Development of a machine learning algorithm for fake news detection

Nur Atiqah Sia Abdullah[1,3], Nur Ida Aniza Rusli[2,3], Nurshaheeda Shazlin Yuslee[1]
[1]College of Computing, Informatics and Mathematics, Universiti Teknologi MARA, Shah Alam, Malaysia
[2]College of Computing, Informatics and Mathematics, Universiti Teknologi MARA, Cawangan Negeri Sembilan, Kampus Kuala Pilah, Negeri Sembilan, Malaysia
[3]Knowledge and Software Engineering Research Group (KASERG), Universiti Teknologi MARA, Shah Alam, Malaysia

Article Info

Article history:

Received Nov 3, 2023
Revised Apr 25, 2024
Accepted May 7, 2024

Keywords:

Decision tree
Fake news
Machine learning algorithm
Naïve Bayes
Social media
Support vector machine
Twitter

ABSTRACT

With the extensive technological advancements and expansion, the persistent issues regarding the creation and rapid dissemination of fake news have become a prevalent and recurrent concern. The manipulation of news content has critical repercussions, such as causing public mistrust, fear, harm, and misinformation. Addressing that, this study developed a supervised machine learning algorithm that can accurately classify social media data as fake news. The methodology of the proposed fake news detection model involved five main components: data acquisition from Twitter, data preprocessing, data transformation, model development using Naïve Bayes, decision tree, and support vector machine (SVM) and model evaluation using accuracy, precision, recall and F1-score. The results revealed that decision tree recorded the highest accuracy for both textual data (100%) and metadata (94.54%) and consistently outperformed both Naïve Bayes and SVM in terms of precision, recall, and F1-score metrics, with a score of 100% for the classification of textual data-based datasets. Regarding the metadata-based classification, decision tree also demonstrated excellent performance, with the highest F1-score of 94% for fake news data. Meanwhile, SVM exhibited the highest precision and recall performance for the metadata-based classification. Overall, the application of the decision tree classifier was deemed the most effective in Twitter fake news detection.

Corresponding Author:

Nur Ida Aniza Rusli
College of Computing, Informatics and Mathematics, Universiti Teknologi MARA
Cawangan Negeri Sembilan, Kampus Kuala Pilah
72000 Kuala Pilah, Negeri Sembilan, Malaysia
Email: idaaniza@uitm.edu.my

1. INTRODUCTION

Technological advancements and expansion have contributed to the development and improvement of communication means. The growing dominance of technologies and the proliferation of social media platforms like Twitter and Facebook have transformed how we communicate with one another. Despite the significant benefits of such advancements, there are negative repercussions to the members of society [1], [2]. The persistent issues of fake news distorting information are not part of a new phenomenon. When it comes to the dissemination of fake news and false information, particularly across social media platforms, there are critical implications at the individual, organizational, and societal levels.

The circulation of fake news content can create public bias [3], confusion [3], [4], panic [5], and mistrust [6], as well as poor confidence in the government or other relevant institutions [7]. Considering these critical repercussions, addressing issues concerning fake news on social media platforms is pivotal. Fake news and spam messages share similar attributes. For example, there are common issues like grammatical errors [8], deliberate manipulation of public opinions [9], and recurrent inclination towards dissemination of inaccurate or misleading information [10]. Additionally, fake news uses of the same limited lexicon for content simplification [11].

Machine learning have been widely used to determine the authenticity of news articles. Studies have proposed various methodologies to detect fake news at higher accuracy, such as the combination of machine learning with text-based data [12], [13], neural networks [14], [15], and deep learning techniques [16], [17]. Although the application of machine learning in fake news detection offers favorable outcomes, certain issues and limitations need to be addressed, such as the constant evolution of words [18], the use of abbreviations [19], and the complexity of unstructured contents [20], which substantiate the need for more studies to enhance the accuracy of machine learning models.

One of the earliest approaches to detect and classify fake news, which involved the application of machine learning algorithms was proposed in [21]. The study suggested a straightforward approach for detecting and classifying fake news, specifically the use of Naïve Bayes classifier, and examined its effectiveness in detecting and classifying fake news on the Facebook platform. With the help of a software system, the study evaluated a manually labeled dataset of true news and fake news, which revealed the model accuracy of correctly classifying fake news at 71.73% and true news at 75.59%. Based on these results, the study proposed Naïve Bayes as an efficient machine learning classifier in detecting fake news, and it highlighted the promising potential of the integration of other machine learning techniques to address issues related to fake news.

Focusing on the extraction of news content and social context features, another study [22] proposed a different approach of detecting fake news using machine learning. The study extracted and classified the features into linguistic-based, visual-based, user-based, generated-post, and network-based features. The derived features were subsequently employed in the fake news detection process that involved model-oriented techniques like Naïve Bayes or support vector machine (SVM). The study further emphasized the significance of examining the efficiency of a machine learning model using performance metrics like precision, recall, F1, and accuracy. Based on the gathered empirical findings, the study recommended a comprehensive combination of news content and social context features to improve the performance of a machine learning classifier in fake news detection.

In recent years, the use of metadata and networking features as part of the machine learning models for fake news detection has been proposed. For instance, a recent study explored the incorporation of metadata features, such as subject, context, speaker, targeting, and statement, into the study's proposed system, XFAKE [14], to examine its system effectiveness and evaluate user understanding of the methodology of the proposed approach. Relevant metadata features were extracted from the dataset that was obtained from PolitiFact website. With the assistance of XGBoost classifier, all data entries were subsequently labelled as true and false. Regarding the user understanding, the evaluation procedure involved human assessment from amazon mechanical turk (AMT). Apart from the trade-off between the speed and accuracy of the generated explanations, the study emphasized the need for meticulous review and interpretation of these explanations for higher accuracy.

