Journal of Computer and Communications, 2022, 10, 1-21
https://www.scirp.org/journal/jcc
ISSN Online: 2327-5227
ISSN Print: 2327-5219

Performance Evaluation of Multiple Classifiers for Predicting Fake News

Arzina Tasnim[1], Md. Saiduzzaman[1], Mohammad Arafat Rahman[1], Jesmin Akhter[2], Abu Sayed Md. Mostafizur Rahaman[3]

[1]Department of Information Systems Security, Bangladesh University of Professionals, Dhaka, Bangladesh
[2]Institute of Information Technology, Jahangirnagar University, Dhaka, Bangladesh
[3]Department of Computer Science and Engineering, Jahangirnagar University, Dhaka, Bangladesh
Email: asmmr@juniv.edu

How to cite this paper: Tasnim, A., Saiduzzaman, Md., Rahman, M.A., Akhter, J. and Rahaman, A.S.Md.M. (2022) Performance Evaluation of Multiple Classifiers for Predicting Fake News. *Journal of Computer and Communications*, **10**, 1-21.
https://doi.org/10.4236/jcc.2022.109001

Received: July 19, 2022
Accepted: September 5, 2022
Published: September 8, 2022

Abstract

The rise of fake news on social media has had a detrimental effect on society. Numerous performance evaluations on classifiers that can detect fake news have previously been undertaken by researchers in this area. To assess their performance, we used 14 different classifiers in this study. Secondly, we looked at how soft voting and hard voting classifiers performed in a mixture of distinct individual classifiers. Finally, heuristics are used to create 9 models of stacking classifiers. The F1 score, prediction, recall, and accuracy have all been used to assess performance. Models 6 and 7 achieved the best accuracy of 96.13 while having a larger computational complexity. For benchmarking purposes, other individual classifiers are also tested.

Keywords

Fake News, Machine Learning, TF-IDF, Classifier, Estimator, F1 Score, Recall, Precision, Voting Classifiers, Stacking Classifier, Soft Voting, Hard Voting

1. Introduction

In the present era, the internet is considered the prime source of information where anyone can share their knowledge. This sharing shall not always bring us blessings due to the abundance of fake news on the internet. The rise of fake news on social media has had a negative impact on society.

Therefore, numerous studies based on supervised and unsupervised machine learning approaches have been offered to address the issue of recognizing fake news or anonymous information. Those strategies were designed to classify internet news and social media articles. It provides a framework for identifying fake and true news, assessing its F1 score, precision, recall, and accuracy to confirm its legitimacy.

Previous research has focused on developing models using supervised machine learning (ML) or a combination of NLP techniques, and suggesting outcomes based on the accuracy of the highest performing individual classifiers, as stated below. Previous exercises were performed with a limited number of classifiers, whereas our proposed paper demonstrates not only the performance evaluation of the highest number of classifiers (14 classifiers) but also a diverse evaluation has been demonstrated with the accumulation of voting and stacking classifiers, making our work truly unique and resilient.

Several previous studies of this nature have been conducted, and scholars have contributed considerable knowledge to the field. A handful of them is told in the following paragraphs. Khanam *et al.* [1] propose a method for creating a model that can assess an article's inventiveness based on its words, phrases, sources, and titles. On a manually classified dataset, they employed supervised machine learning methods. They propose using sci-kit-learn tools for feature extraction and choosing the best-fit features to get the highest precision, with feature selection methods based on confusion matrix results. The main goal of this research, according to Smitha *et al.* [2], is to develop a model that can accurately classify news as false or real by combining three methods of NLP text vectorization named count vector, TF-IDF, and word embedding and applying them to different ML classifiers, and to offer a model for detecting fake news. The SVM Linear classification algorithm with TF-IDF feature extraction yields the highest accuracy of 94%. Though Neural Networks have similar accuracy, SVM is given higher priority than Neural Networks due to its complexity and longtime consumption. Sentimental Analysis using Deep Learning could be introduced as a future improvement area, with datasets being built from many sources with a big number of articles. Vasu Agarwal *et al.* [3] chose SVM and logistic regression above five other classifiers: Nave Bayes, Logistic Regression, Linear SVM, Stochastic Gradient, and Random Forest. GridSearchCV techniques were used to tune the parameters for fake news classification on these candidates, with SVM scoring 61% and logistic regression scoring 52%. The key constraint of this study is the erratic nature of the data, as the projected model can also be anomalies. To ensure deep feature extraction and fine-tuned categorization, future enhancement areas could include POS tagging, word2vec, and topic modeling. Uma Sharma *et al.* [4] use binary classifiers in AI, NLP, and machine learning. They divided the explanation into three parts: static, dynamic, and URL search. The static section deals with the ML classifier, while the dynamic part accepts the user's keyword or text and checks the authenticity of the URL input. At the Work Level and Ngram-Level, this model was developed using Vector Feature-Count Vector and TF-IDF vectors. The K-fold cross-validation technique was used to improve the model's effectiveness. The best model is logistic regression, which has a 65% accuracy. On top of that, grid search parameter optimization is applied, which improves the accuracy to 75%. Knut Hinkelmann *et al.* [5] investigated the lack of an efficient approach to distinguish bogus and true information due to the lack of corpora, using two publicly available datasets from the Kaggle dataset [6]. Here, passive-aggressive, Nave Bayes, and SVM ML classifiers were used. RapidMiner, a strong machine learning tool, was employed for data exploration and extraction. The study's findings suggested a Passive-Aggressive classifier with 93% accuracy, although Naive Bayes (83%) and SVM (84%) are much behind passive-aggressive. Bichitrananda Behera *et al.* [7] established an automated Biomedical Text Classification Process that covers the performance evaluation of all renowned classifiers on one platform. To improve the model's accuracy, a multidisciplinary medical data classification technique called supervised learning was used with three publicly available data sets. The purpose of this paper's future work is to increase the adaptability of deep learning-based models for integrating enormous datasets and make them an inevitable option for future study. Dr. S. Rama Krishna *et al.* [8] examined various research articles and discovered that the best methods for data pre-processing are Word Embedding, Tokenization, and Parts of Speech Tagging, and the best methods for feature extraction are TF-IDF and Count Vectorizer. The process of extracting a segment of important characteristics from data to improve classification performance is known as feature extraction. The efficiency of altered free text is tested using logistic regression and random forest classifier with 3-fold stratified cross-validation, as identified by Resham N. Waykole *et al.* [9]. They draw attention to evaluate the efficiency of the converted free text, we utilized logistic regression and a random forest classifier with 3-fold layered cross-validation. To generate more accurate findings, Dharmaraj R. Patil [10] employed generally used machine learning classifiers and constructed a multi-model false news detection system utilizing the majority voting technique. The findings of the evaluation demonstrate that the majority voting technique produced more accurate outcomes than the individual learning strategy. To assess the models' performance, Tao Jiang *et al.* [11] used five ML and three DL models. On the ISOT and KDnugget datasets, the suggested new stacking model obtained testing accuracy of 99.94% and 96.05%, respectively. Furthermore, when compared to baseline approaches, the proposed strategy outperforms others.

We propose to introduce a platform based on the background study mentioned above, where fourteen different classifiers will be utilized to train a dataset to distinguish fake or authentic news. Accuracy, precision, recall, and F1 score will all be used to evaluate performance in this model. We use the TF-IDF vectorizer as a feature extraction method to improve the accuracy of this classification model. In addition, we evaluated the performance of a variety of classifiers using individual analogies, computing with a different voting technique, and stacking distinct classifiers.

