involved two experiments. First, a fake news dataset from George McIntire in experiment I was used, which was called balance dataset. From George McIntire's dataset for fake instead of class 0 and real news instead of class 1 which was encoded into binary class labels as 0 and 1. The dataset I contained 3,171 real articles and 3,164 fake articles. In experiment II, the ISOT dataset was used, which contained 23,481 fake records and 21,417 real records. The two datasets were divided into a training set and a test set for experimentation and evaluation. The intention of this study was to discover and classify news articles in social media through machine learning using distinctive modeling methods (classifying the article as fake (class 0) or real (class 1) and predicting the accuracy for each model. For selection of classification models, we used a comparison of various machine learning models. We proposed the datasets from sources: George McIntire and ISOT. The dataset is divided into two datasets: training set and the test set. We evaluated the version performance using an evaluation matrix along with precision, recall, F1-score, AUC and accuracy.

### 2.2.2 Text pre-processing

In this part, before testing the algorithm, the dataset was divided into four parts. After performing tokenization, word checking, lemmatization, and part-of-speech tagging processes, the raw data was converted to a vector, and punctuation or other elements were removed from the data. The first step was tokenization in which the tokens represented the documents in the form of vectors. This involved splitting a document context into meaning units, called “words”. The next step was word checking, which involved removal of multiple lines, symbols, and special characters using python programming language such as \ [.*?\], https?://\S+|www\.\S+, and so on to clean the article text and replace capital letters with lower case characters. The processing of the removal of stop words is very important in Natural Language Processing because when stop words are removed, the classification model can classify an efficient sentiment meaning. The sentences are cleaned and can be categorized into types of words such as nouns, prepositions, and adjectives. The next step is called Lemmatizer, which is similar to stemming but gives the context of the words. The last step was part-of-speech tagging or POS, which refers to the ability to give a tag to every word in the wordnet. The implementation of POS tagging is in punctuation symbols. Bedi *et al*. [7] described that POS tagging determines each token in NLP activities to distinguish the words.

#### 2.2.3 Features exaction based on the probability weighting of classification model

In this study, the dataset was analyzed using Count Vectorizer and TF-IDF Vectorizer to measure the sentiment of the news. Count Vectorizer involves counting the frequency of each word that appears in a document. In addition, TF-IDF Vectorizer used to test in experiment, is a method of finding out data relevant to words. The frequency can be divided into two parts, firstly as term frequency (TF). TF means the number of times a word appeared in a document, and secondly as inverse document frequency (IDF), the inversed value of the number of times that the word appears in all documents. Poddar *et al*. [13] described that the Count Vectorizer generates an encoded vector that includes the duration of the whole vocabulary coupled with the frequency of every phrase through which it occurs in the record.

The Count Vectorizer method is an excellent method provided and used to convert a given text article into a vector based on the frequency counting of each word that appears inside the entire textual content. This method provides the many sentences in the sentences that have many words in a sentence. Each word is transformed in every textual content into a vector. In addition, it is very convenient to reduce the feature before creating a classification model.

The TF-IDF Vectorizer method (TF-IDF), which stands for Term Frequency-Inverse Document Frequency, is a statistical measure that evaluates how relevant a phrase is in information retrieval. TF-IDF is essential in the computerized textual content analysis in machine learning algorithms for Natural Language Processing (NLP), and can be seen in equation below:

$$TF - IDF\ (t, d) = tf(t, d) * idf(t, d) \tag{6}$$

Count Vectorizer and TF-IDF can be applied with other methods in which the feature is based on the probability weighting combined with traditional feature extraction. The probability weighting feature is calculated using Naïve Bayes as Algorithm1.

**Algorithm1:** Probability weighted feature process
**Required**
**Input:** Data set D = $\{d_1, d_2, \ldots, d_j\}$;
the class set C = $\{c_1, c_2, \ldots, c_n\}$;
n is the number of classes and j is the number of all features.
1: Initial features F= $\{f_1, f_2, \ldots, f_m\}$ by Count Vectorizer or TF-IDF Vectorizer (Eq. 6) methods for all training set D; m is the number of feature vectors;
2: While i=1; i <= n
3: Calculate the probability weighting of Fn+1 for each class $c_i, \ldots, c_n$ in equation 7;

$$PW_{m+i} = P(c_i)\prod P\left(F_j|c_i\right) \tag{7}$$

4: Generate probability weighting of PWm+i.
5: Go to step 2 until calculation of all the classes n
**Output**: Data set D with probability weighting for each class as new features.

