International Journal of Environmental Sciences
ISSN: 2229-7359
Vol. 11 No. 21s, 2025
https://theaspd.com/index.php

Ensemble Based High Performance Deep Learning Model for Fake News Detection

Ms. Prateema Sharma[1], Dr. Tapsi Nagpal[2]
[1]Research Scholar, Department of Computer Science & Engineering, Lingaya's Vidyapeeth, Faridabad (Haryana) – 121002
[2]Sr. Associate Professor, Department of Computer Science & Engineering, Lingaya's Vidyapeeth, Faridabad (Haryana)- 121002

Abstract
Since the advent of social media platforms such as Facebook and Twitter, information has been disseminated at a velocity never before seen in human history. Many users are submitting statements that are not based on reality because of the current trend in social media usage, which has led to an explosion of user-generated material. Determining if a text contains deceptive or inaccurate information is a challenging task to automate. Even an expert in the field has to look at an article from several perspectives before making a judgement on its credibility. Using an ensemble machine learning approach, we provide a way for automatically categorizing news items. Our study delves into several textual elements to distinguish between factual and fraudulent material. We use these features to train an ensemble of machine learning algorithms, and then we test them on four real-world datasets to see how well they did. We propose an ensemble learner strategy, and the results show that it outperforms the individual learner approaches.
Keywords: *- Fake News Detection, Hybrid, Algorithm, Information Verification, Machine Learning, Data Analysis*

1. INTRODUCTION

Thanks to the meteoric rise of social media platforms like Twitter and Facebook, information is presently being disseminated at a rate never seen by humanity. The widespread use of social media has several benefits, but among the most important is assisting news organizations in providing subscribers with real-time updates on breaking news items. Online resources such as news websites, blogs, and social media feeds have joined traditional print media in disseminating news [1]. It could be easier for customers to get updates. A whopping 70% of all traffic to news websites comes from Facebook referrals. [2] in These days, people talk about vital topics like health, education, and democracy on social media, which is why these platforms are so crucial and effective. In contrast, there are groups who make bad use of these platforms, often for financial gain [3, 4] but also to spread satire or absurdity, construct biased opinions, or influence people's thoughts. The phrase "fake news" describes this phenomenon.

The 2016 US elections were the clearest illustration of the exponential growth in the spread of misinformation over the last ten years [5]. The widespread distribution of inaccurate information online has caused a number of issues in fields other than politics, including health, sports, and science [3]. Since even a little amount of rumor may bring the market to a halt, the financial markets are one area where misleading information may have a big influence [6].

Everything we take in influences our perspective on the world and, therefore, the choices we make. There is mounting evidence that customers have behaved irrationally in response to news stories that turned out to be false [7, 8]. There is some misinformation floating around on the new corona virus's background, traits, and behaviours on the internet [9]. Things became worse when word got out about the fake items on the internet. Finding this kind of news online is a real pain.

I am grateful Using a variety of computational methods, certain publications may be identified as potentially fraudulent [10]. Much of this approach is based on fact-checking websites like "PolitiFact" and "Snopes" that examine assertions. Searches in many databases have yielded lists of domains that have been determined to be fraudulent or otherwise misleading. the eleventh One drawback of these systems is that they rely on human skill to identify fake articles or websites. To add insult to injury, fact-checking websites can only handle content from a certain industry, namely politics, and not pop culture, sports, or contemporary technology.

Many different kinds of data storage are available on the Internet. Because it relies only on human comprehension, detecting and classifying web media without a strict organizational structure—such as news articles, videos, and audio recordings—is infamously difficult. In order to differentiate between factually sound articles and those that are deliberately deceptive, computer approaches such as natural language processing (NLP) may be used [12]. A different approach would be to contrast the dissemination

of real news with that of fake news [13]. The technique, to be more specific, examines how fake and actual news stories propagate over a network. In principle, one might use the response something gets to determine whether it's true or not. A hybrid approach that considers both the social response and the language features of an item may be used to ascertain whether or not the piece is dishonest.

Finding and classifying false material on social media platforms like Facebook and Twitter has been the subject of several research [13, 14]. It is possible to generalize machine learning models to other domains by first conceptualizing false news as belonging to one of many categories [10, 15, 16]. In their study of published texts, Ahmed et al. [17] found n-grams together with other linguistic features. Decision trees (DT), logistic regression (LR), support vector machines (SVM), K-nearest neighbors (KNN), and stochastic gradient descent (SGD) were among the machine learning models that were subsequently developed. The most accurate models (92% accuracy) were those that used logistic regression and support vector machines. The accuracy of the calculated grams for an item decreased as the quantity of grams increased, according to the research. This is a tendency of classification-based learning systems. Several models were improved by Shu et al. [12] by combining textual traits with auxiliary data, such as user social behavior on social media. Various social and psychological strategies for identifying online fraud were also investigated by the authors. Various data mining techniques for building models and extracting features were outlined by the writers. Knowledge, manner of writing, social setting, stance, and transmission are the building blocks of theories.

A alternative strategy is used by Wang [18]. A plethora of machine learning models were developed by the author using metadata and textual features. Convolutional neural networks (CNNs) were the principal tool used by the writer. The connections between the metadata vectors are further explained by a convolutional layer after a bidirectional LSTM layer. A fully connected layer was used to generate the final prediction. It took the text representations obtained from maximum pooling and the metadata representation from the bidirectional LSTM and fed them into it using a SoftMax activation function. A dataset including comments from two separate political parties is used in an analysis. Among the many bits of data included in the feature set is the subject matter; for example, the speaker, occupation, state, political party, location, and background are all potential points of discussion. An accuracy rate of 27.7 percent was achieved when factors such as speaker and text were combined. A 27.4 percent accuracy was produced by combining all metadata components with text. An method for article categorization into "agree," "disagree," "discuss," and "unrelated" is introduced by Riedel et al. [19]. This categorization is based on how well the article's title matches the material it presents in its body. We used a multi-layer perceptron (MLP) classifier that had one hidden layer and used a SoftMax function for the last layer's output. A multi-layer perceptron classifier was created, with a hidden layer and a SoftMax function used for the final layer's output. Word frequency (TF) and term frequency-inverse document frequency (TF-IDF) were among the several linguistic characteristics used by the writers. Accurately named, organized, and tagged articles made up the dataset. When tested with samples labeled as "disagree," the system showed signs of imprecision. In contrast, it performed well with "agree" samples. With careful adjustment of many hyperparameters connected to a simple MLP model, the authors were able to get an overall accuracy of 88.46%. In their discussion of several veracity assessment strategies, Shu et al. [12] covered topics such as how to spot fake news pieces on the web. This article takes a look at two main types of assessment methods: Signals and procedures for network analysis are provided by languages. By combining the two approaches, a hybrid strategy for identifying fake news on the internet may be achieved. Discourse analysis, rhetorical structure, and detailed syntax are all linguistic methods. Various linguistic approaches may be used to create classifiers, such SVM and naïve Bayes models. Methodologies based on network analysis were shown by investigating and making sense of linked data and interactions inside social networks. Using a fresh perspective, Vosoughi et al. [13] look at characteristics of news items that are shared on social media. Specifically, the authors contrasted the propagation of actual news with that of rumors and fake news on platforms such as Twitter. Measures of size, maximum breadth, structural virality, the amount of time it takes for a cascade of rumors to reach a certain depth and number of people on Twitter, average width of rumor cascades at different depths, and the number of people affected by the spread of fake news online are all examined in this paper.