On the other hand, another study opted to integrate networking features [16]. The study proposed the use of a sentence-comment co-attention subnetwork technique, specifically explainable fake news detection (dEFEND), to detect fake news content and user comments. The proposed approach incorporated three subcomponents: news content encoder, user comment encoder, and sentence co-attention. The use of recurrent neural network (RNN) and gated recurrent units (GRU) in the study served to identify significant sentences and user comments within the context of fake news.

Meanwhile, preprocessing pipelines have recently gained growing interest. For instance, data preprocessing techniques, such as stop word removal, punctuation removal, and stemming using the natural language toolkit (NLTK) library, were employed in one of prior studies [23]. The study used Naïve Bayes, SVM, and Passive Aggressive classifiers to classify the pre-processed data and evaluated the model performance in terms of accuracy, precision, recall, and F1-score. The obtained results revealed SVM as the best model for fake news detection. All three classifiers showed good performance in terms of accuracy, precision, and F1-score (>80%). The study concluded the significant influence of data preprocessing techniques on the performance of classifiers in fake news detection and classification.

Besides that, there have been attempts of developing fake news detection in a language other than English. Jamaleddyn *et al.* [24] introduced the Arabic language for the proposed fake news detection system that integrated natural language processing (NLP) into several machine learning models.

The study performed a comparative analysis to examine the efficiency of hyperparameter tuning techniques, specifically in adapting grid search and random search methods for the proposed system. With that, the study improved the hyperparameter tuning techniques with three machine learning models: artificial neural network (ANN), multinomial logistic regression (MLR), and SVM. The study then evaluated the performance of these hyperparameter tuning techniques according to specific performance measures and found that the performance of random search method surpassed the performance of the grid search method.

In addition to identify fake news content, researchers have proposed alternative approaches, focusing on the detection of fake accounts [25], [26]. For example, [27] introduced a predictive stance for social network rumors (PSSNR) framework to improve the prediction of user stance. The proposed framework uses two types of data: conversation threads and user-based features, and the methodology of the study consists of two main phases: data preparation and the application of different machine learning algorithms using an augmented dataset.

The study conducted experiments using several machine learning algorithms and combinations of features through both pre- and post-data augmentation. The results indicated that augmenting data in deny and support classes together with combinations of content and user features had shown a significant improvement. The proposed framework outperformed state-of-the-art results, achieving a notable increase in macro F1-score from 0.672 to 0.7233.

Another significant contribution on the detection of fake news through machine learning presented by [28]. The study utilized linguistic features with an ensemble of several text classifiers through incorporating two voting classifiers. The first voting classifier comprised of logistic regression, random forest, and k-nearest neighbors (KNN), while the second voting classifier system includes the logistic regression, linear SVM and classification and regression trees (CART) to identify the fake news. The study focuses on linguistic features such as punctuation, word-associated emotions (positive or negative), and grammar. Despite its effectiveness on its methodology through the two-voting system, the study centers solely on the textual data. This limitation may result in overlooking valuable insights that other attributes, such as metadata, could contribute to the context of fake news detection.

Thus, this study proposed the development of a supervised machine learning algorithm that can accurately classify social media data as fake news or true news. The main contribution of this study lies in the implementation of two aspects of data, textual data, and metadata features. This study focuses on the content-specific keywords in the political and health domains to improve the performance of the proposed model. In addition to textual data, this study explores tweet-related features, including the number of followers, retweets, number of likes, and tweet counts, in order to implement a thorough analysis of the correlation between these features.

This study is also contributed by inducing both textual data and metadata features into three machine learning classifiers: decision tree, supervised machine learning, and Naïve Bayes. The proposed algorithm further employed grid search functions to optimize the hyperparameters of the classifiers and improve their generalization abilities. Overall, this paper is organised as follows: section 2 describes the methodology for this study; section 3 presents the obtained results; section 4 discusses the results and future works; section 5 concludes the overall study.

2. METHOD

This section presents a comprehensive overview of the methodology employed in this study, including Twitter data acquisition, data preprocessing, data transformation, classification model development, feature extraction, and model evaluation. The development and evaluation of the fake news detection model are conducted using Python programming language. The detailed processes are presented in Figure 1.

2.1. Twitter data acquisition

Real-time tweet data were obtained through API access tokens and keys from Twitter. The requests to Twitter were authenticated using API access tokens and keys. With that, access to the data was acquired. Considering the scope of the study on healthcare and politics, a Twitter scraping command was used for the aggregation of data like user ID, tweet text, language, account creation date, source, number of retweets and likes, Twitter join date, verified user status, number of followers, number of followings, location, tweet and quote count by using keywords (e.g., “COVID-19” and “vaccine”). This comprehensive dataset enabled a targeted analysis of within the specific domains of healthcare and politics of this study, facilitating a nuanced approach to detecting fake news on Twitter.