**Algorithm 2:** Classifier with data set D which includes probability weight as new features.
**Required**
**Input**: training sample set F = ($f_1$, $f_2$,…, $f_m$, PW1, …, PWm);
the class set C = {$c_1$, $c_2$,…, $c_n$};
n is the number of classes and m is the number of feature vectors and probability weights as new features;
1: Base model classifier split train and test set.
2: Train model by training set.
3: Classifier class of testing set
4: Calculate accuracy and other the performances of models (equations 8-10)
**Output:** Class of fake news [0, 1]

### 2.2.4 Machine learning classifier

In this part, the dataset was used to test various algorithms including decision tree, K-nearest neighbor (KNN), Naive Bayes and Naive Bayes with Probability weighting (PWF), which is our proposed model. Our goal was to improve based algorithms in classification models with probability value as weight value close to class labels and thus to improve the accuracy in detecting fake news. The algorithm used libraries in scikit- learn and Python as open-source machine learning tools.

### 2.2.5 Evaluation

In this study, the performance of the results of the various algorithms was expressed in four different combinations of predicted and actual values such as:

TP (True Positive) indicates the reliability of the class 0 that was actually predicted by the model compared to the actual class 0.

FP (False Positive) indicates the reliability of the class 1 that was actually predicted by the model compared to the actual class 1.

FN (False Negative) indicates the reliability of the class 1 that was actually predicted by the model compared to the predicted class 0.

TN (True Negative) indicates the reliability of the class 0 that was actually predicted by the model compared to the predicted class 1.

They were evaluated using measure the following equations:
Precision should be as high as possible when all classes are predicted to be true positives.

$$Precision = \frac{TP}{TP+FP} \tag{8}$$

Recall should be as high as possible when all positive classes are divided by predicted positives.

$$Recall = \frac{TP}{TP+FN} \tag{9}$$

The F1-score is also used to measure recall and precision simultaneously.

$$F1-score = \frac{2*Recall*Precision}{Recall+Precision} \quad (10)$$

Accuracy should be as high as possible when all classes, both positive and negative, are divided by the sum of the possible actual and predicted values.

$$Accuracy = \frac{TP+TN}{TP+FP+FN+FP} \quad (11)$$

Area under the ROC curve (AUC) is a measure in this experiment design that involves using the two-dimensional area underneath the entire ROC Curve. AUC ranges in value from 0 to 1. A model that predicts correctly has an AUC of 1.0.

#### 2.2.6 Data visualization

In this part, the results from the evaluation part were represented by a word cloud to highlight the distinction between fake news and real news [2]. The advantages of using word cloud reveals that it is essential and fast process, easy to understand, and can show the most words that often appear.

## 3. Results and Discussion

### 3.1 Results

The fake news dataset was divided by the hold-out split method into 80% of train data and 20% of test data and simulated the independent training and test dataset. The results were evaluated by comparing performance evaluation methodology as precision, recall, F-1 score, AUC, and the percentage of accuracy obtained from using Decision Tree, K-NN, Naïve Bayes, and probability weighting feature (PWF), which is our proposed model. There were 2 experiments: experimental results I were tested on George McIntire data (Dataset I), and experimental results II were tested on ISOT datasets (Dataset II).

#### 3.1.1 Experimental result I

After testing the George McIntire dataset into a classification algorithm, the Naive Bayes with probability gave the best accuracy of the test, as shown in Table 2. The PWF enhanced Count Vectorizer with NB from 50.67 % to 80.27%. The DT and PWF decreased the accuracy of models and also K-NN and PWF had a slight percentage of accuracy. Other measures, precision, recall, and F-scores were balanced for each class as illustrated in different aspects (Table 2).

Table 3 shows that Naive Bayes with probability gave the best accuracy of the test; the impact of PWF on TF-IDF with NB was enhanced to 88.87%. The DT and PWF decreased the accuracy of models, and K-NN and PWF showed dramatic decrease of percentage of accuracy from 61.24% to 51.12%.