2. Methodology

Our fake news prediction model is illustrated in **Figure 1**. To begin, 14 distinct classifiers are separately trained and tested for accuracy, precision, recall, and F1 score. To evaluate the accuracy on soft and hard voting processes, selective classifiers are clustered to participate in the voting mechanism. Onwards, stacking classifiers are formed by mixing different classifiers using a heuristics technique, and the proposed model shows the expected significant accuracy gain. The dataset for this analogy was obtained from Kaggle and consisted of around 8000 news items. The datasets are then preprocessed in various ways, including Data Cleaning, Remove Punctuation, Tokenization, Remove Stopwords and Stemming and Lemmatization. To locate the necessary characteristics, feature extraction attribute is carried out. To execute the evaluation, the dataset was separated into two parts: training (80%) and testing (20%). We used the training dataset with 14 distinct classifiers to develop the classification model, and then that trained model was used to evaluate the performance with testing datasets using specific performance measurement properties such as precision, recall, F1 score, and accuracy.

The paper has the following contributions:

1) Individual classifier's performance evaluation

2) Performance evaluation of classifiers through a voting algorithm (Soft and Hard Voting)

3) Performance evaluation of classifiers through a Stacking algorithm

Performance evaluation of machine learning algorithms through voting classifiers is performed with a group classifier consisting of a cluster of four and eight members which is selected randomly. Various combinations are assessed to evaluate the performance through a stacking algorithm.

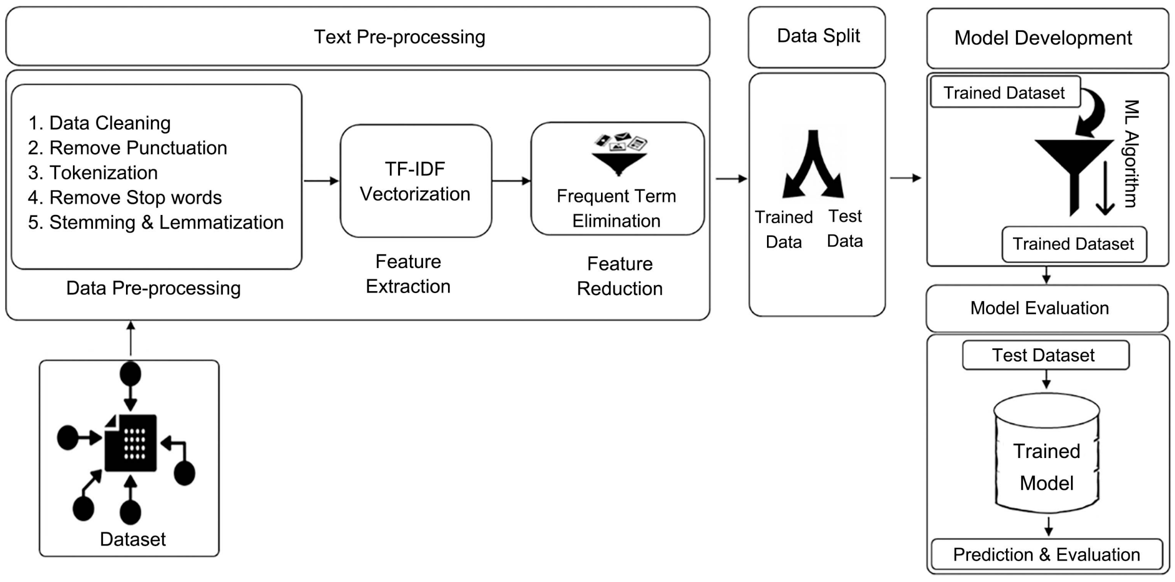


Figure 1. Fake news prediction model.

3. Implementation

3.1. Data Collection

The acquisition of a dataset is the initial step in this model. We will utilize the dataset for this project named news.csv. The shape of this dataset is 7796 × 4. The news is identified in the first column, the title and content are in the second and third columns, and the fourth column has labels indicating if the news is REAL or FAKE. The dataset occupies 29.2 MB of disk space. Following that, the acquired data will be used for testing, training, and for evaluation. As a result, our proposal for this project is to build a model to evaluate the performance of individual 14 classifiers to identify the efficient one.

3.2. Data Preprocessing/Data Cleaning

While working with text documents, text cleaning is a must. Structured, unstructured, and semi-structured text data are the three types of human-written text data. Structured data is well-defined data, whereas unstructured data is data that has lost its pattern, and semi-structured data is more structured than unstructured data. For further development, we need to convert our data set from structured to semi-structured format. To clean our dataset, we took several procedures. The following are some of them:

1) ***Remove punctuation***

All special characters, such as (!?., percent), have been removed. As a result, it is important to make a list of the punctuations that have been removed. The primary goal of removing punctuation from the dataset is to obtain a vector representation of words. The spacing within a sentence was used to achieve this.

2) ***Tokenization***

The process of tokenization is the breakdown of sentences into words. For the rest of the process, we'll require an individual meaningful entity. Every single thing is referred to as a token. This token can't contain any special characters or be a whole sentence.

3) ***Remove stopwords***

Stopwords are words that have little meaning in the dataset. These words, together with "or, am, is, and can," are employed often in the dataset to create sentences. In many circumstances, they aren't beneficial for text analysis, thus it's best to eliminate them from the text.

4) ***Stemming and Lemmatization***

Stemming is commonly used to remove the last few letters or characters from a word, but it also considers the context. By using a basic rule-based method, it can sometimes remove suffices such as ing, ly, s, and so on. However, it frequently has grammatical and typographical problems. Information may be updated to inform, but this does not change the meaning of the information. To address this problem, lemmatization has emerged as a rescuer, as it does not alter the sense of the sentences. The morphological examination of the words is taken into account during lemmatization, and the words are converted sensibly.

3.2.1. Data Extraction

Our categorization model is based on a text-based dataset. As a result, in order to work with this information, textual data must be converted to numeric form. However, extracting important information from a vast amount of text data is tough. As a result, computational text processing is required to extract information [12]. Feature extraction is the process of extracting a list of words from text input and converting them into a feature set that can be used by a classifier. It is a method of numerically representing textual data. To turn a simple text into features, various techniques such as Bag of Words (BOW), TF-IDF, and Word embeddings can be utilized. For the data processing in our suggested model, we have chosen TF-IDF over BOW. The reason of choosing TF-IDF has been shared below.

3.2.2. Bag of Words (BOW)

The bag of words approach is the most popular and straightforward of all the other feature extraction methods. The bag-of-words model converts the frequency with which each word appears in the text into fixed-length vectors, regardless of how many times a word appears in the document. The number of times a word appears or the order in which the words appear is unimportant; all that counts is that the phrase appears in a list of terms. To demonstrate BOW, let's look at an example. Consider given two sentences as documents

1) Feature Extraction is necessary

2) Feature Extraction is necessary and important

We choose the unique words of the sentences and make a set of words that is used for further work. To perform bag-of-words, we all have to count how many times each word appears in each document.

Document No	Feature	Extraction	is	Necessary	And	Important
01	1	1	1	1	0	0
02	1	1	1	1	1	1

Hence, we have the following vectors for each of the documents of fixed length -6.

Document 1: [1, 1, 1, 1, 0, 0]

Document 2: [1, 1, 1, 1, 1, 1]

Bag-of-Words Limitations: Using bag-of-words to build vectors for huge documents will result in enormous vectors with too many null values, resulting in sparse vectors. Because the semantics of distinct document sentences differ, it will generate identical vectors.

TF-IDF Vectorizer: The TF-IDF is a numerical metric for determining the importance of a word in a document. Once the textual data has been converted to numerical data, it can be used in a machine learning model. The TF-IDF approach generates a relevant phrase that can be used to comprehend the complete context. Rather than reading the complete text, the TF-IDF vectorizer has been enhanced to improve the accuracy of the feature extraction approach [13] [14].

Term Frequency (***TF***), TF means term frequency which means how frequently a term occurred in a document

$$\text{TF}(\text{t},\text{d}) = \frac{\text{number of times t occurs in documents 'd'}}{\text{total word count of document 'd'}}$$

Inverse Document Frequency (***IDF***), The IDF is a metric that determines how essential a phrase is. When calculating Term Frequency (TF), all terms are given equal weight. However, it is well known that some phrases, such as "is," "of," and "that," may appear frequently but have little meaning.

