

# The Application of Random Forest to the Classification of Fake News

*Najwan thair ali*<sup>1\*</sup>, *Karrar Falih Hassan*<sup>2</sup>, *Muataz Najim Abdullah*<sup>3</sup>, and *Zainab Salam Al-hchimy*<sup>4</sup>

<sup>1</sup>Computer Techniques Engineering Department College of Engineering and Technologies Al-Mustaqbal University

<sup>2</sup>Department of Computer and Communication Engineering Islamic University Wardanieh, Lebanon

<sup>3</sup> Department of Computer and Communication Engineering Islamic University Wardanieh, Lebanon

<sup>4</sup>University of Alkafeel, Najaf, Iraq

**Abstract.** Fake News is one of the most widespread phenomenon with significant consequences on our daily life, particularly in the political realm. Due to the increasing use of the internet and social media, it is now much simpler to propagate false information. Therefore, the identification of elusive news is a significant issue that must be addressed, mostly due to obstacles such as the limited number of benchmark datasets and the volume of news produced per second. This study suggested using comparative data analysis based on random forest machine learning algorithm to identify bogus 4news. In this study the size of the whole dataset is 20.761 fake news record, whereas the size of it is 4.345 records. The first step in the data preparation process is to remove any unnecessary special characters, numbers, English letters, and whitespace. Before implementing the proposed classification algorithms, the most prevalent feature extraction approach (TF-IDF) is used. The data indicate that the highest level of accuracy attained was 88.24%.

## 1 Introduction

The phrase "Fake News" has entered the lexicon of the industry, often referring to intentionally deceptive and inaccurate stories published with the primary goal of monetizing the site's traffic. Nowadays, most efforts are directed on developing a model that can reliably predict whether a given item should be labeled as true or false. Fake news has started to pose a danger to trustworthy, credible news in recent years. It is now simple to create realistic but false news items that can misinform the public because to recent advancements in artificial intelligence (AI) generation technology. This research suggests an SVM model that can classify articles with 89 percent accuracy based simply on the title and 98 percent accuracy based on the title and the first 1000 characters in order to help distinguish between authentic and false news reports. Systems that monitor social media or other platforms fast may be able to spot dubious articles thanks to this high level of accuracy.

Determining and objectively defining what makes the new site "genuine" is essential. Inaccurate information has detrimental impacts, such as spreading claims that President Trump wants to remove the First Amendment in order to slaughter crowds in India through the WhatsApp app or that Hillary Clinton has a foreign kid. Nowadays, we accept at face value everything we read online or on social media without investigating further to see whether or not the material is accurate. Identifying false from genuine news takes considerable time and effort on the part of humans, who must examine each story's cited sources and verify their veracity to reach a conclusion [1].

---

\* Corresponding author: [najwan.thaeir@mustaqbal-college.edu.iq](mailto:najwan.thaeir@mustaqbal-college.edu.iq)

As a result, there is a growing need for a system that can automatically and intelligently identify disingenuous articles. Thus, the global community of researchers devotes a great deal of time and energy to the problem of identifying false news. In Singapore, Google and Facebook have opposed the introduction of additional regulations to combat fake news, arguing that the current rules are enough and that training people to recognize false news from genuine news is a more successful strategy [2]. Fake news persists in various forms every day despite ongoing efforts from the existing society, its people, its technology, and its processes. Researchers have high hopes that systems built using Artificial Intelligence (AI), Natural Language Processing (NLP), and machine learning would be able to automatically spot bogus news. However, uncovering fake news is a challenging task since it requires models to summarize news and compare them with actual news in order to identify them as false [3].

The objective of this study is to detect fraudulent news on social media using Decision tree and Random forest. The website Kaggle's text-based dataset is used to identify fake news. Comparison of the accuracy of the two methods is performed. It is possible to use a dataset to identify fake news, but it is important to note that doing so is a complex task that requires a combination of techniques and approaches. One approach to identifying fake news is to use machine learning algorithms to analyze the characteristics of the news articles in the dataset and identify patterns or features that are indicative of fake news. This can involve techniques such as natural language processing (NLP) to analyze the language and style of the articles, or analysis of the sources of the articles to identify patterns of misinformation. Another approach is to use fact-checking techniques to verify the accuracy of the information in the articles. This can involve verifying the sources of the information, checking the veracity of quotes or statistics, or consulting with experts in relevant fields.