**Table 2.** The results of evaluation metrics with Count Vectorizer feature extraction with dataset I

| Dataset | George McIntire (6.335 articles: class 0: 3,171, class 1: Real 3,164) | | | | | | |
|---|---|---|---|---|---|---|---|
| **Classifier Models** | **Feature Extraction** | **Evaluation Methods** | **Precision** | **Recall** | **F1-score** | **AUC** | **Accuracy** |
| Decision Tree (DT) | Count Vectorizer | Hold Out 80-20% | 0: 0.80<br>1: 0.81 | 0: 0.81<br>1: 0.80 | 0: 0.81<br>1: 0.81 | 0.80 | 80.87 % |
| K-Nearest Neighbor (K-NN) | | | 0: 0.77<br>1: 0.81 | 0: 0.83<br>1: 0.76 | 0: 0.80<br>1: 0.79 | 0.79 | 79.32 % |
| Naive Bayes (NB) | | | 0: 0.50<br>1: 0.51 | 0: 0.59<br>1: 0.43 | 0: 0.47<br>1: 0.50 | 0.50 | 50.67 % |
| DT+PWF | Count Vectorizer and Probability Weighting | Hold Out 80-20% | 0: 0.77<br>1: 0.81 | 0: 0.81<br>1: 0.77 | 0: 0.79<br>1: 0.79 | 0.78 | 78.85 % |
| K-NN+PWF | | | 0: 0.76<br>1: 0.81 | 0: 0.81<br>1: 0.76 | 0: 0.79<br>1: 0.78 | 0.78 | 78.45 % |
| NB+PWF | | | 0: 0.80<br>1: 0.81 | 0: 0.80<br>1: 0.81 | 0: 0.80<br>1: 0.80 | 0.80 | 80.27 % |

**Table 3.** The results of evaluation metrics with TF-IDF Vectorizer feature extraction with dataset I

| Dataset | George McIntire (6,335 articles: class 0: 3,171, class 1: Real 3,164) | | | | | | |
|---|---|---|---|---|---|---|---|
| **Classifier Models** | **Feature Extraction** | **Evaluation Methods** | **Precision** | **Recall** | **F1-score** | **AUC** | **Accuracy** |
| Decision Tree (DT) | TF-IDF Vectorizer | Hold Out 80-20% | 0: 0.78<br>1: 0.79 | 0: 0.79<br>1: 0.81 | 0: 0.79<br>1: 0.79 | 0.80 | 78.69 % |
| K-Nearest Neighbor (K-NN) | | | 0: 0.56<br>1: 0.98 | 0: 1.00<br>1: 0.23 | 0: 0.72<br>1: 0.37 | 0.61 | 61.24 % |
| Naive Bayes (NB) | | | 0: 0.97<br>1: 0.77 | 0: 0.70<br>1: 0.98 | 0: 0.82<br>1: 0.86 | 0.84 | 84.21 % |
| DT+PWF | TF-IDF Vectorizer and Probability Weighting | Hold Out 80-20% | 0: 0.76<br>1: 0.77 | 0: 0.73<br>1: 0.79 | 0: 0.74<br>1: 0.78 | 0.76 | 76.38 % |
| K-NN+PWF | | | 0: 0.52<br>1: 1.00 | 0: 1.00<br>1: 0.06 | 0: 0.68<br>1: 0.12 | 0.51 | 51.12 % |
| NB+PWF | | | 0: 0.99<br>1: 0.82 | 0: 0.78<br>1: 1.00 | 0: 0.88<br>1: 0.90 | 0.88 | 88.87 % |

Figure 2 illustrates the comparison of AUC performance between classification models and proposed model with dataset I. The probability weighting feature combined with Count Vectorizer gave the best solution, gathering together NB and PWF, which is our proposed model. The experimental results applied Count Vectorizer and PWF, which also had an impact on the accuracy of the models, as shown in Figure 3. The experimental results indicate that feature extraction using probability weighting helps the performance of Naïve Bayes reach the best models dramatically based on the classification model as Naïve Bayes Model combined probability weighting feature.