As a result, we must scale up the rare phrases while weighing down the frequent ones, as seen below.:

$$\text{IDF}(\text{t}) = \log_e\left(\frac{\text{Total number of documents}}{\text{Number of documents with term t in it}}\right)$$

The inverse document frequency of the word across a set of documents. This means, how common or rare a word is in the entire document set.

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

Here, d refers to the document and t refers to a term or word we prefer to evaluate [28]. The closer it is to 0, the more common a word is. So, if the word is very common and appears in many documents, this number will approach 0. Otherwise, it will approach 1. Consider a 1000-word manuscript in which the word hive appears 50 times. Let's say there are ten million documents and the phrase hive appears in 1000 of them.

The TF for hive is then (50/1000) = 0.05.

And IDF is calculated as $\log_e(10,000,000/1000) = 4$

Thus TF-IFD in this case is = (0.05) × 4 = 0.2; which is close to 0. That means the word is relevant to this document.

3.2.3. Feature Reduction

We use the feature reduction step to reduce the text size and make it competitive for the model development.

3.3. Data Split

We divide our dataset into two parts, (1) data to train the model—Training data set and (2) data for testing the model—Testing data set. Here we used 80% of the total data for training purposes and the rest 20% for testing purposes. Using sklearn's "train_test_split ():" method, we have split the dataset.

3.4. Building Classification Model

The classification process began with the goal of creating a categorization model. We employ a total of 14 distinct classifiers. To train the model, the retrieved features from the data processing stage are input into several classifiers such as passive-aggressive, regression, linear SVC, stochastic gradient classifier, and random forest classifiers. The classifier gained prediction knowledge about the test dataset once it was trained.

Machine Learning Algorithms

Classification models of different types have been applied but our goal is to identify the most promising classifiers by tuning their parameters. Accuracy can be checked by comparing the results through performance metrics.

1) *Passive Aggressive Classifier*

Among all the classifiers that deal with big data or large-scale data, Passive Aggressive is the best option. It's best for real-time or online data like Facebook, Twitter, and other social media platforms. It takes the data from the internet and processes it step by step. The model is trained using the data collected. The best example of passive-aggressive classification is the detection of fake news. There are two phases to passive-aggressive work. The algorithm's initial phase is Passive, in which it keeps the model and makes no changes if the prediction is right. The second phase is aggressive [15]. When the algorithm makes changes to the faulty forecast, the result is correct.

2) *Random Forest Classifier*

Random Forest is a supervised machine learning classification algorithm made out of a large number of decision trees. Each decision tree is built using bagging and feature randomness [16]. The random forest's outcome is determined by the decision tree's forecast. The result will be an averaged or largely voted projected result. A random forest method removes the restrictions of a decision tree algorithm. It improves precision while reducing dataset overfitting [17].

3) *Stochastic Gradient Descent Classifier*

SGD Classifier's optimization approach is the use of the Stochastic Gradient Descent (SGD) algorithm. SGD is a simple but effective method for training linear classifiers with convex loss functions, such as SVM and Logistic Regression. SGD facilitates minibatch [18] (online/out-of-core) learning, and zero-mean data with unit variance is necessary for the optimal result. The cost function is calculated using Stochastic Gradient Descent, abbreviated as SGD, with only one observation. Designers proceed onto each observation, computing the cost and changing the parameters one by one through this classifier.

4) *Perceptron Classifier*

Perceptron is one of the most basic types of artificial neural networks (ANN). This single neuron model may be used to solve two-class classification tasks and provides the foundation for much larger networks to be developed. It's supervised binary classifier learning. Based on the layers, Perceptron models are divided into two types (1) Single-layer Perceptron Model & (2) Multi-layer Perceptron model. Single-layer perceptron can learn only linearly separable patterns. Whereas a multi-layer perceptron model has the greater processing power and can process linear and non-linear patterns [19].

5) *Decision Tree Classifier*

By examining a dataset's properties, a Decision Tree method has been built to predict the dataset's class. It operates by comparing the root property's values to the values in the real dataset. It repeats the procedure until it reaches the leaf node, jumping from one sub-node to the next based on the comparison [20]. The primary goal is to develop a training model that can predict the class of targeted variables by learning basic decision rules from training data. The Decision Tree method is one of the simplest and widely used classification algorithms because it closely resembles human decision-making capacity though it may have an overfitting problem that a Random Forest can solve [21].

6) ***Ridge Regression Classifier***

Ridge regression is a technique used in data models to eliminate multicollinearity. It presupposes that the input and target variables have a linear relationship. where the number of observations is less than the number of predictor variables Ridge regression is the best option [22]. During training, it is an extension of linear regression [23].

7) ***K-Nearest-Neighbors Classifier***

K-Nearest-Neighbors (K-NN) is a nonparametric algorithm that is simple but effective in many situations. The K-NN algorithm preserves all existing data and groups new data points together based on their similarity. As new data is generated, the K-NN algorithm can swiftly classify it into the appropriate category [24].

8) ***Adaptive Boosting Classifier***

Adaptive Boosting is a technique for reducing supervised learning's bias and variance. It outperforms all other models because it increases the model's correctness, which can be validated by working in sequence. It raises the weights of misclassified cases and lowers the weights of correctly classified examples with each round [25]. The finest aspect is that the AdaBoost method is distinguished from all other boosting algorithms by the weighting approach used after each iteration. Random Forest differs from AdaBoost in that it creates a "n" number of appropriate trees, each consisting of a start node and multiple leaf nodes, but there is no predetermined depth. However, with AdaBoost, the method only creates a Stump node, which has two leaves. These stumps are slow learners, as it greatly increases classification performance [26].

9) ***Linear SVC*** **(*****Support Vector Classifier*****)**

The Linear SVC technique conducts classification using a linear kernel function and works well with huge dataset. It is designed to fit data and create a "best fit" hyperplane in N-dimension space. The extreme decision boundary is referred to as a hyperplane. The hyperplane is chosen to have the greatest distance from the data point of each group, which assists in categorizing the incoming input [27].

10) ***Extra Trees Classifier***

By fitting many randomized decision trees (a.k.a. extra-trees) on distinct sub-samples of the dataset, this class implements a meta estimator that uses averaging to boost projected accuracy and control over-fitting [28]. To avoid over-learning and overfitting, ExtraTreesClassifier, like RandomForest, randomizes certain decisions and data subsets [29].

11) ***Gradient Boosting Classifier***

Gradient boosting is a regression and classification machine learning technique that generates a prediction model in the form of an ensemble of weak prediction models. This method creates a model step by step and then generalizes it by permitting the optimization of any differentiable loss function. Gradient boosting is an iterative process that merges weak learners into a single strong learner [30]. A new model is fitted for each weak learner to produce a more accurate estimate of the response variable. The new weak learners are maximally correlated with the negative gradient of the loss function, associated with the whole ensemble [31]. The idea of gradient boosting is that you can combine a group of relatively weak prediction models to build a stronger prediction model.

12) ***Logistic Regression***

Despite its name, regression model is a supervised learning method which is mostly used to solve binary "classification" problems. Regardless of the fact that the terms "regression" and "classification" are incompatible, logistic regression focuses on the term "logistic," which refers to the logistic function that performs the classification operation in the algorithm. For binary classification applications, logistic regression is commonly used since it is a simple yet powerful classification technique [32]. Customer churn, spam email, and website or ad click prediction are just a few of the issues that logistic regression can help with. It's even used as a layer activation function in neural networks.

13) ***SVC Classifier***

SVC is a nonparametric clustering technique that makes no assumptions about the size or structure of the data clusters. It works best for low-dimensional data. If data is high-dimensional, then probably need to do some preprocessing, such as utilizing principal component analysis [33].