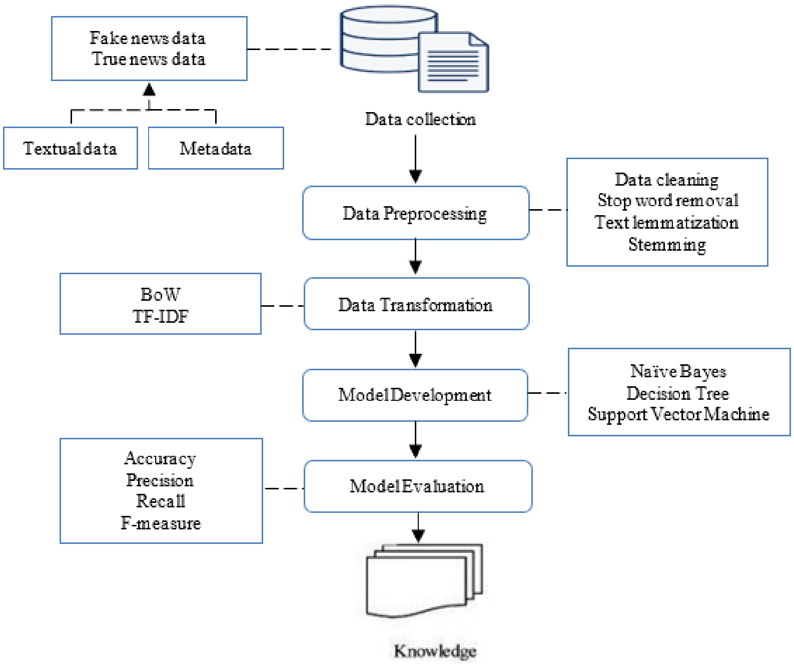


Figure 1. Methodology of Twitter fake news detection model development and evaluation

2.2. Data preprocessing

All collected data were then subjected to data preprocessing. Essentially, irrelevant, and non-textual data like special characters, numbers, and punctuation can be eliminated through the cleaning process, which was executed separately for the textual data and metadata features in this study. In most cases, there is noise and irrelevant information, including stop words and punctuation marks, in textual data, which must be removed. Besides that, the presence of missing values or outliers in metadata features like the number of likes, retweets, and followers requires specific techniques for efficient handling.

The overall data preprocessing process involved converting textual content to lowercase for standardization. Following that, all hyperlinks, punctuation, username, and whitespace in the textual data were eliminated. Adding to that, an empty string was used to replace any term with the hashtag sign (#) within the textual data.

The study proceeded to stop word removal and text lemmatization and stemming. Stop word removal in this study involved eliminating words with not much meaning in a context and minimizing the data noise, which was intended to improve the performance of the machine learning model. On the other hand, text lemmatization and stemming involved converting textual data to its root form, which served to simplify the textual data without losing the semantic meaning.

Metadata of tweets in this study were subsequently subjected to data cleaning. Features like “source”, “user.id”, “user.screen_name”, “user.description”, “user.verified”, “user.status_count”, “user.friends_count”, “user.followers_count”, “user.favourites_count”, “user.created_at”, “user.geo_enabled”, “user.lang”, “place”, “is_quote_status”, “retweet_count”, and “favorite_count” were eliminated due to their lack of contribution for the study’s data classification task. These features were eliminated from the dataset through Python.drop() built-in function. The remaining features like “created_at”, “id”, “full_text”, and “user.location” were retained to facilitate the subsequent data processing and feature selection.

2.3. Data transformation

After data preprocessing, the study proceeded to data vectorization, which was deemed pivotal in transforming textual data into an appropriate numerical structure for the machine learning algorithms. In this case, a specific term or word in the text document was converted into a matrix of numerical values or numerical vectors, which resulted in a comprehensive feature matrix. Data transformation in this study specifically involved the application of bag-of-words (BoW) and term frequency-inverse document frequency (TF-IDF).

In BoW, documents were clustered based on the occurrence frequency of words within the documents. Each document in this study was represented as a vector, in which each dimension corresponded to a term in the dictionary within the corpus. As a result, a document-term matrix form was generated.

A standard representation of BoW document-term matrix, which captured the relationship between the documents and the terms in the documents, is presented in Figure 2.

$$\begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1p} \\ w_{21} & w_{22} & \cdots & w_{2p} \\ \vdots & \vdots & \cdots & \vdots \\ w_{n1} & w_{n2} & \cdots & w_{np} \end{bmatrix}$$

Figure 2. BoW document-term matrix representation

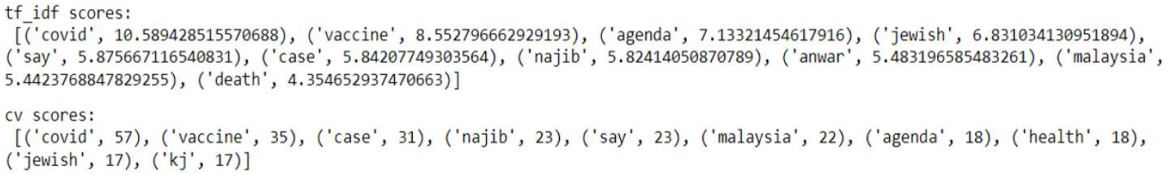
Where the row section, *n*, refers to the number of tweets in the dataset, the column section, *p*, refers to the unique terms related to the domain of healthcare and politics.

Meanwhile, TF-IDF is regarded as one of the most efficient statistically weighted, measured approaches for textual data analysis. TF-IDF in this study was employed to determine the importance of a word within a document according to its frequency in a corpus of documents, which is calculated as follows:

$$\mathrm{TF}_{(cl,td)} = \frac{\sum_{i=1}^{n} cl_i \in tw}{\sum_{j=1}^{m} tw \in td} \tag{1}$$

where *cl* denotes the classification of Twitter document (i.e., fake news and true news), *td* denotes the Twitter document (i.e., the domain of healthcare and politics), *i* denotes the instance in the document, *n* denotes the total number of collected data, *tw* denotes the word in a tweet, *m* denotes the total number of feature, *j* denotes the instance of the feature, *cd* denotes the corpus data.