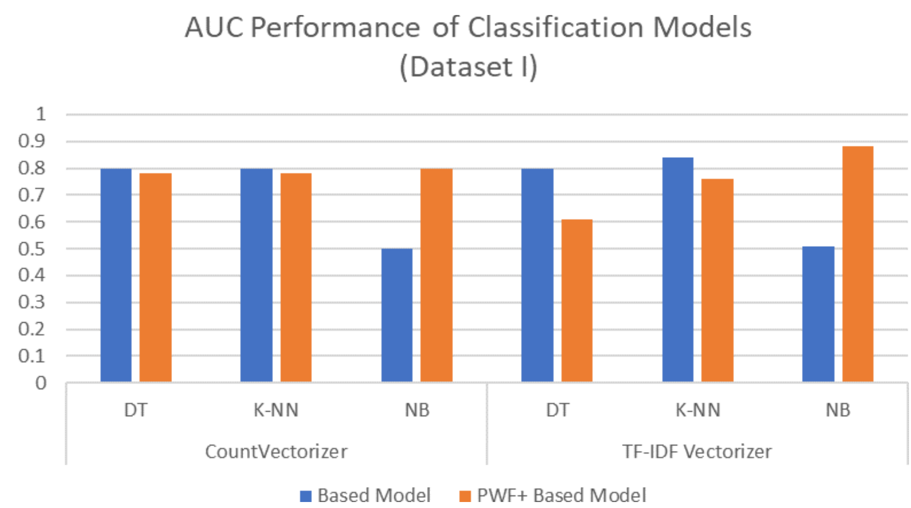



**Figure 2.** A comparison of AUC measure between the models and proposed methods with dataset I

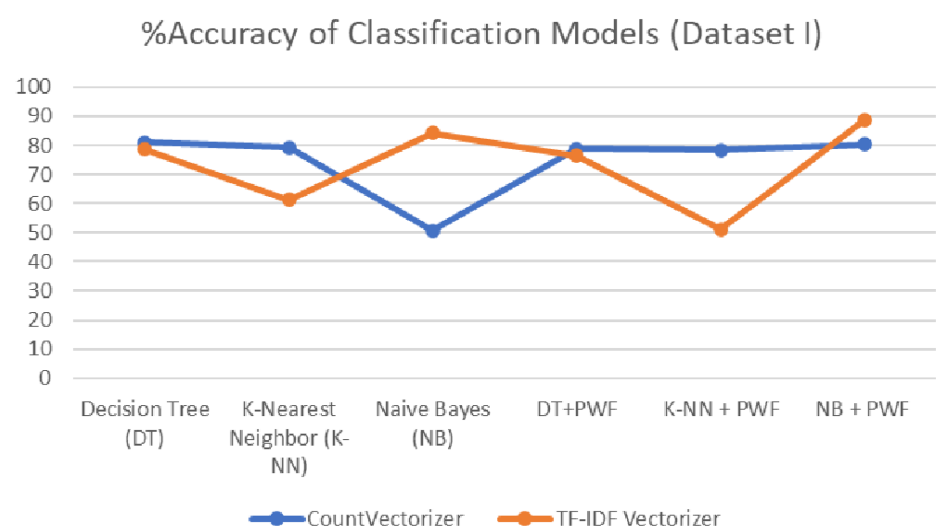



**Figure 3.** A comparison of the accuracy between models and PWF methods with dataset I

### 3.1.2 Experimental results II

The experiment tests of ISOT dataset had overall 44,898 records; a separate fake class as Class 0 with 23,481 records, and a real class as Class 1 with 21,417 records. Table 4 shows feature extraction by Count Vectorizer into a classification algorithm and probability weighting for different models. The results showed that Naive Bayes enhanced with the probability weighting feature gave the lowest accuracy of the test. In contrast, the Decision Tree enhance with the probability weighting feature gave the best accuracy of the experimental results.

**Table 4.** The result of evaluation metrics with Count Vectorizer feature extraction with dataset II

| Dataset | ISOT (44,898 articles: Class 0: Fake 23,481 and Class 1: Real 21,417) | | | | | | |
|---|---|---|---|---|---|---|---|
| **Classifier Models** | **Feature Extraction** | **Evaluation Methods** | **Precision** | **Recall** | **F1-score** | **AUC** | **Accuracy** |
| Decision Tree (DT) | Count Vectorizer | Hold Out 80-20% | 0: 0.80<br>1: 0.81 | 0: 0.81<br>1: 0.80 | 0: 0.81<br>1: 0.81 | 0.81 | 80.87 % |
| K-Nearest Neighbor (K-NN) | | | 0: 0.52<br>1: 0.49 | 0: 0.56<br>1: 0.44 | 0: 0.54<br>1: 0.46 | 0.50 | 50.39 % |
| Naive Bayes (NB) | | | 0: 0.52<br>1: 0.48 | 0: 0.56<br>1: 0.44 | 0: 0.54<br>1: 0.46 | 0.50 | 50.12 % |
| **PWF+DT** | Count Vectorizer and Probability Weighting | Hold Out 80-20% | 0: 0.82<br>1: 0.81 | 0: 0.81<br>1: 0.82 | 0: 0.82<br>1: 0.82 | 0.82 | 81.76 % |
| PWF+KNN | | | 0: 0.52<br>1: 1.00 | 0: 0.57<br>1: 0.45 | 0: 0.55<br>1: 0.47 | 0.51 | 51.18 % |
| PWF+NB | | | 0: 0.53<br>1: 0.49 | 0: 0.58<br>1: 0.45 | 0: 0.56<br>1: 0.47 | 0.52 | 51.95 % |