It is also important to consider the context in which the news is being shared and the motivations of the people or organizations sharing it. This can involve analyzing the social media networks or other platforms where the news is being shared, as well as examining the political or ideological biases of the sources of the news. Overall, identifying fake news is a complex task that requires a multifaceted approach and careful evaluation of the information being shared. By comparing the data to known patterns of real news. If the data does not match these patterns, it may be flagged as potentially fake news. Additionally, if the data includes information that cannot be verified or is contradicted by other reliable sources, it may also be flagged as potentially fake.

## **2 Related Work**

### **2.1 Definition of Fake news**

**Definition of fake news** The credibility of information was defined by many words such as trustworthiness, believability, reliability, accuracy, fairness, objectivity, and other with the same concepts and definitions. There are several research use the machine learning approach to calculate the credibility of tweet's message. Fake news is the contents that claim people to believe with the falsification, sometime it is the sensitive messages. When the messages were received, they will rapidly dispersed it to other. The dissemination of fake news in today's digital world has effected beyond a specific group. Mixing both believable and unbelievable information on social media has made the confusion of truth. That is the truth will be hardly classified. However, the appearance of fake news causes great threat on the safety of people's lives and property. There are misinformation (the distributor believes there are true) or disinformation (the distributor knows it is not fact but he intentionally hoax) in fake news proliferation. For example, In the Royal Cremation Ceremony of His Majesty King Bhumibol Adulyadej of Thailand, many fraud with the serving areas for observing and misinformation were broadcasted. The agenda of the event both the main ceremony and each province are ambiguous news. There caused confusion to the people who wanted to join the event. Many research use the sentiment analysis and emotion classification to identify the fake news but it depend on the language's content.

Most of the prior work on this topic has focused on the use of machine learning and deep learning techniques to identify forgeries. Using a dataset gathered from Signal Media and a list of sources from the Open Sources, Shlock Gilda [4] investigated the applications linked to NLP approaches to identify "fake news," or false news items derived from unreliable sources, in 2017. Collectively, they used TF-IDF on a dataset of about 11000 articles to identify bigrams and probabilistic context-free grammar (PCFG). Use SVMs, Random Forests, Gradient Boosting, Stochastic Gradient Descent, and Bounded Decision Trees to evaluate the dataset's classification accuracy. Research found that a Stochastic Gradient Descent model fed with TF-IDF on bigrams could accurately detect unreliable sources with an accuracy of 77.2%.

Deep learning algorithms were created in 2018 by Arjun Roy, Kingshuk Basak [5]. and Asif Ekbal to identify disingenuous media and place it in specific, well-defined buckets. They started by creating models using convolutional neural networks (CNNs) and long short-term memory (LSTM) networks. Additionally, MLP was used as the final classifier after the representations obtained from the two models were input into it. In addition, several of their experiments on the benchmark dataset show positive results, with an overall accuracy of 44.87%, which is higher than that of state-of-the-art models.

In 2019, ML and NLP viewpoints were shown by Arvinder Pal Singh Bali and Maxson Fernandes [6], The estimations were made using a fresh set of features taken from the datasets' contents and headlines, and they were applied to three industry-standard datasets. In addition, seven ML systems' results were evaluated in terms of F1 scores and precisions. In addition, Gradient Boosting has a higher accuracy than classifiers at 88%.

A model of Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) recurrent NN architecture was proposed in 2019 by Ahlem Drif, Zineb Ferhat Hamida [7], taking advantage of the coarse-grained local features taken from CNN and the long-distance dependencies learned through LSTM, with the articles news of fake news serving as the dataset and the size of the dataset being 50,000. (20,761). The findings demonstrate that CNN-LSTM achieves the highest accuracy (0.725) when compared to the CNN and SVM baselines.