Considering the application of both BoW and TF-IDF in Python, an appropriate library was considered. In addition, count vectorizer and TF-IDF vectorizer were employed in this study to facilitate the overall process of data transformation. Referring to Figure 3, the resultant output of the first 10 words obtained through the application of count vectorizer and TF-IDF vectorizer revealed the word “covid” as the most used word in both vectorization techniques.



```
tf_idf scores:
 [('covid', 10.589428515570688), ('vaccine', 8.552796662929193), ('agenda', 7.13321454617916), ('jewish', 6.831034130951894),
('say', 5.875667116540831), ('case', 5.84207749303564), ('najib', 5.82414050870789), ('anwar', 5.483196585483261), ('malaysia',
5.4423768847829255), ('death', 4.354652937470663)]

cv scores:
 [('covid', 57), ('vaccine', 35), ('case', 31), ('najib', 23), ('say', 23), ('malaysia', 22), ('agenda', 18), ('health', 18),
('jewish', 17), ('kj', 17)]
```

Figure 3. The output of first 10 words through the application of count vectorizer and TF-IDF vectorizer

2.4. Model development

The obtained results from data transformation were utilized for the construction of classification models. Naïve Bayes, decision tree, and SVM algorithms were employed for the classification model development in this study. As a classification machine learning algorithm, Naïve Bayes simplifies the data learning process through generative assumptions and parameter estimations. With the integration of “naïve” assumptions, data categorization or classification was performed based on the theorem of Bayes, in which all features in a document are independent of one another. It is also known as “probabilistic classification”, as it can predict the distribution of probabilities across a set of features using the following formula:

$$P\ (A|B) = \frac{P(B|A) * P(A)}{P(B)} \tag{2}$$

where *A* denotes the classification of Twitter document (i.e., fake news and true news), *B* denotes the set of features in Twitter document.

Among the listed algorithms, decision tree offers the simplest approach for data categorization or classification. It consists of root nodes, branches, and terminal nodes, which form a tree-like structure that generates possible outcomes from a series of sequential choices. The information gain formula was used to construct an effective decision tree. Through this formula, the best attributes for data splitting in a decision tree are identified, resulting in the identification of attributes that yield the highest information gain. Following that, the information gain related to a particular attribute was computed based on in (3). Accordingly, the concept of information gain is based on the evaluation of entropy, which is a measure of impurity within a dataset in (4) and (5).

$$Gain(A) = Info(D) - \ Info_A(D) \tag{3}$$

where *Info(D)* refers to the initial entropy of dataset, *D*, $Infor_A(D)$ refers to the weighted entropy of attribute, *A*, in dataset, *D*.

$$Info(D) = -\sum_{i=1}^{m} p_i \ log_2 \ (p_i) \tag{4}$$

where *p* refers to the probability of instances belong to class labels, *m* refers to the number of distinct class label (i.e., fake news or true news).

$$Info_A \ (D) = \ \sum_{j=1}^{V} \ \frac{|D_j|}{|D|} \times I\left(D_j\right) \tag{5}$$

where *D* denotes the input features in the Twitter document, *v* denotes the total number of features in the collected data.

Last but not least, SMV is a popular type of supervised machine learning that analyses data and recognizes patterns for data classification and regression. When it comes to the context of classification issues, SVM training algorithm employs the form of non-probabilistic binary linear classifier, which results in a model that can effectively divide data points into distinctive categories. Its ability to utilize the concept of high-dimensional space in (6) contributes to its capacity to deal with multifaceted attributes:

$$w.x + b = 0 \tag{6}$$

where *w* refers to the weight vector, *x* refers to features in the Twitter document (i.e., fake news and true news), *b* refers to the bias term.

Data classification was extended to both textual data and metadata. The application of data classification models in this study involved importing the Scikit-Learn library. The process included the adoption of classifier models using commands like “from sklearn.naive_bayes import MultinomialNB”, “from sklearn.naive_bayes import GaussianNB”, “from sklearn.tree import DecisionTreeClassifier”, and “from sklearn import svm”. Both textual data and metadata were then partitioned into training and testing sets. In particular, a training set was used to learn the behaviour of true news data and fake news data, while a testing set was used to generate the predictions of classification. The supervised machine learning algorithm in Algorithm 1 is summarizes the process for detecting fake news in textual data and metadata.

Algorithm 1. Supervised machine learning algorithm for fake news classification in textual data and metadata

```
Input: Textual data and metadata from tweets
Output: Classification of tweets (fake news/true news) and confusion matrix report
Step 1: Read dataset
Step 2: Split data into training and testing
for i=1 to n do
    Split data into training and testing
   x_train, x_test, y_train, y_test = train_test_split (x, y, test_size=0.3,
   random_state=0)
 end for
Step 3: Train the supervised machine learning classifier
if (textual data):
   tfidf_vectorizer = TfidfVectorizer()
   x_train_tfidf = tfidf_vectorizer.fit_transform(x_train)
   x_test_tfidf = tfidf_vectorizer.transform(x_test)
Step 4: Train supervised machine learning classifiers
if (textual data):
   def Naïve_Bayes()
      NB_TFIDF = MultinominalNB()
      NB_TFIDF.fit(x_train_tfidf, y_train)
      end Naïve_Bayes()
   def SVM()
      SVM_TFIDF = svm.SVC()
      SVM_TFIDF.fit(x_train_tfidf, y_train)
      end SVM()
    def Decision_Tree()
        DT_TFIDF = DecisionTreeClassifier()
        DT_TFIDF.fit(x_train_tfidf, y_train)
        end Decision_Tree()
if (metadata):
   def Naïve_Bayes()
      NB_model = GaussianNB()
      NB_model.fit(x_train, y_train)
      end Naïve_Bayes()
   def SVM()
      SVM_model = svm.SVC()
      SVM_model.fit(x_train, y_train)
      end SVM()
   def Decision_Tree()
      DT_model = DecisionTreeClassifier()
      DT_model.fit(x_train, y_train)
      end Decision_Tree()
Step 5: Test the supervised machine learning classifier
if (textual data):
   NB_predictions = NB_TFIDF.predict(x_test_tfidf)
   SVM_predictions = SVM_TFIDF.predict(x_test_tfidf)
   DT_predictions = DT_TFIDF.predict(x_test_tfidf)
if (metadata):
   NB_predictions = NB_model.predict(x_test)
   SVM_predictions = SVM_model.predict(x_test)
   DT_predictions = DT_model.predict(x_test)
Step 6: Generate confusion matrix report
NB_report = NB_confusion_matrix(y_test, NB_predictions)
SVM_report = SVM_confusion_matrix(y_test, SVM_predictions)
DT_report = DT_confusion_matrix(y_test, DT_predictions)
```