As shown in Table 4, the results of evaluation metrics with Count Vectorizer feature exaction with our proposed methods illustrates that feature extraction with Count Vectorizer with the Decision Tree model gave the highest accuracy at 80.87%, and the probability weighting feature with Count Vectorizer gave accuracy of 80.27%. However, the performances for precision, recall and F1-score were similar. Naïve Bayes with Count Vectorizer gave the lowest accuracy, which was 50.67%. The proposed model applied as feature with Count Vectorizer, the PWF and DT, gave the highest accuracy, which was 81.76%, whereas PWF applied to K-NN and NB had a slightly increased accuracy.

In Table 5, the TF-IDF feature extraction used with our proposed methods as probability weighting feature and Decision Tree gave 99.75% whereas probability weighting feature and Naïve Bayes, and probability weighting feature and K-NN gave accuracies of 96.21%, and 90.73%, respectively. However, the overall performance has the highest precision, recall, and F1-score as the best evaluation models. The K-NN model with TF-IDF gave the lowest accuracy value, which was 65.23%, However, the accuracy of probability weighting feature and K-NN gave a much higher value of 90.73%.

In dataset II complied with ISOT dataset, the AUC and the accuracy of TF-IDF Vectorizer combined PWF feature with Decision Tree achieved this overall performance when compared to the TF-IDF Vectorizer model combined PWF with K-NN and Naïve Bayes (Figures 4 and 5). The based model as Naïve Bayes gave the highest AUC and accuracy. In contrast, Naive Bayes and PWF were slightly higher than Naïve Bayes classifier. K-NN gave the lowest accuracy of the classification model, however, the conclusion of PWF helped to achieve AUC and accuracy performance of models.

**Table 5.** The results of evaluation metrics with TF-IDF feature extraction with dataset II

| **Dataset** | **ISOT (44,898 articles: Class 0: Fake 23,481 and Class 1: Real 21,417)** | | | | | | |
|---|---|---|---|---|---|---|---|
| **Classifier Models** | **Feature Extraction** | **Evaluation Methods** | **Precision** | **Recall** | **F1-score** | **AUC** | **Accuracy** |
| Decision Tree (DT) | TF-IDF Vectorizer | Hold Out 80-20% | 0: 0.80<br>1: 0.81 | 0: 0.81<br>1: 0.80 | 0: 0.81<br>1: 0.81 | 0.80 | 80.87 % |
| K-Nearest Neighbor (K-NN) | | | 0: 0.60<br>1: 0.96 | 0: 0.99<br>1: 0.2 | 0: 0.75<br>1: 0.44 | 0.65 | 65.23 % |
| Naive Bayes (NB) | | | 0: 0.96<br>1: 0.94 | 0: 0.94<br>1: 0.96 | 0: 0.95<br>1: 0.95 | 0.95 | 95.24 % |
| **PWF+DT** | TF-IDF Vectorizer and Probability Weighting | Hold Out 80-20% | 0: 1.00<br>1: 0.99 | 0: 0.99<br>1: 1.00 | 0: 1.00<br>1: 1.00 | **0.99** | **99.75 %** |
| PWF+KNN | | | 0: 0.96<br>1: 0.86 | 0: 0.86<br>1: 0.96 | 0: 0.91<br>1: 0.91 | 0.90 | 90.73 % |
| PWF+NB | | | 0: 0.97<br>1: 0.95 | 0: 0.95<br>1: 0.97 | 0: 0.96<br>1: 0.96 | 0.96 | 96.20 % |