### 3 Applied Model

Selecting and preparing the right fake news dataset from kaggle.com is the first stage. After dividing the dataset using cross-validation. The TF-IDF algorithm is then used to extract word characteristics. As seen in Figure 1, the next stage is to classify the dataset using (Random Forest) classifiers and assess model performance using multiple metrics such as (accuracy, recall, and precision)

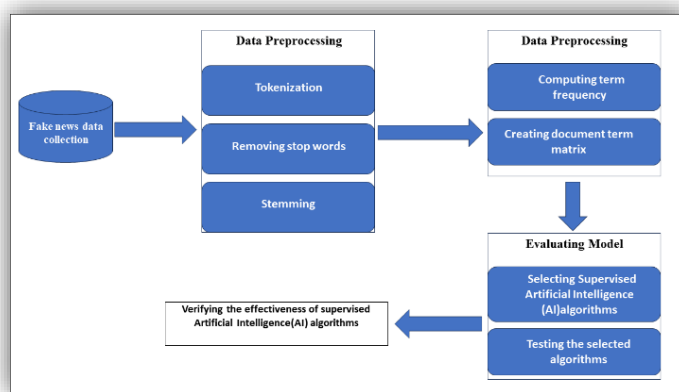


Fig.1. APPLICABLE MODEL.

## 4 Fake news kaggle Dataset

Multiple sources, including news agency websites and social media platforms such as Twitter, Facebook, and Instagram, may be mined for datasets pertaining to false news. Nevertheless, manually distinguishing the different types of news is difficult. Therefore, a specialist in examining the claims, facts, and context of reputable sources is necessary. News data may be gathered in a variety of methods, including by skilled journalists, fact-checking websites, and crowd source workers. There are currently no concurrent benchmark datasets for false news detection challenges .

This work uses the dataset (.CSV file of false news stories) acquired from kaggle.com. This dataset contains around 16,088 entries from different Internet-based publications, and its properties include (URLs, headings, body and label). After preprocessing, the size of the dataset increased to 20,761 records. This information is separated into two groups. In this study, just two characteristics (text and label) are utilized to identify false news classifiers.

It is possible to use a dataset to identify fake news, but it is important to note that doing so is a complex task that requires a combination of techniques and approaches. One approach to identifying fake news is to use machine learning algorithms to analyze the characteristics of the news articles in the dataset and identify patterns or features that are indicative of fake news. This can involve techniques such as natural language processing (NLP) to analyze the language and style of the articles, or analysis of the sources of the articles to identify patterns of misinformation.

Another approach is to use fact-checking techniques to verify the accuracy of the information in the articles. This can involve verifying the sources of the information, checking the veracity of quotes or statistics, or consulting with experts in relevant fields. It is also important to consider the context in which the news is being shared and the motivations of the people or organizations sharing it. This can involve analyzing the social media networks or other platforms where the news is being shared, as well as examining the political or ideological biases of the sources of the news.

Overall, identifying fake news is a complex task that requires a multifaceted approach and careful evaluation of the information being shared. by comparing the data to known patterns of real news. If the data does not match these patterns, it may be flagged as potentially fake news. Additionally, if the data includes information that cannot be verified or is contradicted by other reliable sources, it may also be flagged as potentially fake by analyzing patterns and trends in the data. This could include looking for inconsistencies or anomalies in the data, or comparing the data to known sources to verify its accuracy. The analysis could also involve using machine learning algorithms to identify patterns or trends that may indicate fake news. Ultimately, the goal of measuring the dataset is to identify and remove fake news from circulation in order to promote reliable, accurate information.

To identify fake news using a measured dataset in Pycharm, you can use a variety of machine learning techniques. One option is to use a classification algorithm, such as a decision tree or a support vector machine, to classify news articles as either real or fake based on the features of the article. You can also use natural language processing techniques to analyze the language used in the article and identify patterns that are more common in fake news articles.

To begin, you will need to gather and pre-process your data by selecting a dataset of news articles and labeling them as either real or fake. You can then use Pycharm to create a machine learning model using the selected algorithm and train it on the labeled data. After the model has been trained, you can then use it to classify new news articles as real or fake based on their features.