2.5. Feature extraction

Feature extraction identifies important features in a document by counting the significant score or weight of each word or phrase. In this study, the counting process involved calculating the frequency of each word and assigning a weight to them based on their occurrence. Features were ranked based on their importance score. Twitter fake news were detected based on the identified best features. Through the "classifier.feature_importances" from the Python library, frequencies were calculated, and a series of feature importance scores was stored. The features were sorted from the highest score (the most important) to the lowest score (the least important). Referring to Figure 4, the obtained results of feature importance revealed "number of followers" as the most important feature with the highest importance score of 0.287, which was followed by "tweet_count" (0.185), "source" (0.148), "verified" (0.143), "quote_count" (0.111), "no_retweet" (0.054), "no_likes" (0.042), and lastly, "no_following" (0.026).

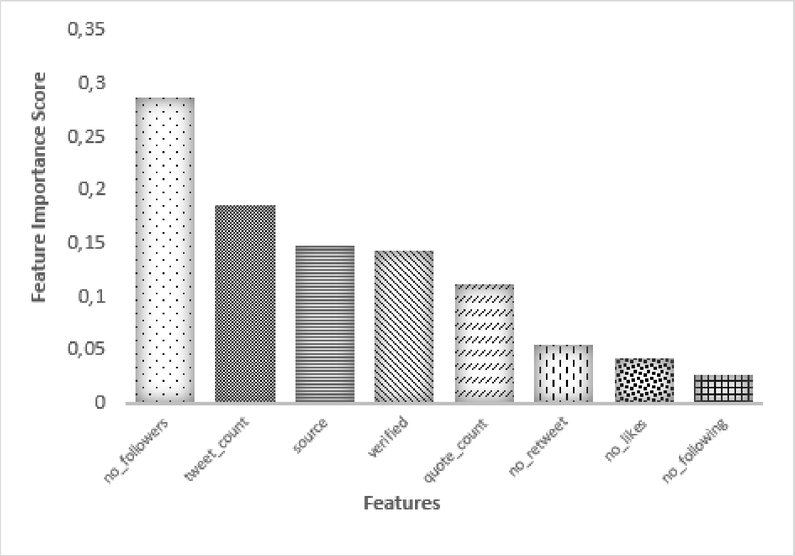


Figure 4. Results of feature importance for Twitter fake news and true news

2.6. Model evaluation

Performance metrics, specifically accuracy, precision, recall and F1-score, were used for the evaluation of the developed classifier models. Accordingly, these metrics are based on the concept of a confusion matrix, as illustrated in Table 1. The confusion matrix consists of several components with specific calculations. Firstly, true positive (TP) refers to the number of instances correctly predicted as positive. Secondly, true negative (TN) refers to the number of instances correctly predicted as negative. Thirdly, false positive (FP) refers to the number of instances inaccurately predicted as positive (but actually negative). Lastly, false negative (FN) refers to the number of instances inaccurately predicted as negative (but actually positive).

Table 1. Confusion matrix

		Actual class	
		Positive (P)	Negative (N)
Predicted class	Positive (P)	True positive (TP)	False positive (FP)
	Negative (N)	False negative (FN)	True negative (TN)

When it comes to machine learning classification model evaluation, accuracy is the most frequently used performance metric. It offers the most straightforward approach of calculating the proportion of correctly predicted instances.

$$Accuracy = \frac{TP+TN}{P+N} \tag{7}$$

Relying on accuracy alone for model evaluation is inadequate when highly skewed data distribution issues are involved. Addressing that, the current study included additional performance metrics, specifically precision, recall, and F1-score, to achieve a more optimal balance in data classification. As shown in (8), precision refers to the measurement of the accuracy or percentage of instances that the classifier correctly predicts as positive (out of the total instances predicted as positive).

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

As shown in (9), recall or also known as sensitivity measures the completeness or percentage of actual positive instances predicted by the classifier as positive. A high recall indicates that the model successfully identifies most of the positive cases, minimizing the risk of overlooking false negatives. It is particularly crucial in scenarios at which the inability to detect true positives could have significant implications, especially in the detection of fake news. Referring to (10), F1-score measures the harmonic mean between precision and recall, with the score of 1.0 indicates the perfect balancing score between the two metrics. The implementation of performance metrics in Python requires the use of “accuracy_score” and “classification report” functions.

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

$$F1 - score = \frac{2 \times precision \times recall}{precision \times recall} \tag{10}$$

3. RESULTS

A comparative analysis of the performance metrics for the machine learning classification models between the cases of textual data and metadata was conducted. The obtained results were intended to identify the most effective setting combination for Twitter fake news (and true news) detection. The results of accuracy for Naïve Bayes, decision tree, and SVM classifiers in the cases of textual data and metadata are tabulated in Table 2. Decision tree recorded the highest accuracy, with a 100% score for the case of textual data and a 94.54% score for the case of metadata.