14) ***Bagging Classifier***

Bagging classifiers are ensemble meta-estimators that fit base classifiers to random subsets of the original dataset and then average their individual predictions to obtain a final prediction [34]. A meta-estimator that incorporates randomness into the building technique of a black-box estimator (e.g., a decision tree) can frequently be used to reduce the variance of a black-box estimator.

3.5. Evaluation and Performance Module

After fitting all the classifiers into the system model, we verify the accuracy and performance based on some parameters like accuracy, precision, recall, and F-1 scores, confusion matrix. These are the evaluation matrices based on which our performance evaluation exercise has been performed.

Evaluation Metrics

When creating a machine learning model, it's important to remember that we'll need certain metrics to assess the model's quality. This is used to predict the model's accuracy for future data.

Confusion Matrix: The confusion matrix is a matrix for evaluating the classification model's performance [35]. The amount of right and incorrect guesses is the key to the confusion matrix, which is summarized using count values and broken down by class. Although it is simple to comprehend, the parameter employed is perplexing. The following are the conditions:

True Positive (*TP*): When the model classified the actual value and it is True

True Negative (*TN*): When the model classified the actual value and it is negative

False Positive (*FP*): When the classifier predicts the news is true but actually its false

False Negative (*FN*): When the classifier predicts the news is false but actually its true

Precision Metrics: Precision metrics tell us how many of the correctly predicted cases turned out to be positive. These metrics determine whether the model is reliable or not [36].

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

Recall Metrics: Recall metrics shows the number of really positive cases that could be predicted correctly using the model [36].

$$\text{Re} \quad \text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

F1 Score: F1 gives us the combined idea about precision and Recall metrics. That means when we try to upgrade the value of precision Recall goes down and vice-versa [36].

$$\text{F1}_{\text{score}} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

Accuracy: Accuracy is the fraction of correct predictions and total predictions made by the classifiers [36]

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

4. Results

The accuracy of all the fourteen classifiers is measured based on the dataset we have taken. It has been observed that, among all the classifiers, 5 classifiers' accuracy is higher than 94%. They are Passive Aggressive Classifier, SGD Classifier, Perceptron Classifier, RidgeClassifier, and LinearSVC Classifier. Uma Sharma *et al.* [37] used confusion matrices to demonstrate the effectiveness of various classifiers for static and dynamic systems. For dynamic systems, Passive Aggressive Classifier's accuracy was 92.73%, while Logistic Regression Classifiers accuracy was 65%, which was higher than Nave Bayes and Random Forest Classifiers. Our evaluation was based entirely on the Static system, and we evaluated the performance of a total of 14 classifiers, as shown below. Table 1 shows the results of various classification methods.

Now we've done some research to see how well a soft and hard voting system performs. When four (04) classifiers are combined in a soft voting analogy, the overall performance does not improve satisfactorily. A similar comparison has been used for the Hard voting algorithm, which results in a stronger assessment response than Soft Voting, but the output is still not higher than the individual performance of the classifiers themselves. We tried to integrate more classifiers to check the performance but combining 4 classifiers yielded no improvement as shown in Table 2. There is no substantial gain when adding more 4 classifiers to the previous configuration as shown in Table 3. In fact, as compared to the previous one, the overall accuracy has decreased.

Table 1. Performance comparison of various classification methods.

SN	Classifications Methods	Precision (%)	Recall (%)	F1 Score (%)	Accuracy (%)
1	Extra Trees Classifier	88	96	92	91.47
2	Gradient Boosting Classifier	92	92	92	91.79
3	Logistic Regression	94	92	93	93.13
4	Passive Aggressive Classifier	95	96	96	95.89
5	SGD Classifier	94	94	94	94.08
6	Perceptron Classifier	94	94	94	94.23
7	RidgeClassifier	95	94	95	94.71
8	LinearSVC Classifier	96	95	95	95.50
9	Random Forest Classifier	91	95	93	92.65
10	AdaBoost Classifier	89	87	88	88.31
11	SVC Classifier	50	100	66	49.80
12	Bagged Classifier	00	00	00	50.19
13	Decision Tree Classifier	82	85	83	83.10
14	KNeighborsClassifier	99	13	23	57.00

Table 2. Performance comparison of voting classifier (combining 4 classifiers).

SN	Voting Classifier	Classifications Methods	Individual Accuracy (%)	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)
1	Soft	Logistic Regression	93.13	94.24	95	93	94
2		SGD Classifier	94.08				
3		SVC Classifier	49.80				
4		AdaBoost Classifier	88.31				
1	Hard	Logistic Regression	93.13	93.92	95	93	94
2		SGD Classifier	94.08				
3		SVC Classifier	49.80				
4		AdaBoost Classifier	88.31				

Table 3. Performance comparison of voting classifier (Combining 08 Classifiers).

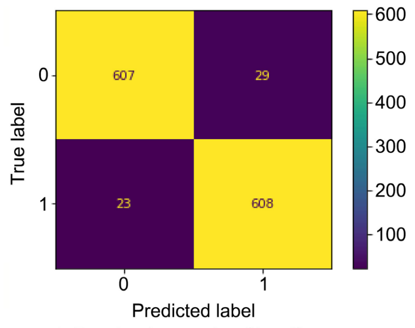
SN	Voting Classifier	Classifications Methods	Individual Accuracy (%)	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)
1	Soft	Logistic Regression	93.13	91.71	97	86	91
2		SGD Classifier	94.08				
3		SVC Classifier	49.80				
4		AdaBoost Classifier	88.31				
5		KNeighborsClassifier	57.00				
6		Decision Tree Classifier	83.10				
7		Random forest Classifier	92.65				
8		Bagged Classifier	50.19				
9	Hard	Logistic Regression	93.13	92.74	96	89	93
10		SGD Classifier	94.08				
11		SVC Classifier	49.80				
12		AdaBoost Classifier	88.31				
13		KNeighborsClassifier	57.00				
14		Decision Tree Classifier	83.10				
15		Random forest Classifier	92.65				
16		Bagged Classifier	50.19				

For all the 14 classifiers, identified confusion matrix has been tabulated below in Figure 2. In addition to that, confusion metrics for voting classifiers (Soft Voting and Hard Voting) with binding 4 classifiers and 8 classifiers are shown here.

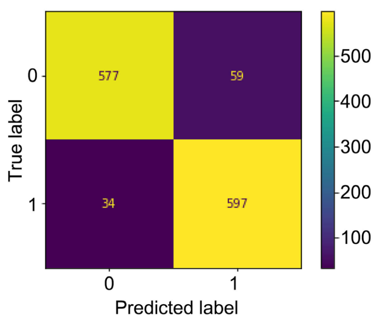
Using the voting analogy idea, however, no significant gain in performance has been discovered. To compare the performance of different classifiers, we used the stacking classification technique. In this comparison, we've used a variety of classifier combinations to stack and analyze performance.

We tested 7 distinct stacks (from Stack 1 to Stack 7), which we call it "model", to see how accurate they are. We utilized "Logistic Regression" as the final estimator classifier in these 7 stacking classifiers. The performance of those models is shown in Table 4. Here model 4 (comprising Logistic Regression, Passive Aggressive, LinearSVC, and Random Forest classifiers) and model 5 (comprising Passive Aggressive, SGD, Perceptron, Ridge, and LinearSVC classifiers) both provide an accuracy score of 95.97 percent, which is a higher than the individual accuracies and also higher than the earlier accuracy we saw in voting classifiers.

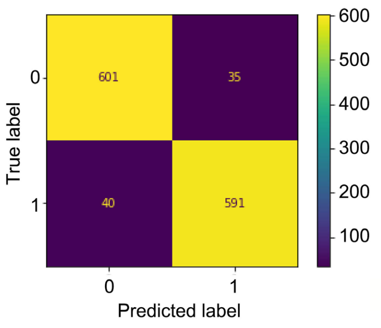
Model 6 and model 7, on the other hand, have the highest scores. Logistic Regression, Passive Aggressive, SGD, Ridge, and Random Forest Classifier are included in model 6, whereas Logistic Regression, Passive Aggressive, Ridge, LinearSVC, and Random Forest Classifier are included in model 7. The accuracy of both models is 96.13%. This is the most extreme of all the changes we've seen in any analogy so far.