1.1. Findings

The existing corpus of misinformation includes numerous examples of text classification employing both supervised and unsupervised learning methods. [20, 21]. Most of the research is specific to domains or datasets, with the political domain receiving the most extensive coverage [10, 19, 21]. Therefore, the

trained algorithm is domain-specific, meaning it performs poorly when presented with articles from different fields. The textual structure of articles from various domains is varied, making Training a generic algorithm to excel across all distinct news areas is a daunting task. Using an ensemble learning strategy, we provide a method to identify false news in this study. In order to differentiate between actual and false information, our research investigates several textual features. Utilizing these characteristics, we train a set Several techniques for machine learning that have not yet been adequately investigated in the literature using a variety of ensemble approaches. Learning models that utilize ensemble learners generally reduce error rates through methodologies like bagging and boosting, which have proven effective in various contexts.[22]. Using these techniques, training a variety of ML algorithms becomes much simpler efficiently and effectively. In addition, on four datasets that are available to the public and depict real-life situations, we conducted extensive testing. With the help of the F-1 score, accuracy, precision and recall, four of the most popular performance indicators. The results confirm the enhanced efficiency of our suggested method.

2. MATERIALS AND METHODS

Let's go over the algorithms, datasets, and performance assessment criteria that will make up our proposed framework.

2.1. Proposed Framework

Figure 1 shows how by utilizing ensemble approaches with diverse sets of linguistic data, our suggested strategy expands upon prior research features to determine the veracity of news stories across different domains. The proposed methodology is distinctive because it makes use of the LIWC feature set and ensemble methodologies from Linguistic Inquiry.

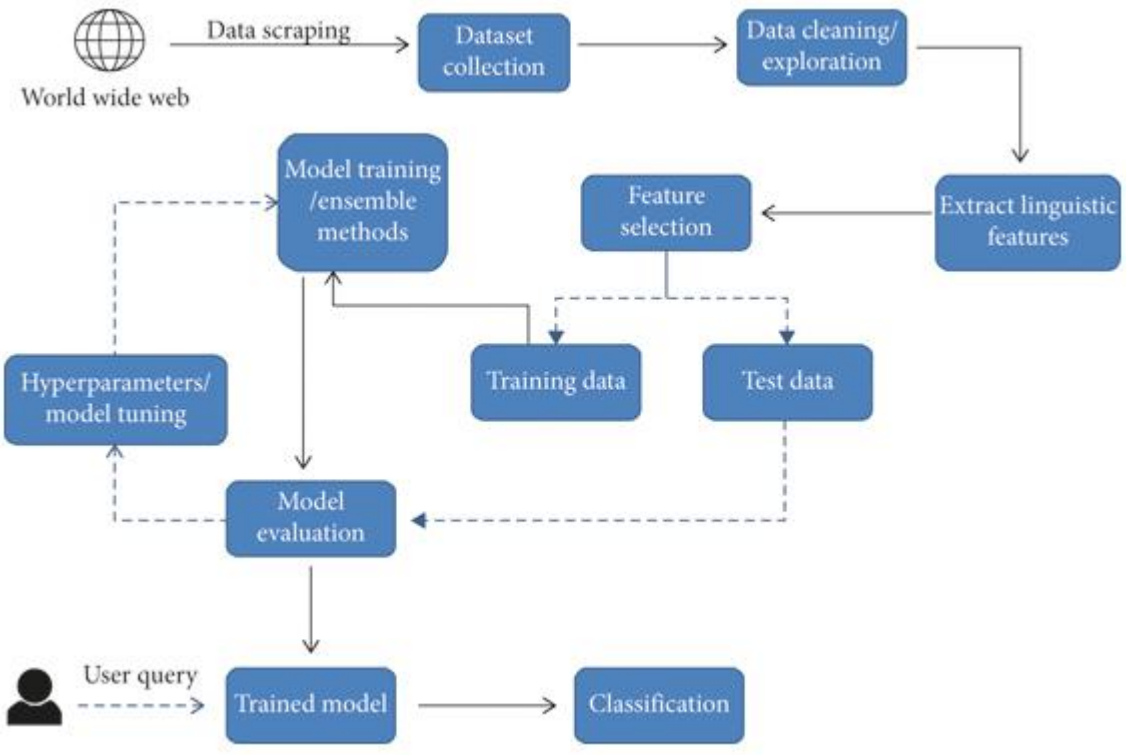

Figure 1 **Methodology for news article categorization and algorithm training**

Several reputable websites provide verifiable news stories, while others, like PolitiFact and Snopes, are used for verifying claims. Researchers also retain a current catalog of databases that are presently accessible in open repositories [11]. Furthermore, these repositories offer links to various fact-checking websites that may assist in combating the dissemination of misleading information. Nevertheless, for the sake of tests, three datasets that include both real and false news stories drawn from all walks of life, including the realms of politics, show business, technology, and athletics were chosen. The records are retrieved from the Internet and made publicly accessible online. All three datasets are publicly accessible on Kaggle. The ISOT Fake News Dataset [23] represents an additional publicly accessible dataset option. In Section 2.5, we provide a comprehensive overview of the datasets.