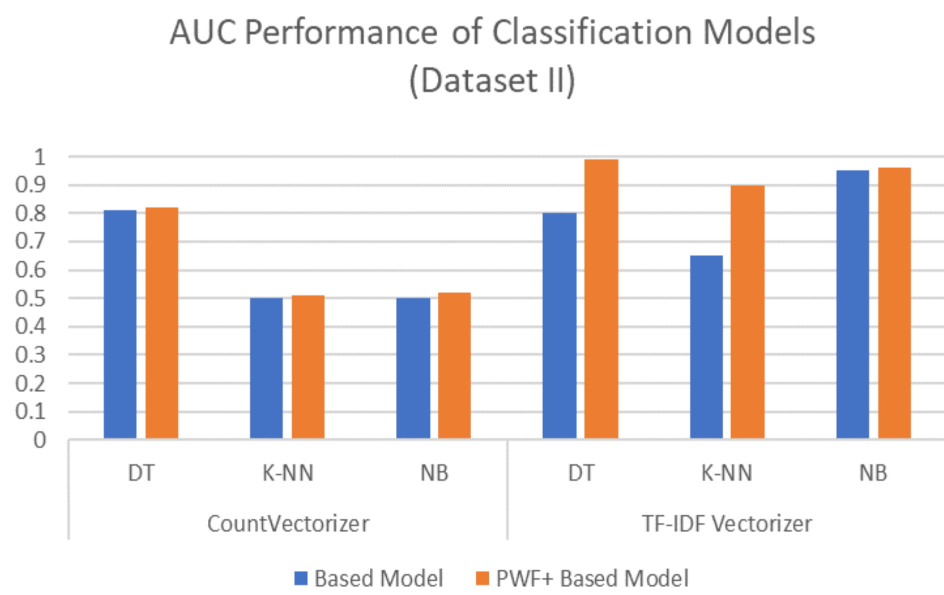



**Figure 4.** A comparison of AUC measure between models and proposed methods with dataset II

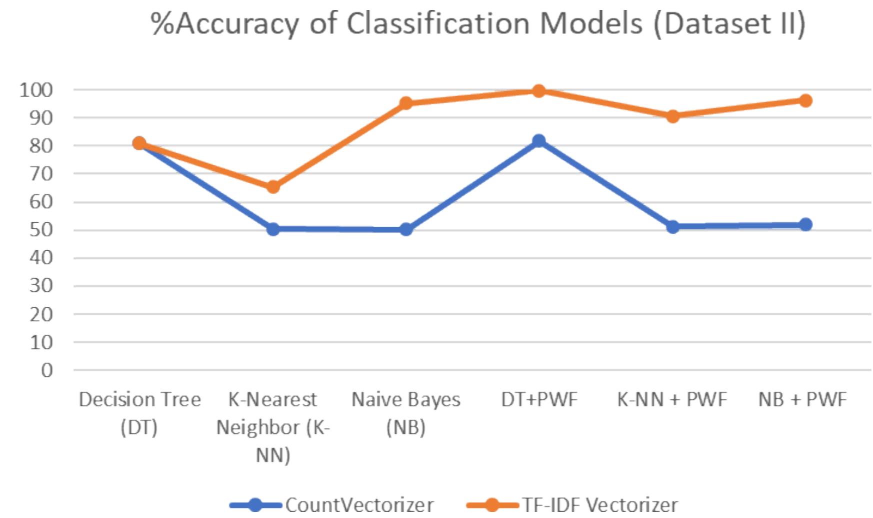


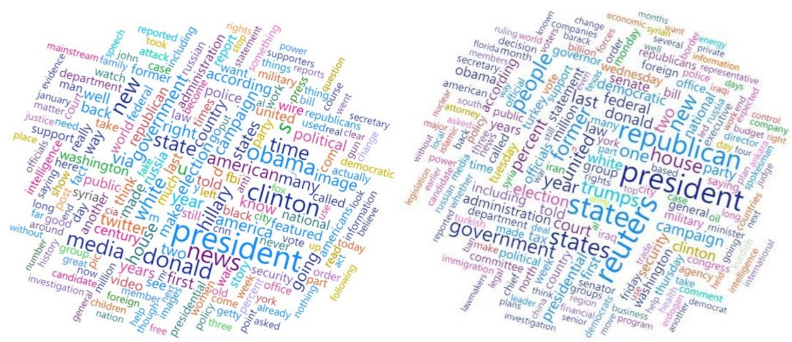

**Figure 5.** A comparison accuracy between models and PWF methods with dataset II

### 3.1.3 Word cloud visualization

For an overview of the dataset, our methodology performed a visualization of the dataset in order to have a better comprehension of the contexture of the dataset. The visualization took the form of a word cloud. Figure 6 illustrates many words related to political issues and domains. Fake news is mostly found in untrusted media. A characteristic of fake news is that it has a provocative title to catch people's interest to read their news. Otherwise, the word cloud visualization is indicated by the words as the most frequent word, for example, real news type is hard to read the overall text article. The frequent words enable human to interact, to read and to notice that the text article was classified by the word cloud visualization. The example news is one of the world's leading news organizations and produces realistic news articles. This proves that the news written is based on facts and comes from trusted sources. Also, the news is not focused on one aspect but is comprehensive depending on the latest conditions. In summarize, word cloud visualization normally helps us to segment the vocabulary frequent words in class labels. The advantage of model classifies is to make the articles more comprehensive than the exploration only on the output of class label from the classification model. The study on real and fake news domain datasets can be applied for data visualization that has seen frequent words in different classes gathering from data visualization analysis.