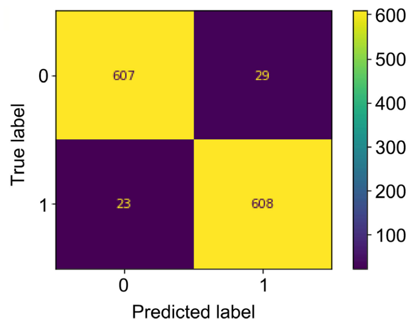
1. Passive Aggressive Classifier



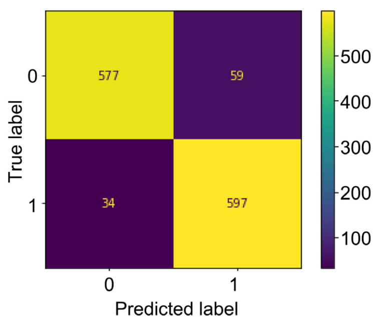
2. Random Forest Classifier



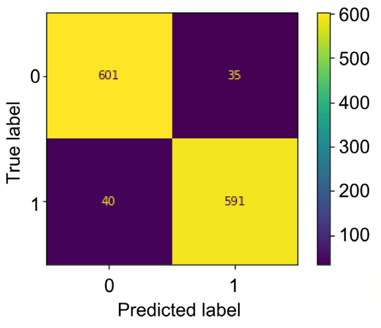
3. SGD Classifier



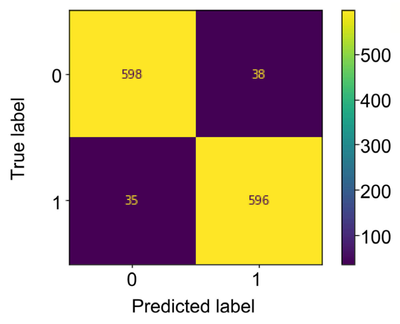
4. Passive Aggressive Classifier



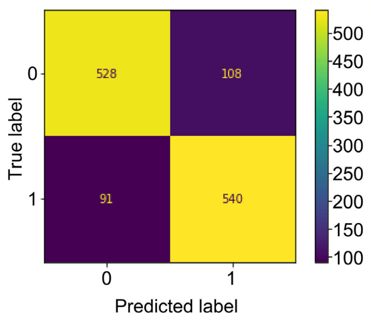
5. Random Forest Classifier



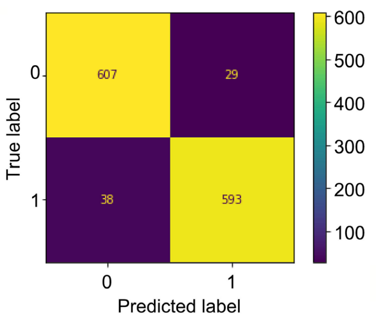
6. SGD Classifier



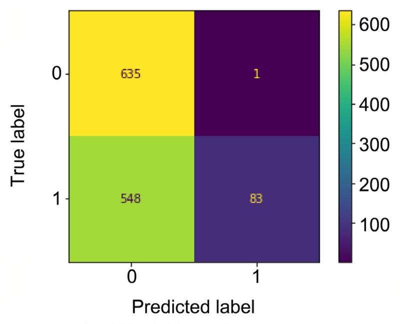
7. Perceptron Classifier



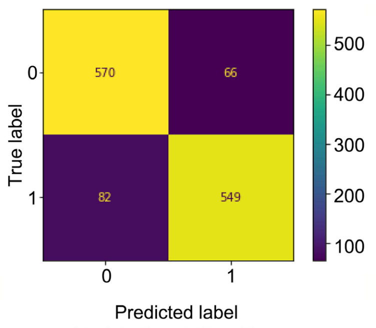
8. Decision Tree Classifier



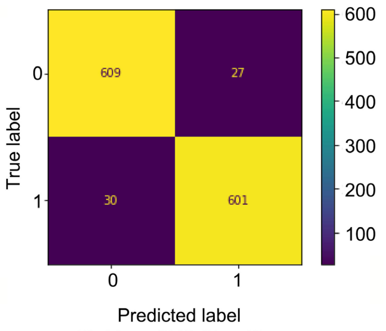
9. Ridge Classifier



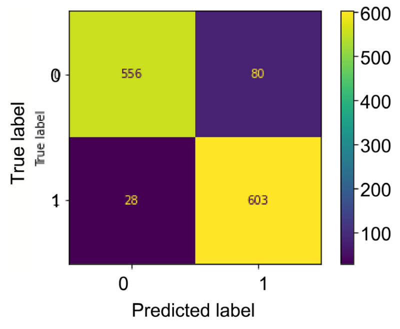
10. K-Neighbors Classifier



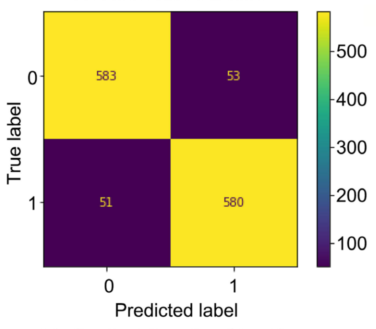
11. Ada Boost Classifier



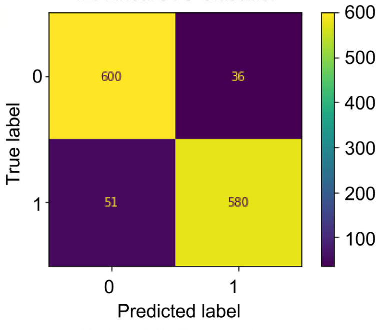
12. LinearSVC Classifier



13. Extra Trees Classifier



14. Gradient Boosting Classifier



15. Logistic Regression

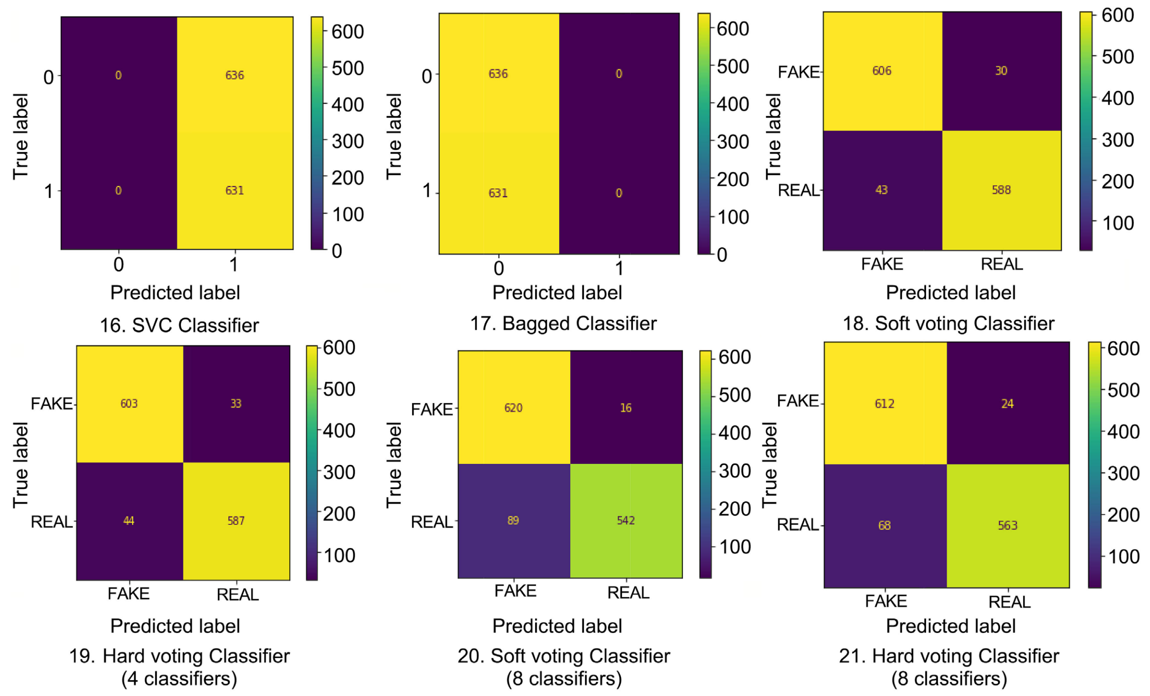


Figure 2. Confusion metrics for all the individual classifiers and voting classifier.