It is important to note that accurately identifying fake news can be a difficult task, as the characteristics of fake news can vary widely. It may be necessary to continuously update and refine your model as new fake news articles emerge. Additionally, it is important to consider the ethical implications of using artificial intelligence to identify fake news, as the decisions made by the model may have significant consequences for the dissemination of information. To identify fake news with code in PyCharm, the first step would be to import the necessary libraries and load the dataset. This can be done using the following code.

```
import pandas as pd
import numpy as np

# Load the dataset
df = pd.read_csv('fake_news_dataset.csv')
```

**Fig. 2.** Program execution image.

Next, we will need to preprocess the data by cleaning it and removing any irrelevant columns, This can be done using the following code:

Once the data has been cleaned, we can then start building the model to identify fake news. There are several different machine learning algorithms that can be used for this task, such as decision trees, support vector machines, and neural networks. To build the model, we will need to split the dataset into training and test sets, and then fit the model to the training data. This can be done using the following code.

```
# Predict the label of a new article
article_features = [1, 2, 3, 4, 5] # Replace with actual article features

prediction = model.predict([article_features])
print(prediction) # Will print either 'fake' or 'real'
```

**Fig. 3.** Program execution image-part2

Once the model has been trained, we can then use it to predict whether a given news article is fake or not. To do this, we will need to pass the article's features to the model and then use the model's prediction function to get the predicted label. This can be done using the following code:

```
# Evaluate the model's accuracy
from sklearn.metrics import accuracy_score

y_pred = model.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
print(f'Model accuracy: {accuracy}')
```

**Fig. 4.** Program execution image-part3

Finally, we can use the model's accuracy to evaluate its performance on the test set. This can be done using the following code:

```
# Evaluate the model's accuracy
from sklearn.metrics import accuracy_score

y_pred = model.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
print(f'Model accuracy: {accuracy}')
```

**Fig. 5.** .Program execution image-part3

### 4.1 Preprocessing step

The data must go through a series of filters and cleaning procedures before it can be represented and features extracted using TF-IDF. These procedures include getting rid of stop words, punctuation, case-sensitive characters, the special character, numerals, and white space. As a result of this process, the dataset will be much smaller, with just the useful information remaining [2].

### 4.2 Feature selection

To manage natural language processing and develop a vector space model for extracted features, TF-IDF is a standard, well-defined approach. The article analyzes what this phrase means. The paper provides an analysis of the significance of the term. The frequency with which a word or phrase appears in a particular text lends credence to its significance. contrasted with the opposite of the term across the whole corpus of writings. The TF-IDF index is essentially linked to the word t that: [8]

A lower number if the document as a whole has fewer occurrences of the phrase t, or if the term t occurs in more than one document;

-A greater value when a phrase t appears several times in a limited set of texts;

- A smaller number if the phrase t appears in almost all of the papers. In formal terms:

Consider the set of documents to be  $D = d_1, d_2, \dots, d_n$ , and a term to be t inside D. For the purpose of computing the frequency-inverse document size, the steps are as follows:

$$TF - IDF(t, d, D) = TF(t, d) \times IDF(t, D).$$

### 4.3 Random Forest Algorithm

The term "bagging" or "bootstrap aggregation" refers to a method for decreasing the dispersion of a predicted function estimate. In classification, methods with high variance and low bias, like trees, function well with bagging. Significantly improving upon traditional bagging, random forests first construct a huge number of independently generated trees before averaging them out. Random Forest improved upon bagging by reducing the connection between trees without increasing the variance.

The performance of random forests is often comparable to that of boosts, and they are often easier to train and modify. As a consequence, random forests have become a popular technique used in many software applications [9].

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

Fig. 6. Random Forest Algorithm

## 5 Result and Discussion

### 5.1 Random Forest Algorithm

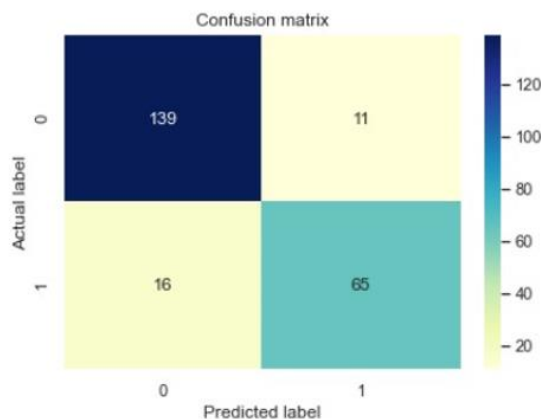


Fig. 7. Confusion Matrix Random Forest after preprocessing steps

Table 1: RESULTS OF RANDOM FOREST AFTER PREPROCESSING STEPS

Pointer	Result
Correctly classified as 1	837
Incorrectly classified as 1	73
Correctly classified as 0	927
Incorrectly classified as 0	239
Precision	92.02 %
Recall	79.53 %
Accuracy	24

From above table 1 result we can see that random forest give **Accuracy** of 88.24% and 79.53 % Recall value and 92.02 % precision score.