Before being used as a basis for training the models, the web-based corpus undergoes pre-processing. Unwanted information about the articles, including their authors, publishing date, URL, and category, is removed. Additionally, articles that do not have any body content or have a body of fewer than twenty words are also eliminated. In order to achieve consistency in format and structure, articles with multiple columns are converted to single-column articles. All the datasets undergo these processes to ensure that they are structured and formatted consistently.

Upon completion of data cleansing and investigation, the subsequent phase involves the extraction of linguistic features, contingent upon the identification of relevant qualities. To make linguistic features usable as training model inputs, we numerically transformed several textual properties. The detected features include the frequency of certain words that express good or negative emotions, the incidence of punctuation and function words, the usage of informal language, and the predominance of adjectives, prepositions, and verbs inside phrases. The LIWC2015 program employed a variety of discrete and continuous variables to systematically organize text and conduct feature extraction from the corpus. The LIWC program can extract 93 distinct traits from any given text. Encoding is not required for categorical variables, as the data collected by the program consists entirely of numbers. Scaling is utilized to guarantee that the values of all features are limited within the range of zero to one. This is essential because although the range of certain variables, such as percentages, is from zero to one hundred, while the range of others, like word counts, is completely subjective. Numerous machine learning models are subsequently trained utilizing the input characteristics. The datasets are partitioned into training and testing sets, utilizing a distribution of 70% for training and 30% for testing. To ensure training and test instances contain an equal distribution of real and fraudulent articles, the articles are subjected to a shuffling process.

Achieving an ideal equilibrium between variance and bias is crucial for improving accuracy on a particular dataset, necessitating the training of learning algorithms with a range of hyperparameters. The models undergo repeated training using a grid search to get the ideal values for each parameter. Although it is computationally expensive to find the ideal parameters using a grid search, doing so is essential for preventing models from being overfit or underfit to the data [26].

This study employs a distinctive methodology to analyse multiple ensemble algorithms, including bagging, boosting, and voting classifiers, to assess their effectiveness across a range of datasets. The first voting classifier employs random forest and KNN, while the next classifier incorporates logistic regression and SVM. Both voting classifiers utilize ensembles that include logistic regression, linear support vector machine (SVM), and classification and regression trees. Three learning models are utilized by both voting classifiers. Vote classifier training involves finding the optimal parameters for each model, and then testing them by having them choose an output label according to the consensus reached by all three models. A bagging ensemble comprising 100 decision trees was created using two boosting ensemble techniques: XGBoost and AdaBoost. The structure of all ensemble learners utilizes a k-fold cross-validation approach, with k established at 10. Section 2.2 provides an in-depth examination of the learning models employed. The evaluation of the models was conducted utilizing the metrics outlined in Section 2.6, these encompass accuracy, precision, recall, and F1 score.

2.2. Algorithms

The suggested technique was improved by using the following learning algorithms in order to evaluate the effectiveness of classifiers for determining whether or not news stories are fake.

$T(vdi(t))$ = 1/(1 + e^(−$vdi(t)$)) T(vid(t)) = 1 / (1 + e^(-vid(t))) (1)

Equation (2) serves as an example of the S-shaped TFs. In this equation, the symbol $Xdi(t+1)$ signifies the fourth component in the X solution for dimension d during iteration $t+1$, where rand indicates a random probability distribution derived from a random probability distribution. Once a probability has been established for each of the position vector's components, it is used to perform an update on those components.

$Xdi(t+1)$={0,1,if$rand<S_TF(vdi(t+1))$if$rand\geqslant S_TF(vdi(t+1))$Xid(t+1)=0,ifrand<S_TF(vid(t+1))1,ifrand⩾S_TF(vid(t+1))

(2)

The component of the next iteration is revised according to Equation (4), which is based on the probability values obtained from Equation (3). This is accomplished by the utilization of V-shaped TFs. For converting the GSA to binary, this equation was used in [46].

$T(Xdi(t))$ = |$tanh(Xdi(t))$ |T(Xid(t)) = |tanh (Xid(t)) |

(3)

$Xt+1=\{Xt, \neg Xt, \text{if} rand<V\_TF(\Delta Xt+1) \text{if} rand \geqslant V\_TF(\Delta Xt+1)$ Xt+1=Xt, ifrand<V_TF(ΔXt+1)¬Xt,ifrand⩾V_TF(ΔXt+1)
(4)

Binary Salp Swarm Algorithm

Salps, a kind of marine mammal that dwells in the seas and oceans, served as the inspiration for one of the SI algorithms that is commonly utilized. This method is known as SSA. The behavior they are replicating involves the pursuit of a food source. The foremost individual in the salp chain occupies the central role. The remaining salps in the chain align themselves with the food source. The subsequent salps in the chain adhere to the same directional trajectory as the leading salp. A dynamic movement occurs among the salps in relation to one another. As a result, there is a shift, either directly or indirectly, in the locations of the salps of the followers in relation to the salp of the leader. There was a continuous nature to the first SSA method. As a result, the same procedures employed in the conversion of the PSO to binary were also applied in the conversion of the SSA to binary. Figure 2 illustrates a single salp alongside a chain of salps.

$Fitness=\alpha\times(1-accuracy)+(1-\alpha*|S||W|)$
$Accuracy=TP+TNTP+TN+FP+FN;$
$Precision=TPFP+TP;$
$Recall=TPFN+TP;$
$F-Measure=(2\times Precision\times Recall)(Precision+Recall).$
(5)

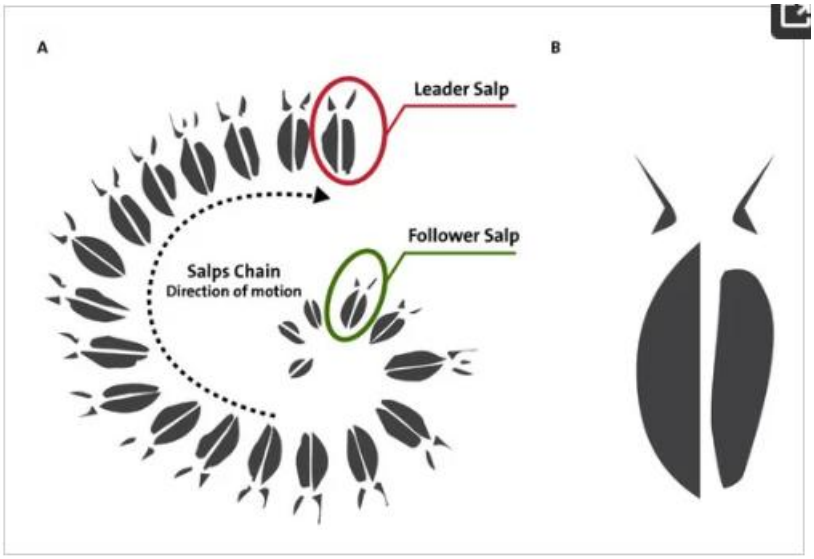

Figure 2. (A) Chain of salps(B)Single salp.