Table 4. Performance comparison of stacking classifier with various groups.

SN	Final Estimator	LR	LR	LR	LR	LR	LR	LR	AB	BC				
	Classifications Methods	Model 1	Model 2	Model 3	Model 4	Model 5	Model 6	Model 7	Model 8	Model 9	Precision (%)	Recall (%)	F1 Score (%)	Accuracy (%)
1	Extra Trees										88	96	92	91.47
2	Gradient Boosting										92	92	92	91.79
3	Logistic Regression (LR)			🟧	🟩		🟨	🟧	🟪	🟥	94	92	93	93.13
4	Passive Aggressive	🟨	🟦		🟩	🟦	🟨	🟧	🟪	🟥	95	96	96	95.89
5	SGD	🟨		🟧		🟦		🟧			95	93	94	94.00
6	Perceptron		🟦			🟦					94	94	94	94.23
7	Ridge	🟨	🟦			🟦	🟨	🟧	🟪	🟥	95	94	95	94.71
8	LinearSVC	🟨	🟦	🟧	🟩	🟦	🟨		🟪	🟥	96	95	95	95.50
9	Random Forest			🟧	🟩		🟨	🟧	🟪	🟥	91	94	93	92.59
10	AdaBoost (AB)										89	87	88	88.31
11	Decision Tree										82	85	83	83.10
12	SVC Classifier										50	100	66	49.80
13	Bagged Classifier (BC)										0	0	0	50.19
14	KNeighborsClassifier										99	13	23	57.00
15	Model (version 1)	🟨									95	96	96	95.65
16	Model (version 2)		🟦								95	96	96	95.73
17	Model (version 3)			🟧							94	94	94	94.08
18	Model (version 4)				🟩						95	97	96	**95.97**
19	Model (version 5)					🟦					95	97	96	**95.97**
20	Model (version 6)						🟨				96	97	96	**96.13**
21	Model (version 7)							🟧			96	97	96	**96.13**
22	Model (version 8)								🟪		95	96	95	95.42
23	Model (version 9)									🟥	95	95	95	95.34

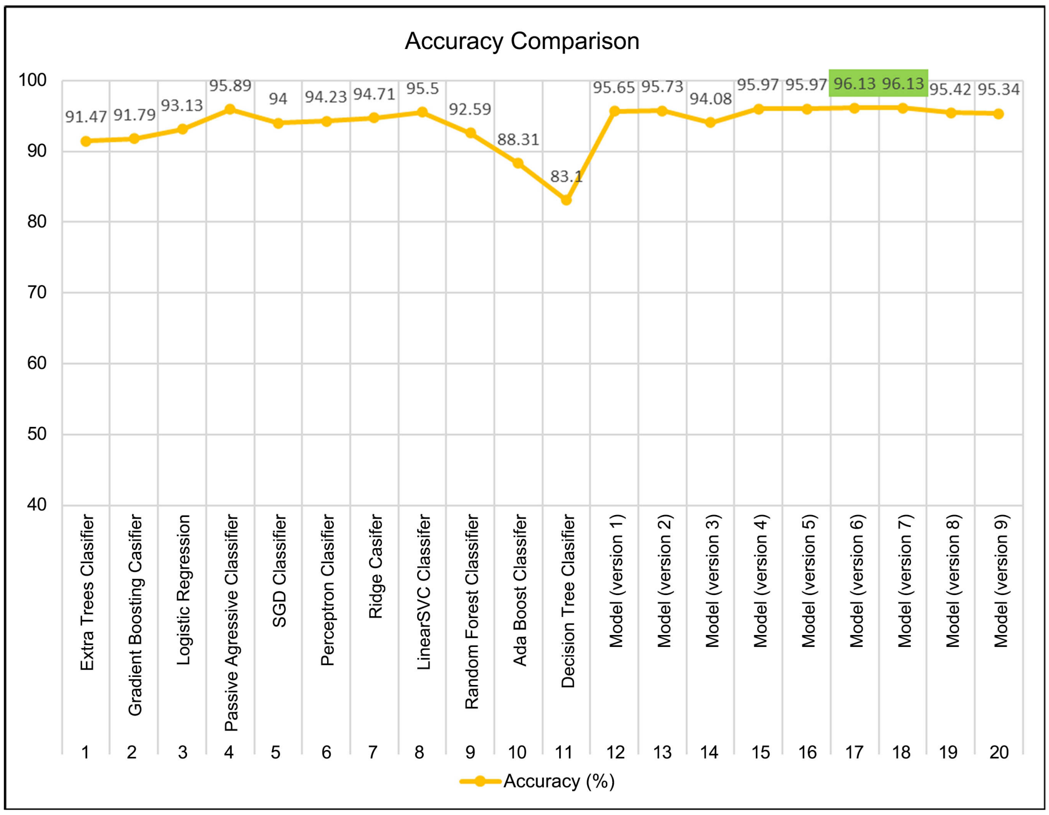


Figure 3. Accuracy comparison curve.

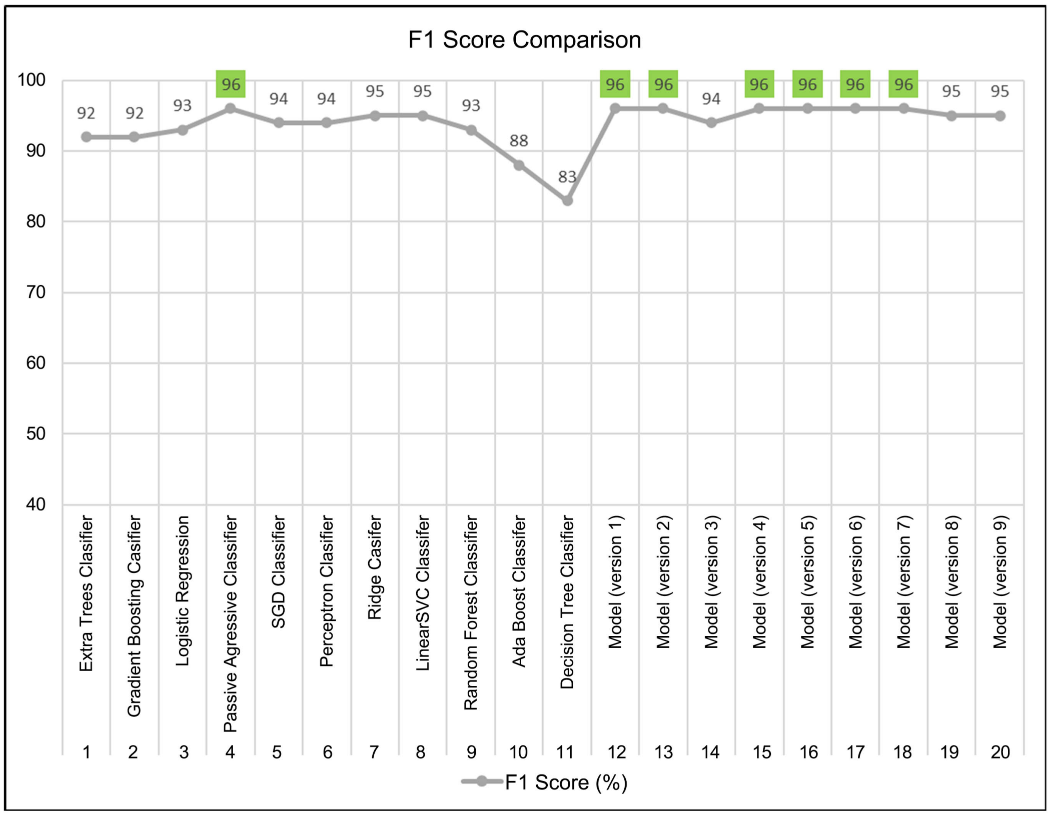


Figure 4. F1 Score comparison curve.