### 5.2 Decision Tree

The J48 method is widely used as a classification algorithm. All study data must be either numerical or categorical in kind, as required by the C4.5 algorithm upon which it is built. As a result, we won't be looking at any data with a continuous time scale. J48 makes use of two distinct methods of trimming. Sub tree replacement is the first technique, and it indicates the potential of swapping out individual leaves of a decision tree in order to reduce the total number of tests along the convincing route. Sub tree raising often has a little effect on decision tree models. The value of a given option is notoriously difficult to estimate, yet it may be prudent to disable it if doing so would shorten the induction procedure's execution time and save the computing resources required to raise the sub tree.

**Fig.8.** Decision Tree Algorithm

```
Algorithm 2: Decision Tree Algorithm.  
Input: Predefined classes  
Output: Built decision tree  
Num of features =17000  
Max -depth =2  
Begin  
Step 1: Create a root node for the tree  
Step 2: If all examples are positive, return leaf node  
‘positive.’  
Else if all examples are negative, return leaf node  
‘negative.’  
Step 3: Calculate the entropy of current state H(S)  
Step 4: For each attribute, calculate the entropy  
concerning the attribute ‘x’ denoted by H(S, x)  
Step 5: Select the attribute which has a maximum value of  
IG(S, x)  
Step 6: Remove the attribute that offers the highest IG  
from the set of attributes  
Step 7: Repeat until we run out of all attributes or the  
decision tree has all leaf nodes.  
End
```

### 5.3 Support Vector Machines (SVM)

Support Vector Machines (SVM) and Random Forests are both machine learning algorithms that can be used to classify items into categories. Both algorithms can be used to classify fake news in the context of machine learning. However, they have some differences that may make one more suitable for a particular task than the other.

One key difference between SVM and Random Forests is the way they make predictions. SVMs are based on the idea of finding a hyper plane in a high-dimensional space that maximally separates different classes. In contrast, Random Forests make predictions by combining the predictions of multiple decision trees, each of which is trained on a different subset of the data.

Another difference is the level of interpretability of the models. SVMs can be difficult to interpret because they are based on complex mathematical concepts, whereas Random Forests are more transparent and easy to understand because they are based on simple decision rules.

In terms of performance, SVM and Random Forests can both achieve good results on a wide range of tasks. However, Random Forests tend to be more robust and less prone to over fitting than SVMs, especially when dealing with large and complex datasets.

In terms of implementation, both SVM and Random Forests are available in popular machine learning libraries such as scikit-learn, which can be used in PyCharm or any other Python development environment. To compare the performance of SVM and Random Forests for detecting fake news in PyCharm, you would need to follow these steps:

- Collect a dataset of fake news articles and real news articles. Make sure the dataset is balanced, meaning it contains roughly the same number of fake and real news articles.
- Preprocess the data by converting the text of the articles into numerical features that can be used as input to the machine learning algorithms. This can be done using techniques such as tokenization, stemming, and factorization.
- Split the dataset into training and test sets. Use the training set to train the SVM and Random Forest models.
- Evaluate the performance of the SVM and Random Forest models on the test set using metrics such as accuracy, precision, and recall.