2.2.1. Logistic Regression

If you have a lot of characteristics and want to categorize texts as either true or false, or as a legitimate article or a fraudulent article, logistic regression (LR) models are a good fit since they give a simple equation for dividing problems into two or more classes [27]. We tested a plethora of settings in the LR model before we got the best accuracy and tuned hyperparameters to get the best outcome for each dataset. According to [27], In mathematical terms, this is the logistic regression hypothesis function:

The aim of logistic regression is to reduce the cost function to attain the best possible probability. The procedure entails utilizing a sigmoid function to transform the output into a probability value. The cost function seems to be well-defined as a result.

2.2.2. Support Vector Machine

The support vector machine (SVM) model acts as a supplementary approach for addressing binary classification challenges. This is accessible in various kernel functions [28]. Estimating a hyperplane, sometimes called a decision border, using a feature set One of the main purposes of a support vector machine (SVM) model is to classify data bits [29]. The hyperplane's dimension changes as the number of features increases. Finding the plane that most clearly divides two classes' data points is the goal, as there

are many possible hyperplanes covering an N-dimensional region. We may establish the SVM model's cost function in accordance with [30] mathematically, and it is shown in such a way that
Using a linear kernel, the function as described above. When fitting pieces of information that are complex or hard to categorize, kernels are often used. Our usage of support vector machines includes the sigmoid, kernel, Gaussian, and basic linear varieties.

2.2.3. Multilayer Perceptron

A multilayer perceptron (MLP) is characterized by its architecture, which consists of input, hidden, and output layers, setting it apart from other types of artificial neural networks. We have tweaked the model's parameters and layer count in our experiments to optimize it for prediction, even though a simple MLP just needs the three layers. An example of a simple multilayered perceptron model with one hidden layer is the function [31] that follows. Here we have the activation functions, the weight matrices, and the bias vectors. The Adam solver is employed in this configuration, utilizing ReLU as the activation function, and the architecture consists of three hidden layers.

2.2.4. K-Nearest Neighbors (KNN)

To make predictions based on given data, KNN does not need a dependant variable, making it an unsupervised machine learning model. After feeding the model enough training data, we step pause and allow it to decide whether area a specific dataset pertains to point is a member of. If K is set to 1, the KNN model assigns a new data point to the class with the closest distance to its nearest neighbors based on an estimate of the distance between the two points and the majority of the votes from those neighbors. The subsequent mathematical formulas can be utilized to calculate the distance between two points [31] :

2.3. Ensemble Learners

To get better results when classifying articles as true or false, we suggested combining textual features as feature input with current ensemble algorithms. Ensemble learners employ a variety of models developed through a particular approach to minimize the overall error rate and boost model performance, often leading to enhanced accuracy. Ensemble modelling is based on the same principle as the one we use daily, such as considering several experts' opinions before deciding to lessen the chances of making a bad one or having a terrible result. An example would be training a classification algorithm on a specific dataset with a one-of-a-kind set of parameters; this would allow the program to provide a decision boundary that somewhat matches the data. The parameters employed during the model's training process, as well as the particular data, are essential in influencing the algorithm's output. The risk of overfitting and skewed conclusions when applied to new data increases when there is less variation or homogeneity in the training data. Consequently, methods such as the likelihood of overfitting can be mitigated using cross validation. It is possible to train several models with different parameter sets using a random sample of data points, resulting in numerous decision limits. Hence, multi-algorithm training and merging their outputs for near optimal outcome, these difficulties may be handled and reduced utilizing ensemble learning approaches. Among these methods is the use of voting classifiers, in which the aggregated votes of all algorithms determine the final classification [32]. On the other hand, more ensemble approaches exist and may be used in many contexts, for example,

2.3.1. Random Forest (RF)

Another kind of supervised learning model Compared to decision trees (DT), random forests (RF) are superior. A network of many decision trees, Random Forest (RF) takes the most popular class into account when making its final prediction, but each tree in the network makes its own prediction regarding the outcome of a class. Because there are less connections between trees in random forest than in other models, its error rate is lower [33]. A grid search to train our random forest model with varying parameters, or the number of estimators, to find the optimal model for making accurate predictions was utilized. If someone is working on a regression or classification problem, you may use one of several methods to partition your decision trees. We have calculated the cost of splitting the dataset according to the Gini index in order to solve the classification problem. To determine the Gini index, aggregate all the squared probabilities of each class and subsequently subtract one from this sum. The following formula can be employed to compute the Gini index [34].

2.3.2. Bagging Ensemble Classifiers

Bootstrap aggregating, commonly referred to as a bagging classifier, was primarily employed as one of the initial ensemble methods to mitigate variation or overfitting within a training set. The random forest model represents a variant of bagging classifiers that enjoys widespread utilization. The bagging model aims to minimize overall variance by predicting significant votes derived from the number of trees. For each tree, data is randomly selected from the complete dataset through replacement sampling. This addresses the classification issue, as it appears to reason. The bagging model computes the average of multiple estimates when addressing regression problems.

2.3.3. Boosting Ensemble Classifiers

When training struggling role models to excel in the classroom, boosting is another often-utilized ensemble strategy. To do this, we train a forest of randomly selected trees, and then we use the results of each tree's majority vote to make our final forecast. This procedure, when used incrementally, allows low-achieving learners to correctly classify often misclassified data objects. All data points are first categorized for a particular problem using the same weighted coefficients. In later iterations, the weighted factors are increased for data points that are misclassified and reduced for those that are correctly classified [35]. Every iteration requires the development of a new tree to improve accuracy by examining prior errors and correcting data points that were misclassified in previous iterations. A notable issue related to enhancing ensemble techniques is the risk of overfitting to the training data, which can result in erroneous predictions for new instances. [36] Numerous solutions exist for those in need of an algorithm for regression or classification; however, for the classification objective, AdaBoost [38] and XGBoost [37] have been selected.