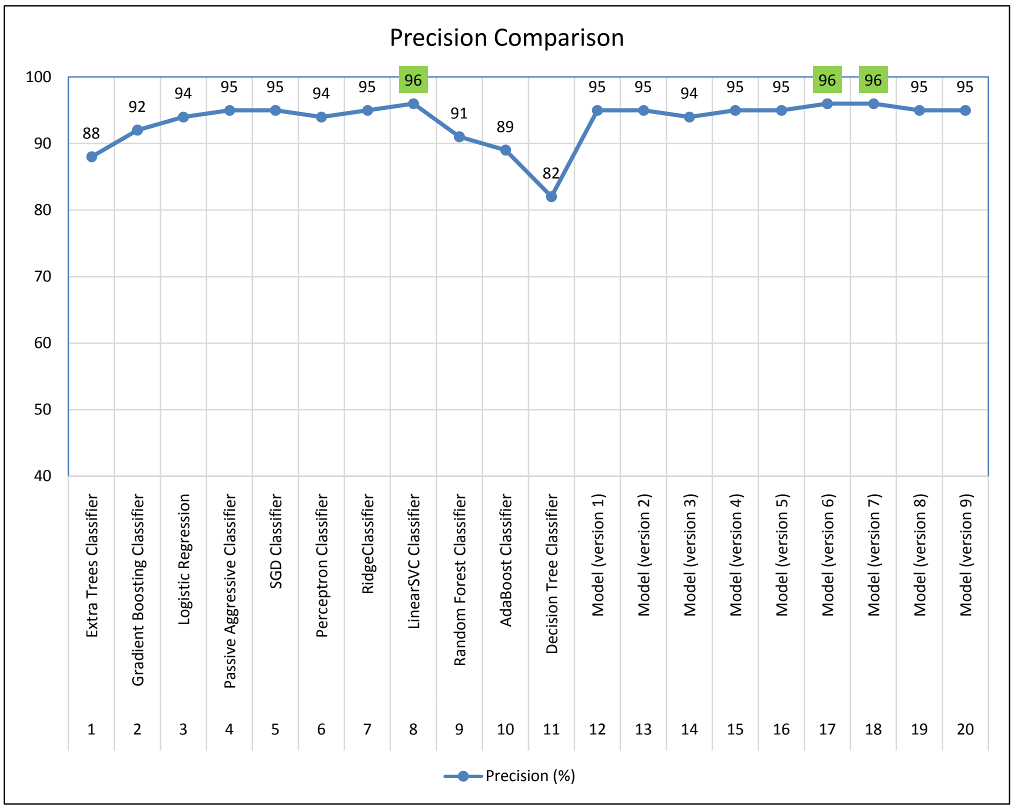


Figure 5. Precision comparison curve.

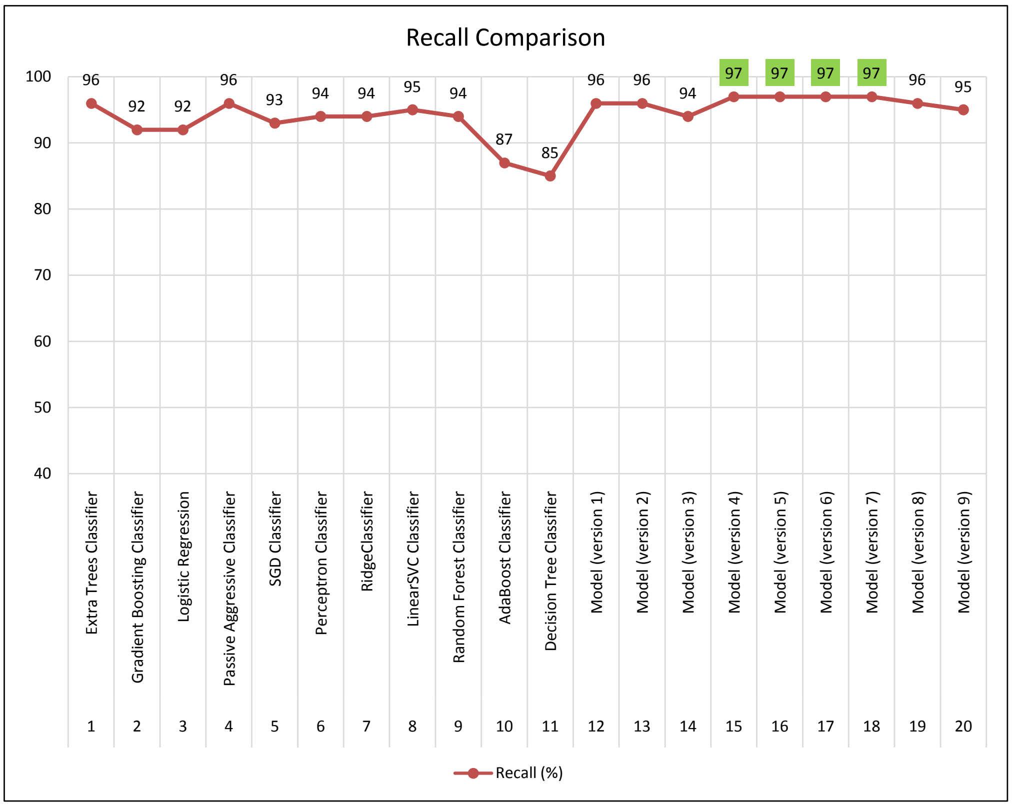


Figure 6. F1 Score comparison curve.

Instead of utilizing Logistic Regression, we tried a new analogy by using Adaboost and Bagged Classifier as the final estimator classifier. However, according to the examination, the accuracy was 95.42% and 95.34%, respectively. Therefore, no significant improvement has been found.

Instead of utilizing Logistic Regression, we have investigated a new analogy by using Adaboost and Bagged Classifier as the final estimator classifier. However, according to the examination, the accuracy was 95.42% and 95.34%, respectively which is lower than the gain identified on model 6 and 7. Among 14 Classifiers as SVC Classifier, Bagged Classifier & Kneighbors Classifier gives lower accuracy with a value of 49.80%, 50.19% & 57.00% respectively. We have dropped those classifiers from the graphical representation. Thus, the accuracy, F1 Score, Precision and recall curve are shown in **Figure 3-6** respectively.

5. Conclusions

In this experiment, we assessed the performance of 14 different classifiers and determined their accuracy. We attempted to discover the stacks of classifiers whose performance stands out above the rest based on our data. The accuracy has been listed in **Table 4** based on our analytics: Extra Trees, Gradient Boosting, Logistic Regression, Passive Aggressive, SGD, Perceptron, Ridge, Linear SVC, Random Forest, AdaBoost, Decision Tree, SVC, Bagged and KNeighbors Classifiers. We discovered that Passive Aggressive, SGD, Perceptron, Ridge, and LinearSVC provide a higher accuracy rate than the others, with accuracy levels exceeding 93%. The accuracy of the Rest 9 classifiers is insignificant.