## 6 Evaluation metrics

Here, we employ the most used statistic for measuring false news detection (the Confusion Matrix). Considering this as a classification problem allows us to define the confusion matrix's attributes as follows:

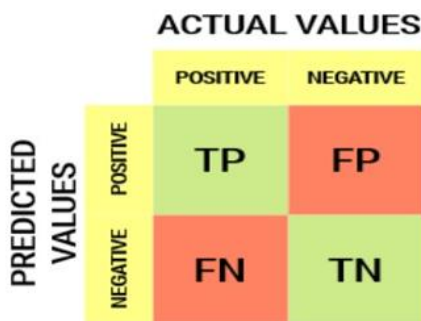
$$accuracy = \frac{tp + tn}{tp + tn + fp + fn} \quad (1)$$

$$precision = \frac{tp}{tp + fp} \quad (2)$$

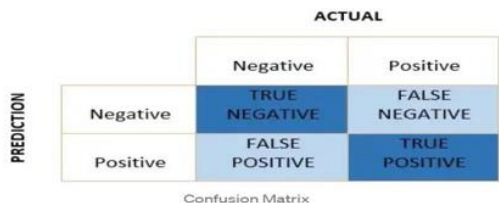
$$recall = tp / tp + fn \tag{3}$$

**Table 2 .** Result of random forest after preprocessing.

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



**Fig. 9** Confusion Matrix structure.



**Fig. 10.** Confusion Matrix.

These metrics are often used in the sequence of machine learning algorithms since they let assessing a classifier's efficiency from a variety of estimations. In particular, the precision measure that stands for the similarity between fake news predictions and actual false news. Precision measures the proportion of false stories that have been identified, providing a solution to the critical problem of false news labeling. However, the dataset of fake news is often biased; high accuracy may be achieved by producing fewer optimistic predictions; hence, recall is used to evaluate sensitivity, or the proportion of the annotated fraudulent articles projected as fake. In terms of Recall, Precision, and Accuracy, it is important to remember that greater numbers indicate better performance [10].

### 6.1 Decision Tree Result

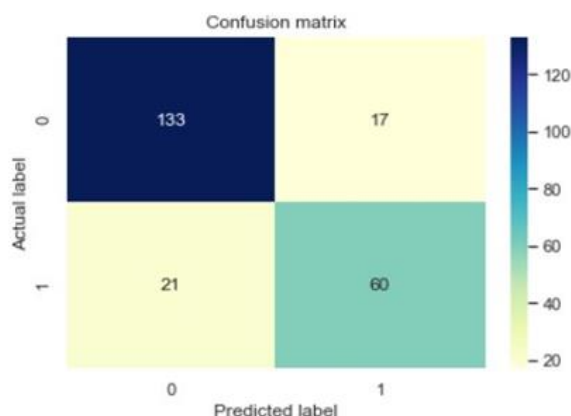


Fig. 11. Confusion Matrix Decision Tree after preprocessing steps.

Table 3. RESULTS OF DT AFTER PREPROCESSING STEPS

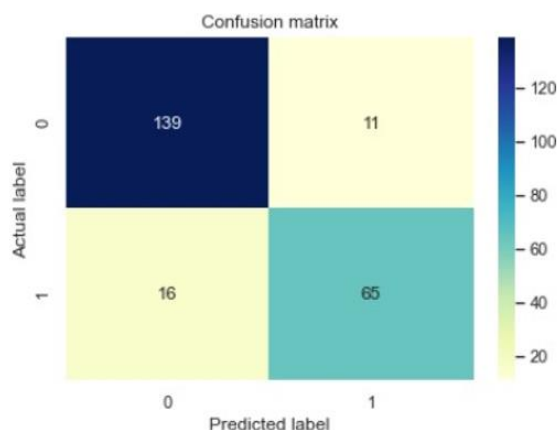
Pointer	Result
Correctly classified as 1	959
Incorrectly classified as 1	109
Correctly classified as 0	891
Incorrectly classified as 0	117
Precision	89.10%
Recall	88.39%
Accuracy	89.11%

### 6.2 SVM algorithm result

Table 4. Svm Result

	Precision	Recall	F-score	Accuracy
SVM	99.80%	99.90%	99.80%	99.90%

Fig. 12. Confusion Matrix SVM after preprocessing steps.