2.3.4. Voting Ensemble Classifiers

Voting ensembles facilitate the integration of multiple learning models that have been trained on the complete dataset, and are frequently employed to address classification problems[39]. All models cast a vote for the class it thinks will be most successful based on its prediction for a single data point from the sample. The final forecast is derived from the majority vote for a certain class after all models have predicted the result [32]. The implementation of voting ensemble is less complex than that of bagging and boosting algorithms. Bagging algorithms generate many datasets because, as previously said, they construct several subsets of data from the whole dataset by random sampling and replacement. Each dataset is utilized to train a model, and the ultimate outcome is the aggregate of all the contributions from the models. Boosting is a technique that involves training many models consecutively, each of which builds on the preceding one by assigning more weight to the misclassified data; the end result is a general model that can correctly detect the problem. Contrarily, a voting ensemble is a collection of separate models that work together to provide classification results that, by majority voting, add to the final forecast.

2.4. Benchmark Algorithms

Here, we go over the benchmark algorithms that we used to evaluate our technique.

2.4.1. Linear SVM

As mentioned in [21], a linear SVM approach is used here. To guarantee a valid comparison, the feature set is trained using a 5-fold cross validation using the linear SVM, as described in [21]. It is important to note that the method will be consistently referred to as Perez-LSVM throughout this study.

2.4.2. Convolutional Neural Network

To automate the identification procedure false news, in his study, Wang [18] made use of CNNs. We also applied the same procedure to our dataset. Using Wang's feature set was not possible since the dataset only contains brief remarks [18]. A Wang-CNN algorithm is used.

2.4.3. Bidirectional Long Short-Term Memory Networks

While Wang [18] also used Bi-LSTM, we simply used a different set of characteristics to mimic his approach. The Wang-Bi-LSTM approach is used in the book.

2.5. Datasets

Our research relies on publicly accessible, open-source datasets. False and true news stories from various domains are included in the data. Contrasted with false news websites that provide assertions unsupported by evidence, genuine news stories accurately portray actual events that have taken place in

the globe. Fact-checking services like politifact.com and snopes.com may be used to independently verify the veracity of political statements made in many articles. The following is a synopsis of the three datasets that were used in this investigation.

The "ISOT For the first time, a dataset called "Fake News Dataset" [23] (hereafter DS1) has been assembled from real and fake news items taken from the Internet. Plagiarism from many sites, mostly those that politifact.com has deemed as fraudulent, was used to create the phony reports, while the legitimate ones were sourced from the esteemed news website reuters.com. The collection includes 44,898 items, 21,417 of which are factual and 23,481 of which are false. Political news is the most heavily targeted domain within the overall corpora, which do include pieces from other domains.

The dataset, subsequently designated as DS2, comprises a total of 20,386 training articles and 5,126 testing articles, which can be accessed on Kaggle [24]. A variety of online resources played a role in the compilation of the dataset. The articles encompass a diverse array of subjects, spanning from politics to various other domains, and they feature both accurate and inaccurate information.

A third dataset, designated as DS3, comprises 3,352 articles, encompassing both authentic and fabricated content. This dataset is accessible on Kaggle [25]. We get the real stories from reputable internet publications like CNN, Reuters, and the New York Times, among others, and we sift through questionable news websites to find the false ones. It touched on topics including politics, culture, and sports.

The articles from all three datasets (DS4) are then integrated into one large dataset. The articles in each dataset are different in kind, thus we built a fourth dataset to test algorithms on datasets that span many domains.

When doing analysis of false news using optimization techniques, the relevant unstructured textual data set is seen as a search space. A binary search space is necessary for the optimization-based approach. The necessary textual data collection is pre-processed to do this. Case conversion and other filtering procedures (number, N-char, punctuation, etc.) are used to locate word roots in the pre-processing step. Afterwards, the masses (W_i) of every syllable is determined. The frequency with which each word appears in the dataset is used in this computation. Consequently, the value of a word is determined by using Wi=RiRmax. Here, R_i is the number of repetitions of the ith word, and R_{max} is the most iterations allowed. The optimization process will be negatively impacted by include terms with extremely low weights, thus the search area only contains words with weights over a particular threshold. Lastly, every record in the dataset is searched for using the search space's contained terms. An evaluation of 1 is performed if the term is found in the relevant record; an evaluation of 0 is performed otherwise. This transforms the dataset into a space where each row has just one value, either 1 or 0.

The procedure then builds potential members of the population to fill this search area. For instance, in a meta-heuristic algorithm that relies on swarm intelligence, there is a population P that has N potential candidates. (P={X1−→,X2−→,..XN−→−}). The variables of each candidate in the population take values between [0,1] (Xj−→={f1,f2,..fi..fK},fi∈[0,1],K=the number of the attributes,i∈[1,N]andi∈Z). Each candidate is assessed for every record in the modified dataset in order to determine their fitness values. During the fitness assessments, two factors are considered. Is the similarity ratio between the candidate values and the relevant record higher than a predetermined threshold value (τ)? This is the first criteria to be considered. This is when similarity functions like Jaccard Similarity, as shown in Eq., come in handy. 1, it is usable. The candidate's continuous value may be used in the similarity test [17] or transformed to binary form. Because they worked better, researchers in this study represented candidate values in similarity controls as binary numbers. The identity of the relevant record's class relative to the candidate class is the second factor to consider. Table 1.1 shows that these two criteria are considered simultaneously. The applicant may then determine the present values of TP, TN, FP, and FN. Once this is done for every record in the dataset, Eq. 2-[3] determines the candidate's fitness value. Accuracy, Precision, and Recall metrics may be obtained from the values of the top candidate at the conclusion of the iterations using (Eq. 3).

$$\text{Jaccard Value}X \to = \sum Ki = 1\text{Round}(fi) \times \text{Record}i \sum Ki = 1\text{Round}(fi) + \sum Ki = 1\text{Record}i - \sum Ki = 0\text{Round}(fi) \times \text{Record}i \quad (6)$$

$$F = k1 \times TP \times TN(TP + FN)(TN + FP) + k2 \times TPTP + FP + k3 \times TNTN + FN \quad (7)$$

$$\text{Acuracy} = TP + TNTP + TN + FP + FN\text{Precision} = TPTP + FP\text{Recall} = TPTP + FN$$

(8)

2.6. Performance Metrics

We employed many criteria to assess the efficiency of the algorithms. To a large extent, they all revolve around the confusion matrix. There are There are four factors that determine how well a categorization model does on the test data: genuine negative, erroneous positive, and confusion matrix, which is tabular and may be found in Table 1.