Referring to Table 3, the results of precision, recall, and F1-score for Naïve Bayes, decision tree, and SVM classifiers regarding textual data-based classification tasks revealed additional insights on the performance of each model. Based on the obtained results for the case of textual data, the performance of Decision tree surpassed the performance of Naïve Bayes and SVM in terms of precision, recall, and F1-score for Twitter fake news and true news. Meanwhile, the results of precision, recall, and F1-score for

Naïve Bayes, decision tree, and SVM classifiers regarding metadata classification task are presented in Table 4. Decision tree recorded the highest F1-score for both cases of Twitter fake news and true news. As compared to Naïve Bayes and SVM, decision tree demonstrated a better balance of precision and recall. Meanwhile, SVM recorded the best precision (100%) and recall (100%) for Twitter fake news and true news within the context of the metadata classification task.

Table 2. Accuracy for cases of textual data and metadata

Machine learning	Accuracy (%)	
	Textual data	Metadata
Naïve Bayes	97.27	89.09
Decision tree	100	94.54
SVM	55.19	58.18

Table 3. Precision, recall and F1-score for textual data-based classification task

Machine learning	Fake news			True news		
	Precision (%)	Recall (%)	F1-score (%)	Precision (%)	Recall (%)	F1-score (%)
Naïve Bayes	100	94	97	95	100	98
Decision tree	100	100	100	100	100	100
SVM	0	0	0	55	100	71

Table 4. Precision, recall and F1-score for metadata classification task

Machine learning	Fake news			True news		
	Precision (%)	Recall (%)	F1-score (%)	Precision (%)	Recall (%)	F1-score (%)
Naïve Bayes	83	96	89	96	83	89
Decision tree	92	96	94	97	93	95
SVM	100	8	15	57	100	72

Overall, decision tree demonstrated good performance in classifying Twitter fake news and true news for both cases of textual data and metadata. With the aim to refine the model accuracy, the study further performed hyperparameter tuning on decision tree for metadata classification task. The refinement was executed using the gridsearchcv() function from the Python library. The results of the hyperparameter tuning are summarised in Table 5, which clearly revealed improved accuracy, suggesting better model performance in practice after the implementation hyperparameter tuning.

Table 5. Hyperparameter tuning on decision tree for metadata classification task

Machine learning	Accuracy before hyperparameter tuning (%)	Accuracy after hyperparameter tuning (%)
Decision tree	94.54	96.36

A crucial aspect of this study's methodology relies in its utilization of two types of data, namely textual and metadata. This is important in understanding the different characteristics that are inherent in each kind of data, highlighting the necessity for personalized models that are optimized for each type of data and, therefore, enhancing the overall performance of the proposed model. Another important feature in this study is the selection of the classification model, where the implementation of the fake news classifications relies on the three classifiers: Naïve Bayes, decision tree, and support vector machine. The three classifiers were chosen for their general ability to process high-dimensional data, including textual data and metadata features. The thorough evaluation of these classifiers aligns with the objective of this study, which is to construct a comprehensive and adaptable model for the accurate classification of fake news.

4. DISCUSSION AND FUTURE WORKS

This study has proposed a supervised machine learning algorithm that integrates both textual data and metadata features for the classification of fake news in social media posts, a method not explored prior. However, the methodology underpinning the proposed models necessitates further exploration. Firstly, the efficacy of machine learning models, particularly for fake news detection, substantially relies on data availability and quality. The current study employed Twitter API for data extraction in the domain of healthcare and politics to ensure accurate analysis. In particular, information like user ID, username, location, tweet counts, and date range were acquired. In order to ensure the model accuracy and efficacy, all collected data were then subjected to data preprocessing and cleaning considering the complexities and presence of noise in real-world data that may result in false predictions and erroneous conclusions.

The approach of data preprocessing and cleaning in this study involved converting text to lowercase, removing hyperlinks, punctuation, and whitespace, as well as performing hashtag removal and text lemmatization and stemming. Considering the significance of ensuring the accuracy of fake news detection model, it is recommended for future research to consider integrating other preprocessing techniques like n-gram analysis and properly handling misspelled words in tweets. Adding to that, the current study demonstrated that decision tree generally outperformed Naïve Bayes and SVM. Despite that, it is recommended for future research to incorporate another model, particularly deep-learning model architectures like convolutional neural networks or recurrent neural networks, for enhanced classification performance. The integration of machine learning and deep learning models potentially improves accuracy and presents a broader understanding of how the models deal with complex datasets.

Furthermore, the obtained tweets in this study were classified as true news or false news using the binary classification method, 0 or 1. However, this method might not adequately capture the linguistic variation present in fake news. News articles often convey a wide range of information, comprising elements that are both true and false as well as diverse opinions. Hence, this study envisions exploring the multi-label classification incorporating labelling of “true news”, “fake news” and “partially fake news” to improve comprehensive analysis of the characteristics of fake news, especially in health and political domains. This effort would increase the validity of the proposed model.

Another piece of future work could be done by investigating the other diversifications that contribute to the detection of fake news. While the proposed model concentrated on two types of data, textual data and metadata, there is potential for improvement by incorporating additional features, such as user profile features. Attributes such as age, registration time, account verification status, and follower count could enhance the performance of the proposed fake news detection model.