Among the top 05 performing classifiers, the Passive-Aggressive Classifier has the highest accuracy and an F1 score. It also has the highest F1 score of all 14 classifiers. We analyzed the performance of a 04 classifiers stack (Logistic Regression, SGD, SVC & AdaBoost Classifier) using soft and hard voting classifiers and found the aggregated accuracy, precision, recall, and F1 score of the soft voting classifier is higher than the value obtained from hard voting classifiers. However, if a larger number of classifiers are stacked for voting classifiers' performance evaluation, the scenario changes. Stacking classifiers showed significantly improved accuracy in identifying fake news, while voting classifiers were unable to provide any appreciable gain. In the case of stacking classifiers for specified combinations, a significant improvement in performance has been observed. Although the performance is higher, besides having higher efficiency, its computational efficiency might be to some extent degraded in nature.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Khanam, Z., Alwasel, B.N., Sirafi, H. and Rashid, M. (2021) Fake News Detection

Using Machine Learning Approaches. *IOP Conference Series: Materials Science and Engineering*, **1099**, Article No. 012040.
https://doi.org/10.1088/1757-899X/1099/1/012040

[2] Smitha, N. and Bharath, R. (2020) Performance Comparison of Machine Learning Classifiers for Fake News Detection. 2020 *Second International Conference on Inventive Research in Computing Applications* (*ICIRCA*), Coimbatore, 15-17 July 2020, 696-700. https://doi.org/10.1109/ICIRCA48905.2020.9183072

[3] Agarwal, V., Sultana, H.P., Malhotra, S. and Sarkar, A. (2019) Analysis of Classifiers for Fake News Detection. *Procedia Computer Science*, **165**, 377-383.
https://doi.org/10.1016/j.procs.2020.01.035

[4] Uma Sharma, S.S. (2021) Fake News Detection Using Machine Learning Algorithms. *International Journal of Engineering Research & Technology* (*IJERT*), **9**, 509-518.

[5] Ahmed, S., Hinkelmann, K. and Corradini, F. (2020) Development of Fake News Model Using Machine Learning through Natural Language Processing. *International Journal of Computer and Information Engineering*, **14**, 454-460.

[6] About Dataset.
https://www.kaggle.com/datasets/antonioskokiantonis/newscsv?select=news.csv

[7] Behera, B. and Kumaravelan, G. (2020) Performance Evaluation of Machine Learning Algorithms in Biomedical Document Classification. *Performance Evaluation*, **29**, 5704-5716.

[8] Ramakrishna, S. (2021) Survey on Fake News Detection Using Machine Learning Algorithm. Bapatla Engineering College, Bapatla.

[9] Waykole, R.N. and Thakare, A.D. (2018) A Review of Feature Extraction Methods for Text Classification.

[10] Patil, D.R. (2022) Fake News Detection Using Majority Voting Technique.

[11] Jiang, T., Li, J.P., Haq, A.U., Saboor, A. and Ali, A. (2021) A Novel Stacking Approach for Accurate Detection of Fake News. *IEEE Access*, **9**, 22626-22639.
https://doi.org/10.1109/ACCESS.2021.3056079

[12] Waykole, R.N. and Thakare, A.D. (2018) A Review of Feature Extraction Methods for Text Classification. Pimpri Chinchwad College of Engineering, Pimpri-Chinchwad.

[13] Patil, L.H. and Atique, M. (2013) A Novel Approach for Feature Selection Method TF-IDF in Document Clustering. *IEEE 3rd International Advance Computing Conference* (*IACC*), Ghaziabad, 22-23 February 2013, 858-862.

[14] Jing, L.-P., Huang, H.-K. and Shi, H.-B. (2002) Improved Feature Selection Approach Tfidf in Text Mining. *Proceedings of the First International Conference on Machine Learning and Cybernetics*, Beijing, 4-5 November 2002, 944-946.

[15] Passive Aggressive Classifiers.
https://www.geeksforgeeks.org/passive-aggressive-classifiers

[16] Donges, N. (2021) Random Forest Algorithm: A Complete Guide.
https://builtin.com/data-science/random-forest-algorithm

[17] Mbaabu, O. (2020) Introduction to Random Forest in Machine Learning.
https://www.section.io/engineering-education/introduction-to-random-forest-in-machine-learning

[18] Simplilearn (2022) Stochastic Gradient Descent in SKLearn and Other Types of Gradient Descent.
https://www.simplilearn.com/tutorials/scikit-learn-tutorial/stochastic-gradient-desc

ent-scikit-learn

[19] Perceptron in Machine Learning. https://www.javatpoint.com/perceptron-in-machine-learning

[20] Decision Tree Algorithm, Explained. https://www.kdnuggets.com/2020/01/decision-tree-algorithm-explained.html

[21] Decision Tree Classification Algorithm. https://www.javatpoint.com/machine-learning-decision-tree-classification-algorithm

[22] Brownlee, J. (2020) How to Develop Ridge Regression Models in Python. https://machinelearningmastery.com/ridge-regression-with-python

[23] Great Learning Team (2020) What Is Ridge Regression? https://www.mygreatlearning.com/blog/what-is-ridge-regression

[24] K-Nearest Neighbor(KNN) Algorithm for Machine Learning. https://www.javatpoint.com/k-nearest-neighbor-algorithm-for-machine-learning

[25] Wu, S.Q. and Nagahashi, H. (2015) Analysis of Generalization Ability for Different AdaBoost Variants Based on Classification and Regression Trees. *Journal of Electrical and Computer Engineering*, **2015**, Article ID: 835357. https://www.hindawi.com/journals/jece/2015/835357/?utm_source=google&utm_medium=cpc&utm_campaign=HDW_MRKT_GBL_SUB_ADWO_PAI_DYNA_JOUR_X&gclid=CjwKCAjw2vOLBhBPEiwAjEeK9ht1j9EXIQVDz3XQrz_zpiAA45MQQIBPEmGAyQfLzWQZzSv3jtke9hoC9wQQAvD_BwE

[26] Great Learning Team (2022) The Ultimate Guide to AdaBoost Algorithm|What Is AdaBoost Algorithm? https://www.mygreatlearning.com/blog/adaboost-algorithm

[27] Pandey, S., *et al.* (2022) Fake News Detection from Online Media Using Machine Learning Classifiers. *Journal of Physics: Conference Series*, **2161**, Article ID: 012027. https://iopscience.iop.org/article/10.1088/1742-6596/2161/1/012027/pdf

[28] https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.ExtraTreesClassifier.html

[29] Bhandari, N. (2018, October 23) ExtraTreesClassifier: How Does ExtraTreesClassifier Reduce the Risk of Overfitting? https://medium.com/@namanbhandari/extratreesclassifier-8e7fc0502c7

[30] Brownlee, J. (2016, September 9) A Gentle Introduction to the Gradient Boosting Algorithm for Machine Learning. https://machinelearningmastery.com/gentle-introduction-gradient-boosting-algorithm-machine-learning

[31] What Is Gradient Boosting? https://deepai.org/machine-learning-glossary-and-terms/gradient-boosting

[32] Logistic Regression in Machine Learning. https://www.javatpoint.com/logistic-regression-in-machine-learning#:~:text=Logistic%20regression%20is%20one%20of,of%20a%20categorical%20dependent%20variable

[33] Support Vector Machine Algorithm. https://www.javatpoint.com/machine-learning-support-vector-machine-algorithm

[34] ML|Bagging Classifier. https://www.geeksforgeeks.org/ml-bagging-classifier

[35] Guide to Confusion Matrix Terminology. https://www.dataschool.io/simple-guide-to-confusion-matrix-terminology

[36] Brownlee, J. (2020, January 3) How to Calculate Precision, Recall, and F-Measure for Imbalanced Classification. https://machinelearningmastery.com/precision-recall-and-f-measure-for-imbalance

d-classification

[37] Sharma, U., Saran, S. and Patil, S.M. (2020) Fake News Detection Using Machine Learning Algorithms. *International Journal of Creative Research Thoughts*, **8**, 1394-1402. https://doi.org/10.22214/ijraset.2020.6125