Table 1 **Confusion matrix.**

Random forest (RF)	**0.99**	0.35	0.95	**0.91**
Voting classifier (RF, LR, KNN)	0.97	0.88	0.94	0.88
Voting classifier (LR, LSVM, CART)	0.96	0.86	0.92	0.85
Bagging classifier (decision trees)	0.98	0.94	**0.94**	0.9
Boosting classifier (AdaBoost)	0.98	0.92	0.92	0.86
Boosting classifier (XGBoost)	0.98	0.94	**0.94**	0.89
Benchmark algorithms				
Perez-LSVM	**0.99**	0.79	**0.96**	0.9
Wang-CNN	0.87	0.66	0.58	0.73

Fig 3

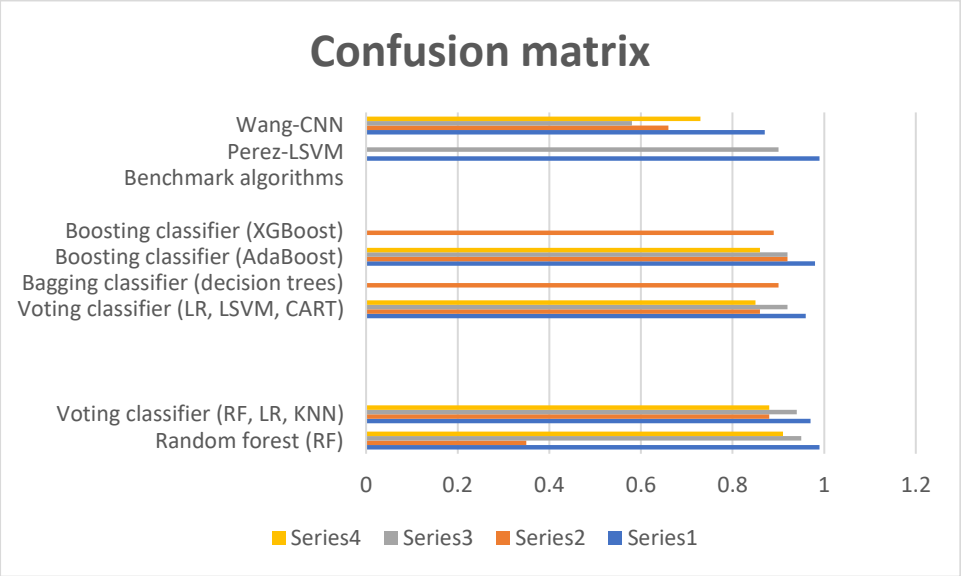

Fig 4

2.6.1. Accuracy

A common and widely used statistic is accuracy, which measures the proportion of true or erroneous observations that are accurately anticipated. Here is an equation that may be used to determine how accurate a model is:

A good model is often indicated by a high accuracy value. While training a classification model, there is a chance of undesirable results, such as false positives or trust concerns, if real data causes articles to be mistakenly forecasted as true or false, respectively. So, to make up for the misclassified observation, we have added three more metrics: recall, precision, and F1-score.

$$fj^{---} = 1CI + 1\sum CIk = 0fjk$$

(9)

$$f^{-} * j = 1N\sum Nm = 1fmj^{----}, N = \text{population size}$$

(10)

2.6.2. Recall

A high recall indicates that a large proportion of the genuine classes were correctly identified. For our purposes, it stands for the proportion of true articles anticipated as true relative to all true articles.

2.6.3. Precision

Alternatively, the accuracy score shows how many real positives out of all the projected true occurrences are. Here, accuracy is the percentage of positively predicted (true) articles that really get tagged as true:

$$xRi = \{\omega c - \partial L(\omega c - 1), ifxPi < \partial Land\rho i\omega u > 0.5\omega c\,\partial U, ifxPi > \partial Uand\rho i\omega u > 0.5)$$
$$\omega c = e - \alpha\omega c2 + \beta, (\alpha, \beta \epsilon R)$$

(11)

2.6.4. F1-Score

The F1-score demonstrates the balance between accuracy and recall. The process entails computing the average of the two values to ascertain the harmonic mean. The analysis includes both false positives and false negatives. The formula for calculating the F1-score is as follows:

$$\mathbf{\rho i = \{e - CISc, if\ cosS(xPnew, xPprovious) > \tau c1, otherwise)}$$

(12)

$$\mathbf{cosS(xPnew, xPprovious) = \sum Ki = 1xPnewixPpreviousi\sum Ki = 1xPnewi - - - - - - - - \sqrt{\sum Ki = 1xPpreviousi - - - - - - - - - - \sqrt{cosS(xPnew, xPprovious) = \sum Ki = 1xPnewixPpreviousi\sum Ki = 1xPnewi - - - - - - - - \sqrt{\sum Ki = 1xP}}}}$$

(13)

3. RESULTS AND ANALYSIS

Table 2 presents the results of the accuracy tests conducted on each of the four datasets. The random forest methodology combined with Perez-LSVM attains a peak accuracy of 99% on DS1, specifically the ISOT Fake News Dataset. Boosting classifiers, multilayer perceptrons, linear SVM, and bagging classifiers were used to achieve a 98% success rate. When group learning techniques are used instead of solitary learners, the mean DS1 accuracy is 97.67%. Between solitary and group learners, there is no statistically significant difference (absolute difference = 2.42%). In comparison to other algorithms, benchmarking tests demonstrated significant differences, particularly with Wang-CNN and Wang-Bi-LSTM. The bagging classifier, utilizing decision trees, and the boosting classifier, specifically XGBoost, are identified as the two most effective methodologies on dataset DS2, achieving an accuracy rate of 94%. The linear support vector machine (SVM), random forest, and Perez-LSVM models produced some intriguing DS2 results. The accuracy rate for individual learners was 47.75%, whereas the accuracy rate for ensemble learners was 81.5%. The results show a similar pattern for DS3, where the accuracy rate for ensemble learners is 93.5% and that for individual learners it is 80%. In contrast to DS2, Perez-LSVM produced the greatest results on DS3, with a 96% accuracy rate. As a whole, DS4 (DS1, DS2, and DS3) yields the greatest results when tested with random forest (91% accuracy). Ensemble learners averaged 88.16% accuracy, whereas individual learners only managed 85%. Among the algorithms tested, Wang-Bi-LSTM performed the poorest with a 62% accuracy rate.