5. CONCLUSION

As compared to traditional printed news, news consumption via social media platforms is preferred following the immense popularity of social media. However, this shift has facilitated the rapid dissemination of fake news, adversely affects individuals and society. Addressing that, the current study proposed a supervised machine learning algorithm that can accurately detect fake news on social media, particularly Twitter. With that, Twitter data in the domain of healthcare and politics were obtained through the use of Twitter API. The obtained dataset was subsequently divided into textual data and metadata. In particular, textual data contained textual content, while metadata contained attributes like “no_follower”, “tweet_count”, “source”, “verified”, “quote_count”, “no_retweet”, “no_likes”, and “no_following”. Both textual data and metadata were subjected to data preprocessing and cleaning using techniques like stop word removal and text lemmatization and stemming, which was followed by data transformation (data vectorization) using BoW and TF-IDF. With these pre-processed datasets, the study proceeded to develop machine learning classifiers, which specifically involved Naïve Bayes, decision tree and SVM. The performance of these three proposed classifiers was evaluated in terms of accuracy, precision, recall, and F1-score for both cases of textual data and metadata. Based on the obtained results, decision tree demonstrated the highest accuracy for both cases of textual data and metadata, which clearly outperformed Naïve Bayes and SVM. Although SVM recorded the highest precision score for the case of metadata, decision tree demonstrated the best performance in terms of recall and F1-score. In conclusion, decision tree was identified as the best machine learning classifier model for Twitter fake news detection. Future research can explore other preprocessing techniques and integrate machine learning and deep learning methodologies to enhance the performance of the proposed fake news detection model.

ACKNOWLEDGEMENTS

The authors would like to thank the Universiti Teknologi MARA for funding the research project under MyRa Grant Scheme File No.: 600-RMC/GPM LPHD 5/3 (180/2021). The authors gratefully acknowledge the College of Computing, Informatics, and Mathematics at Universiti Teknologi MARA Cawangan Negeri Sembilan and Universiti Teknologi MARA Kampus Shah Alam for supporting the publication of this paper.

REFERENCES

[1] X. Zhang and A. A. Ghorbani, “An overview of online fake news: Characterization, detection, and discussion,” *Information Processing & Management*, vol. 57, no. 2, p. 102025, Mar. 2020, doi: 10.1016/j.ipm.2019.03.004.

[2] M. Nasery, O. Turel, and Y. Yuan, “Combating fake news on social media: a framework, review, and future opportunities,” *Communications of the Association for Information Systems*, vol. 53, no. 1, pp. 833–876, 2023, doi: 10.17705/1CAIS.05335.

[3] S. Aggarwal, T. Sinha, Y. Kukreti, and S. Shikhar, “Media bias detection and bias short term impact assessment,” *Array*, vol. 6, p. 100025, Jul. 2020, doi: 10.1016/j.array.2020.100025.

[4] M. Flintham, C. Karner, K. Bachour, H. Creswick, N. Gupta, and S. Moran, “Falling for fake news: investigating the consumption of news via social media,” in *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, New York, NY, USA: ACM, Apr. 2018, pp. 1–10. doi: 10.1145/3173574.3173950.

[5] J. Demuyakor and E. M. Opata, “Fake news on social media: predicting which media format influences fake news most on Facebook,” *Journal of Intelligent Communication*, vol. 2, no. 1, Jun. 2022, doi: 10.54963/jic.v2i1.56.

[6] S. Park, C. Fisher, T. Flew, and U. Dulleck, “Global mistrust in news: the impact of social media on trust,” *International Journal on Media Management*, vol. 22, no. 2, pp. 83–96, Apr. 2020, doi: 10.1080/14241277.2020.1799794.

[7] C. Wang and H. Huang, “When ‘Fake News’ becomes real: the consequences of false government denials in an authoritarian country,” *Comparative Political Studies*, vol. 54, no. 5, pp. 753–778, Apr. 2021, doi: 10.1177/0010414020957672.

[8] H. Alhazmi and H. N. Alhazmi, “Text mining in online social networks: a systematic review,” *IJCSNS International Journal of Computer Science and Network Security*, vol. 22, no. 3, pp. 396–404, 2022, doi: https://doi.org/10.22937/IJCSNS.2022.22.3.50.

[9] M. Höller, “The human component in social media and fake news: the performance of UK opinion leaders on Twitter during the Brexit campaign,” *European Journal of English Studies*, vol. 25, no. 1, pp. 80–95, Jan. 2021, doi: 10.1080/13825577.2021.1918842.

[10] G. Di Domenico, J. Sit, A. Ishizaka, and D. Nunan, “Fake news, social media and marketing: a systematic review,” *Journal of Business Research*, vol. 124, pp. 329–341, Jan. 2021, doi: 10.1016/j.jbusres.2020.11.037.

[11] F. Prieto-Ramos, J. Pei, and L. Cheng, “Institutional and news media denominations of COVID-19 and its causative virus: Between naming policies and naming politics,” *Discourse & Communication*, vol. 14, no. 6, pp. 635–652, Dec. 2020, doi: 10.1177/1750481320938467.

[12] J.-S. Shim, Y. Lee, and H. Ahn, “A link2vec-based fake news detection model using web search results,” *Expert Systems with Applications*, vol. 184, p. 115491, Dec. 2021, doi: 10.1016/j.eswa.2021.115491.

[13] B. Ahmed, G. Ali, A. Hussain, A. Baseer, and J. Ahmed, “Analysis of text feature extractors using deep learning on fake news,” *Engineering, Technology & Applied Science Research*, vol. 11, no. 2, pp. 7001–7005, Apr. 2021, doi: 10.48084/etasr.4069.

[14] F. Yang *et al.*, “XFake: explainable fake news detector with visualizations,” in *The World Wide Web Conference*, New York, NY, USA: ACM, May 2019, pp. 3600–3604. doi: 10.1145/3308558.3314119.