## 3.2 Discussion

From our proposed model, the results from general based classifiers model seem to have the best accuracy. We expanded the issue into two big points to describe the contribution of our model. Firstly, the experimental results in the dimension of overall performance, accuracy value, and experimental results showed that the Decision Tree and TF-IDF used for testing fake news data gave the best performance, which was 99.75%, TF-IDF used with our proposed model gave higher accuracy value than the models. Decision Tree with TF-IDF gave higher accuracy than the Decision Tree with Count Vectorizer. The K-Nearest neighbor model with Count Vectorizer gave the lowest account but the K-Nearest neighbor model with TF-IDF gave higher accuracy than with Count




**(a)** **(b)**

**Figure 6.** Word cloud of (a) fake news and (b) real news article in dataset II

Vectorizer. Therefore, the proposed model helped to produce a range of high accuracy and good performance that enabled the application of feature extraction by Count Vectorizer and TF-IDF Vectorizer. The new feature, which is probability weight, enhances the performance consistency. In addition, the probability weighting features converge exactly on the class labels of classification following the experimental results. The assumption of using the probability weight feature has the effect of increasing the value of accuracy for machine learning classifiers. Secondly, the issue is using the probability weight feature to measure other performances such as recall, precision, and F1-score for each class, for example, real news and fake news that are balanced. As seen in Table 2, the Naïve Bayes giving weak results for parameters such as recall meant that the number of accuracies of real news in class 0 for all real data in real news was 0.59. For the fake news classifier in class 1 of Naïve Bayes, the recall performance remained at 0.43 whereas the Naïve Bayes combined with the probability weighting feature and Count Vectorizer reached AUC at 0.80 which was higher than the traditional Naïve Bayes. In Table 3, the K-NN gave the recall at 0.23. For the fake news classifier in class 1 of Naïve Bayes, the recall performance remained 0.70 whereas the Naïve Bayes combined with the probability weighting feature and TF-IDF reached AUC at 0.88 which was higher than the traditional Naïve Bayes. In Table 4, the K-NN and Naïve Bayes gave the recall result of 0.56. For the fake news classifier in class 1 of Naïve Bayes and K-NN, the recall performance remained at 0.44 whereas the Decision Tree combined with the probability weighting feature and Count Vectorizer reached AUC at 0.82 which was higher than the traditional Decision Tree. In Table 5, the K-NN gave the lowest recall of 0.2 and the Decision Tree combined with the probability weighting feature and TF-IDF reached AUC at 0.99 which was higher than the traditional Decision Tree.

Therefore, probability weighting can be one of the feature vectors, which was mostly improved by applying a based classifier model in machine learning to enhance the efficiency of existing classification models. The efficiency of the prediction can be compared with the accuracy of classification models such as Decision Tree, K-NN and Naïve Bayes.

## 4. Conclusions

In this paper, we presented various types of algorithms for detecting fake news in social media. The text-mining stage included the steps of tokenization, word-checking, word lemmatization, and part-of-speech tagging to ensure that the dataset was suitable for testing the classification model. Our model proposed to enhance pre-processing feature extraction used probability weighting from classification models to prepare a fake news article. The proposed model with PWF produced the best percentage of accuracy compared with other based classification models. In aspect of performances such as precision, recall, F-score selected to solve class of fake news in real and fake news are illustrated in the experimental results with different datasets with the combination of probability weighting feature. Our purposed model also added sentiment analysis to determine social media user's views of articles with real or fake news in terms of cognitive aspects and human factors and presented in a data visualization that can help people understand the distinction or characteristics of fake news. The experimental results show that the Naive Bayes algorithm with probability weighting can improve accuracy and other aspects. The parallel implementation from the results shows that data visualization using word illustrates differences classes between articles.

In future work, the proposed model can be used to study other feature extractions, for example, the methodology of supervised term weighting and other term frequency-relevance frequency for improvement and robustness for classifier models. In addition, deep learning will be compared with other probability weighing feature.

## References


[1] Shu, K., Mahudeswaran, D., Wang, S., Lee, D. and Liu, H., 2018. FakeNewsNet: A data repository with news content, social context and Spatialtemporal information for studying fake news on social media. *Big Data*, 8(3), 171-188, DOI: 10.48550/arxiv.1809.01286.