$$\rho i = \{e - CISc, if\ cosS(xPnew, xPprovious) > \tau c1, otherwise)$$

(14)

$$cosS(xPnew, xPprovious) = \sum Ki = 1xPnewixPpreviousi\sum Ki = 1xPnewi - - - - - - - - \sqrt{\sum Ki = 1xPpreviousi - - - - - - - - - - \sqrt{cosS(xPnew, xPprovious) = \sum Ki = 1xPnewixPpreviousi\sum Ki = 1xPnewi - - - - - - - - \sqrt{\sum Ki = 1xP}}}}$$
$$\Delta X \to i + 1 = (sS \to i + aA \to i + cC \to i + fF \to i + eE \to i) + w\Delta Xi - \to$$

(15)

$$X \to i + 1 = X \to i + \Delta X \to i + 1$$

Table 2 Overall accuracy score

Actual true	True positive (TP)	False negative (FN)
Actual false	False positive (FP)	True negative (TN)

Logistic regression (LR)	0.97	0.91	0.91	0.87
Linear SVM (LSVM)	0.98	0.37	0.53	0.86
Multilayer perceptron	0.98	0.35	0.94	0.9
K-nearest neighbors (KNN)	0.88	0.28	0.82	0.77

International Journal of Environmental Sciences
ISSN: 2229-7359
Vol. 11 No. 21s, 2025
https://theaspd.com/index.php

The algorithms' accuracy, averaged across all four datasets, is presented in Table 2. Upon evaluation, the bagging classifier utilizing decision trees demonstrated the highest accuracy, recorded at 94%. In contrast, the Wang-Bi-LSTM method exhibited the lowest accuracy, measured at 64.25%. Ensemble learners demonstrate an accuracy of 92.25%, whereas individual learners achieve an accuracy of 77.6%. Except for DS2, random forest improved accuracy across the board. But because accuracy score isn't the be-all and end-all of model performance metrics, we supplement it with recall, precision, and F1-score to assess how well learning models are doing.

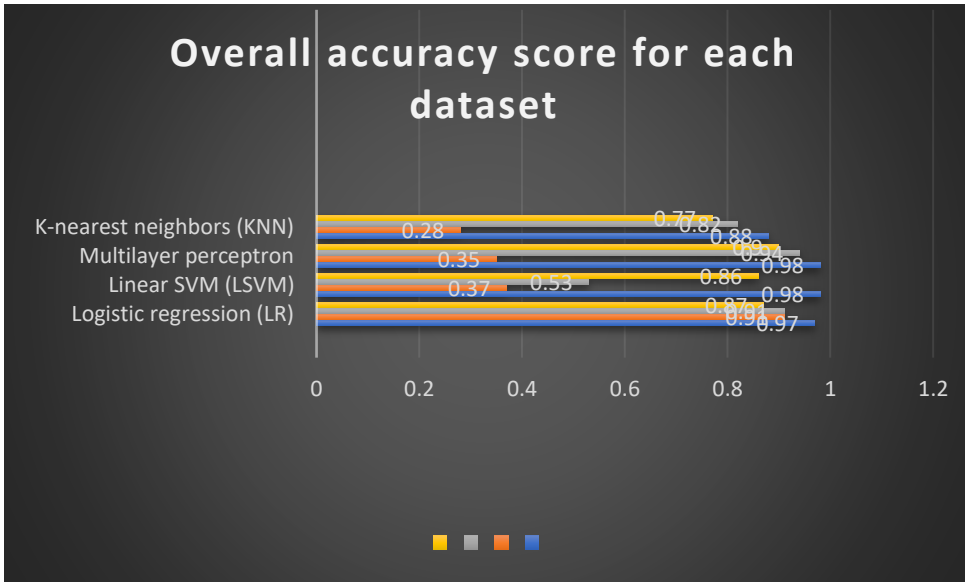

Figure 5 -Average accuracy over all datasets.

Tables 3-5 include details on the recall, precision, and F1 score for each of the four datasets and approaches. Table 3 shows that the boosting classifier, namely XGBoost, had the greatest average accuracy of the examined models. XGBoost, a boosting classifier, achieves an average accuracy of 95.25% across all four datasets. After removing the lowest-scoring dataset, DS2, the average accuracy of the random forest (RF) model climbed to 96.3% across the remaining three datasets, compared to 79.75% accuracy on the original dataset. The boosting classifier, XGBoost, scored 96.3%.

Table 3

	DS1	DS2	DS3	DS4
Logistic regression (LR)	0.98	0.92	0.93	0.88
Linear SVM (LSVM)	0.98	0.31	0.54	0.88
Multilayer perceptron	0.97	0.32	0.93	**0.92**
K-nearest neighbors (KNN)	0.91	0.22	0.85	0.8
Ensemble learners				
Random forest (RF)	0.99	0.3	0.98	0.92
Voting classifier (RF, LR, KNN)	0.96	0.88	0.92	0.86
Voting classifier (LR, LSVM, CART)	0.94	0.86	0.88	0.83
Bagging classifier (decision trees)	0.98	**0.94**	0.93	0.9
Boosting classifier (AdaBoost)	0.98	0.92	0.92	0.86
Boosting classifier (XGBoost)	**0.99**	**0.94**	**0.96**	**0.92**
Benchmark algorithms				
Perez-LSVM	**0.99**	0.79	**0.96**	0.9
Wang-CNN	0.84	0.65	0.48	0.72
Wang-Bi-LSTM	0.92	0.43	0.5	0.65

International Journal of Environmental Sciences
ISSN: 2229-7359
Vol. 11 No. 21s, 2025
https://theaspd.com/index.php