[15] B. Xie, X. Ma, J. Wu, J. Yang, and H. Fan, “Knowledge graph enhanced heterogeneous graph neural network for fake news detection,” *IEEE Transactions on Consumer Electronics*, vol. 70, no. 1, pp. 2826–2837, Feb. 2024, doi: 10.1109/TCE.2023.3324661.

[16] K. Shu, L. Cui, S. Wang, D. Lee, and H. Liu, “dEFEND: explainable fake news detection,” in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, New York, NY, USA: ACM, Jul. 2019, pp. 395–405. doi: 10.1145/3292500.3330935.

[17] M.-Y. Chen, Y.-W. Lai, and J.-W. Lian, “Using deep learning models to detect fake news about COVID-19,” *ACM Transactions on Internet Technology*, vol. 23, no. 2, pp. 1–23, May 2023, doi: 10.1145/3533431.

[18] Z. A. Khan *et al.*, “Identifying hot topic trends in streaming text data using news sequential evolution model based on distributed representations,” *IEEE Access*, vol. 11, pp. 98787–98804, 2023, doi: 10.1109/ACCESS.2023.3312764.

[19] D. K. Sharma, B. Singh, and A. Garg, “An ensemble model for detecting sarcasm on social media,” in *2022 9th International Conference on Computing for Sustainable Global Development (INDIACom)*, IEEE, Mar. 2022, pp. 743–748. doi: 10.23919/INDIACom54597.2022.9763115.

[20] I. Taleb, M. A. Serhani, and R. Dssouli, “Big data quality assessment model for unstructured data,” in *2018 International Conference on Innovations in Information Technology (IIT)*, IEEE, Nov. 2018, pp. 69–74. doi: 10.1109/INNOVATIONS.2018.8605945.

[21] W. Han and V. Mehta, “Fake news detection in social networks using machine learning and deep learning: performance evaluation,” in *2019 IEEE International Conference on Industrial Internet (ICII)*, IEEE, Nov. 2019, pp. 375–380. doi: 10.1109/ICII.2019.00070.

[22] K. Shu, A. Sliva, S. Wang, J. Tang, and H. Liu, “Fake news detection on social media: a data mining perspective,” *ACM SIGKDD Explorations Newsletter*, vol. 19, no. 1, pp. 22–36, Sep. 2017, doi: 10.1145/3137597.3137600.

[23] J. Shaikh and R. Patil, “Fake news detection using machine learning,” in *2020 IEEE International Symposium on Sustainable Energy, Signal Processing and Cyber Security (iSSSC)*, IEEE, Dec. 2020, pp. 1–5. doi: 10.1109/iSSSC50941.2020.9358890.

[24] I. Jamaleddyn, R. El ayachi, and M. Biniz, “An improved approach to Arabic news classification based on hyperparameter tuning of machine learning algorithms,” *Journal of Engineering Research*, vol. 11, no. 2, p. 100061, Jun. 2023, doi: 10.1016/j.jer.2023.100061.

[25] A. K. Ali and A. M. Abdullah, “Fake accounts detection on social media using stack ensemble system,” *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 12, no. 3, p. 3013, Jun. 2022, doi: 10.11591/ijece.v12i3.pp3013-3022.

[26] F. Benabbou, H. Boukhouima, and N. Sael, “Fake accounts detection system based on bidirectional gated recurrent unit neural network,” *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 12, no. 3, p. 3129, Jun. 2022, doi: 10.11591/ijece.v12i3.pp3129-3137.

[27] K. Khaled, A. ElKorany, and C. A. Ezzat, “Enhancing prediction of user stance for social networks rumors,”*International Journal of Electrical and Computer Engineering (IJECE)*, vol. 13, no. 6, p. 6609, Dec. 2023, doi: 10.11591/ijece.v13i6.pp6609-6619.

[28] I. Ahmad, M. Yousaf, S. Yousaf, and M. O. Ahmad, “Fake news detection using machine learning ensemble methods,” *Complexity*, vol. 2020, pp. 1–11, Oct. 2020, doi: 10.1155/2020/8885861.

BIOGRAPHIES OF AUTHORS

Nur Atiqah Sia Abdullah is Associate Professor at the Computing Sciences Studies, College of Computing, Informatics and Mathematics, Universiti Teknologi MARA, Shah Alam, Selangor, Malaysia. She currently serves in an administrative post as the Deputy Dean of Industrial, Community, and Alumni Network. Throughout her academic career, she has supervised and co-supervised both master's and Ph.D. students. Her substantial contributions to academia encompass over 50 publications, comprising both indexed and non-indexed publications. Her work has garnered recognition with a Scopus H-index of 7 and over 100 citations. Her research interests include software metrics, social media intelligence, and data visualization. She can be contacted at email: atiqah684@uitm.edu.my.

Nur Ida Aniza Rusli is a Senior Lecturer at the Computing Sciences Studies, College of Computing, Informatics and Mathematics, Universiti Teknologi MARA, Cawangan Negeri Sembilan, Kampus Kuala Pilah, Malaysia. She received the B.Sc. degree in information system from the Universiti Tun Hussein Onn Malaysia, the M.Sc. degree and Ph.D. degree in computer science from Universiti Teknologi MARA, Malaysia. Her research interests include software metrics, data mining and machine learning. She can be contacted at email: idaaniza@uitm.edu.my.

Nurshaheeda Shazlin Yuslee is a holder of a BSc and Master’s degree in computer science from the College of Computing, Informatics and Mathematics, Universiti Teknologi MARA, Shah Alam, Selangor, Malaysia. Her research interest is in computer science, with a particular focus on software development and machine learning. She can be contacted at email: shazleenys0512@gmail.com.