[2] Shu, K., Mahudeswaran, D. and Liu, H., 2018. Fakenewstracker: A tool for fake news collection, detection, and visualization. *Computational and Mathematical Organization Theory*, 25(1), 60-71, DOI: 10.1007/s10588-018-09280-3.

[3] Yan, H., Wang, J. and Xia, C., 2017. Research and application of the test data visualization. *Proceedings of 2017 IEEE 2nd International Conference on Data Science in Cyberspace (DSC)*, Shenzhen, China, 26-29 June 2017, pp. 661-665.

[4] Alonso, M.A., Vilares, D., Gómez-Rodríguez, C. and Vilares, J., 2021. Sentiment analysis for fake news detection. *Electronics*, 10(11), 1-32, DOI: 10.3390/electronics10111348.

[5] Dey, A., Rafi, R.Z., Parash, S.H, Arko, S.K. and Chakrabarty, A., 2018. Fake news pattern recognition using linguistic analysis. *Proceedings of 2018 Joint 7th International Conference on Informatics, Electronics & Vision (ICIEV) and 2018 2nd International Conference on Imaging, Vision & Pattern Recognition (icIVPR)*, Japan, 25-29 June 2018, pp. 305-309.

[6] Raza, S. and Ding, C., 2022. Fake news detection based on news content and social contexts: a transformer-based approach. *International Journal of Data Science and Analytics*, 13(1), 335-362, DOI: 10.1007/s41060-021-00302-z.

[7] Bedi, A., Pandey, N. and Khatri, S.K., 2019. A framework to identity and secure the issues of fake news and rumours in social networking. *Proceedings of 2019 2nd International Conference on Power Energy, Environment and Intelligent Control (PEEIC)*, Noida, India, 18-19 October 2019, pp. 70-73.

[8] Traylor, T., Straub, J., Gurmeet and Snell, N., 2019, Classifying fake news articles using natural language processing to identify in-article attribution as a supervised learning estimator. *Proceedings of 2019 IEEE 13th International Conference on Semantic Computing (ICSC)*, New Port Beach, USA, 30 January-1 February 2019, pp. 445-449.

[9] Ghinadya and Suyanto, S., 2020. Synonyms-based augmentation to improve fake news detection using bidirectional LSTM. *Proceedings of 2020 8th International Conference on Information and Communication Technology (ICoICT)*, Yogyakarta, Indonesia, 24-26 June 2020, pp. 1-5.

[10] Thakur, A., Shinde, S., Patil, T., Gaud, B. and Babanne, V., 2020. MYTHYA: Fake news detector, real time news extractor and classifier. *Proceedings of 2020 4th International Conference on Trends in Electronics and Informatics (ICOEI)(48184)*, Tirunelveli, India, 15-17 June 2020, pp. 982-987.

[11] Kesarwani, A., Chauhan, S.S. and Nair, A.R., 2020. Fake news detection on social media using K-Nearest Neighbor classifier. *Proceedings of 2020 International Conference on Advances in Computing and Communication Engineering (ICACCE)*, Las Vegas, USA, 22-24 June 2020, pp. 1-4.

[12] Kaliyar, R.K., 2018, Fake news detection using a deep neural network. *Proceedings of 2018 4th International Conference on Computing Communication and Automation (ICCCA)*, Noida, India, 14-15 December 2018, pp. 1-7.

[13] Poddar, K., Geraldine, B.A.D. and Umadevi, K.S., 2019. Comparison of various machine learning models for accurate detection of fake news. *Proceedings of 2019 Innovations in Power and Advanced Computing Technologies (i-PACT)*, Vellore, India, 22-23 March 2019, pp. 1-5.

[14] Bhutani, B., Rastogi, N., Sehgal, P. and Purwar, A., 2019. Fake news detection using sentiment analysis. *Proceedings of 2019 Twelfth International Conference on Contemporary Computing (IC3)*, Noida, India, 8-10 August 2019, pp. 1-5.

[15] Song, C., Shu, K. and Wu, B., 2021. Temporally evolving graph neural network for fake news detection. *Information Processing and Management*, 58(6), 1-18, DOI: 10.1016/j.ipm.2021.102712.

[16] Yuan, H., Zheng, J., Ye, Q., Qian, Y. and Zhang, Y., 2021. Improving fake news detection with domain-adversarial and graph-attention neural network. *Decision Support Systems*, 151(1), 1-11, DOI: 10.1016/j.dss.2021.113633.