## 6 CONCLUSION

In this research, This issue of false news has been given a lot of coverage. New data and studies show that around 62% of individuals who share fake news on social media in the US are adults. In this study, we used the TF-IDF features extraction method to provide a detection model for disinformation. In addition, we use not one but two unique ML algorithms. In the instance when the decision tree classifier is used, the realized model has reached its maximum accuracy. The best possible result was 88.24% accuracy. The outcome is superior to the mentioned related work, hence using this technique improves classification precision. The above analysis leads us to the following conclusions:

The decision tree outperforms a random forest in terms of the classification accuracy of the fake news dataset. Second, unlike decision trees, the outputs of a random forest are not predetermined; rather, the optimal result is determined by a majority vote. This makes random forests better suited to huge datasets. Improved outcomes may be achieved by the use of our dataset in the preparation processes. The categorization accuracy improved significantly after implementing these changes. The accuracy of the categorization in this study is also heavily influenced by the kind of dataset obtained (Fake news articles of dataset).

Machine learning techniques successfully identify fake news. Twenty-two characteristics are used to characterize the data taken from Twitter for the experiment. This data suggests that machine learning techniques, such as the support vector machine, excel in identifying instances of fake news. It's possible that this doesn't cover the whole range of what's going on in the realm of actual news.

However, there is sufficient data to suggest that, at least in some contexts, detecting fake news is not very difficult. It's also hard to determine with certainty how much this experiment's findings can be transferred to actual news events. Going forward, we'd like to collect data from a wider range of sources and try to apply our approach to a wider variety of problems.

## References

1. V. D. Lakshmi and C. S. Kumari, "Detection of Fake News using Machine Learning Models," *Int. J. Comput. Appl.*, vol. 183, no. 47, pp. 22–27, 2022, doi: 10.5120/ijca2022921874.
2. T. A. Wotafi and B. N. Dhannoon, "Improving Prediction of Arabic Fake News Using Fuzzy Logic and Modified Random Forest Model," *Karbala Int. J. Mod. Sci.*, vol. 8, no. 3, pp. 477–485, 2022, doi: 10.33640/2405-609X.3241.
3. R. Jehad and S. A. Yousif, "Fake News Classification Using Random Forest and Decision Tree (J48)," *Al-Nahrain J. Sci.*, vol. 23, no. 4, pp. 49–55, 2020, doi: 10.22401/anjs.23.4.09.
4. S. B. S. Mugdha, S. M. Ferdous, and A. Fahmin, "Evaluating Machine Learning Algorithms for Bengali Fake News Detection," *ICCIT 2020 - 23rd Int. Conf. Comput. Inf. Technol. Proc.*, 2020, doi: 10.1109/ICCIT51783.2020.9392662.

5. T. Blanke and T. Venturini, "A network view on reliability: using machine learning to understand how we assess news websites," *J. Comput. Soc. Sci.*, vol. 5, no. 1, pp. 69–88, 2022, doi: 10.1007/s42001-021-00116-w.
6. Abedalla, A. Al-Sadi, and M. Abdullah, "A closer look at fake news detection: A deep learning perspective," *ACM Int. Conf. Proceeding Ser.*, no. October, pp. 24–28, 2019, doi: 10.1145/3369114.3369149.
7. C. M. M. Kotteti, X. Dong, N. Li, and L. Qian, "Fake news detection enhancement with data imputation," *Proc. - IEEE 16th Int. Conf. Dependable, Auton. Secur. Comput. IEEE 16th Int. Conf. Pervasive Intell. Comput. IEEE 4th Int. Conf. Big Data Intell. Comput. IEEE* 3, pp. 193–199, 2018, doi: 10.1109/DASC/PiCom/DataCom/CyberSciTec.2018.00042.
8. M. Fayaz, A. Khan, M. Bilal, and S. U. Khan, "Machine learning for fake news classification with optimal feature selection," *Soft Comput.*, vol. 26, no. 16, pp. 7763–7771, 2022, doi: 10.1007/s00500-022-06773-x.
9. Y. Yu, L. Wang, H. Huang, and W. Yang, "An improved random forest algorithm," *J. Phys. Conf. Ser.*, vol. 1646, no. 1, 2020, doi: 10.1088/1742-6596/1646/1/012070.
10. C. Whitehouse, T. Weyde, P. Madhyastha, and N. Komninos, "Evaluation of Fake News Detection with Knowledge-Enhanced Language Models," *Proc. Int. AAAI Conf. Web Soc. Media*, vol. 16, pp. 1425–1429, 2022, doi: 10.1609/icwsm.v16i1.19400.