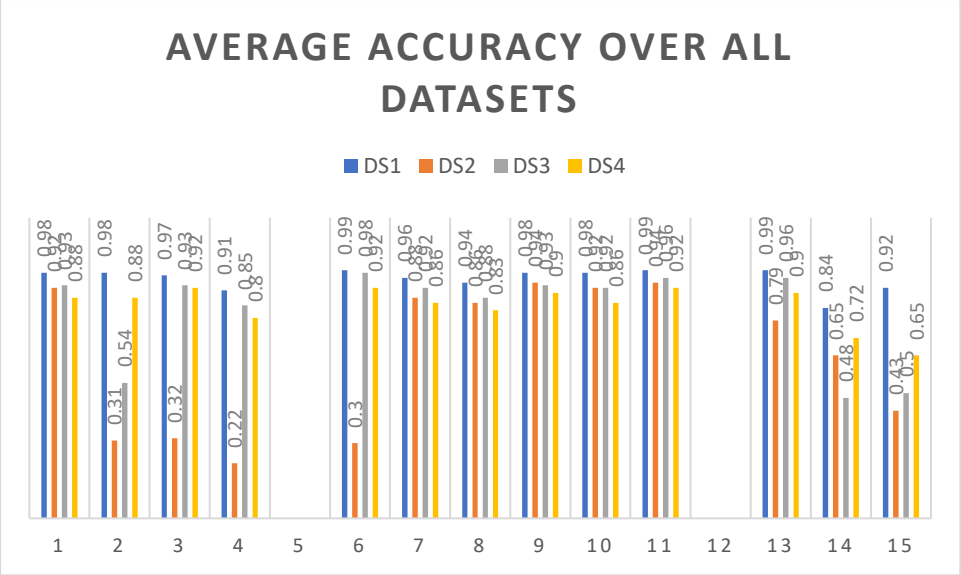

Fig 6

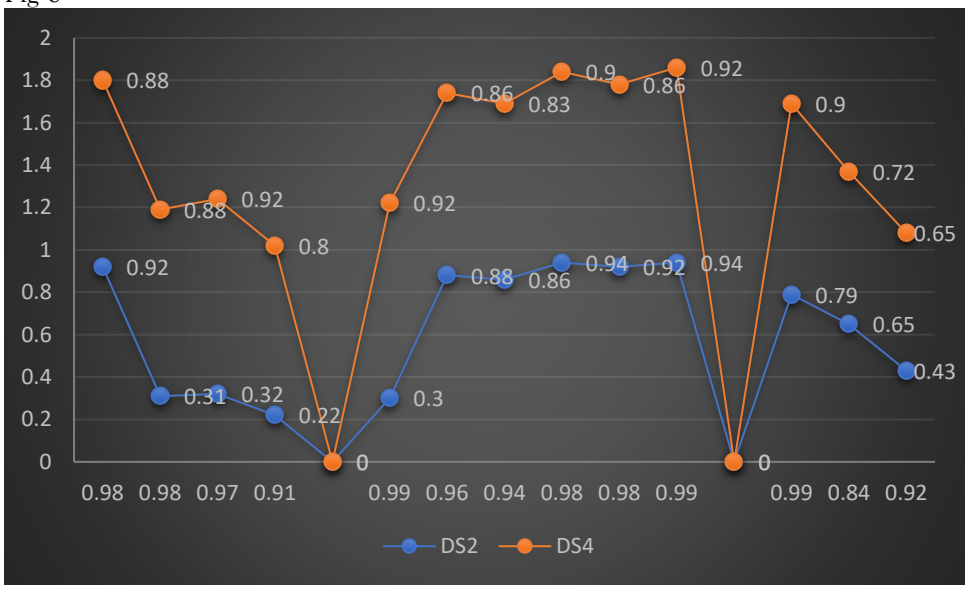

Fig 7

Precision on the 4 datasets.

	DS1	DS2	DS3	DS4
Logistic regression (LR)	0.98	0.9	0.92	0.86
Linear SVM (LSVM)	0.98	0.32	**1**	0.86
Multilayer perceptron	**1**	0.36	0.96	0.88
K-nearest neighbors (KNN)	0.87	0.24	0.81	0.74
Ensemble learners				
Random forest (RF)	**1**	0.34	0.93	**0.91**
Voting classifier (RF, LR, KNN)	0.97	0.89	0.96	0.9
Voting classifier (LR, LSVM, CART)	0.97	0.87	0.96	0.89
Bagging classifier (decision trees)	0.97	**0.95**	0.94	**0.91**
Boosting classifier (AdaBoost)	0.98	0.93	0.92	0.86
Boosting classifier (XGBoost)	0.99	0.94	0.94	0.89
Benchmark algorithms				
Perez-LSVM	0.99	0.81	0.97	**0.91**
Wang-CNN	0.9	0.71	0.29	0.75
Wang-Bi-LSTM	0.78	0.59	0.35	0.61

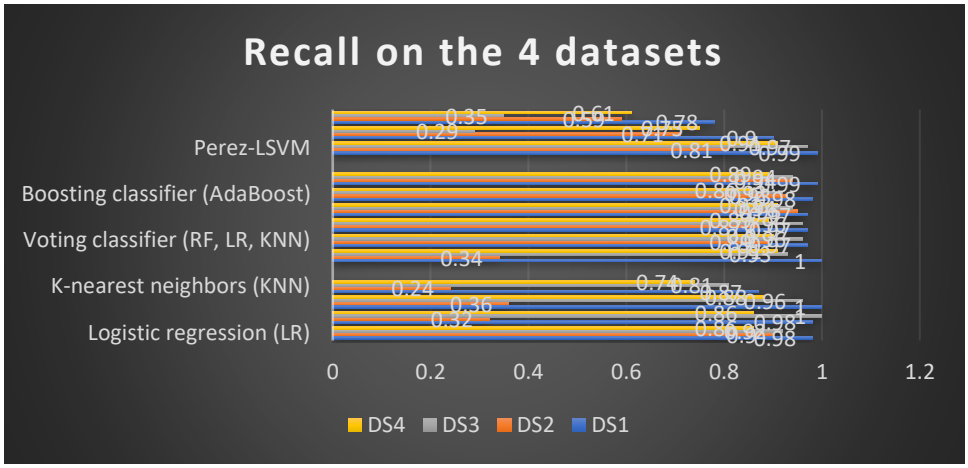

Fig 8

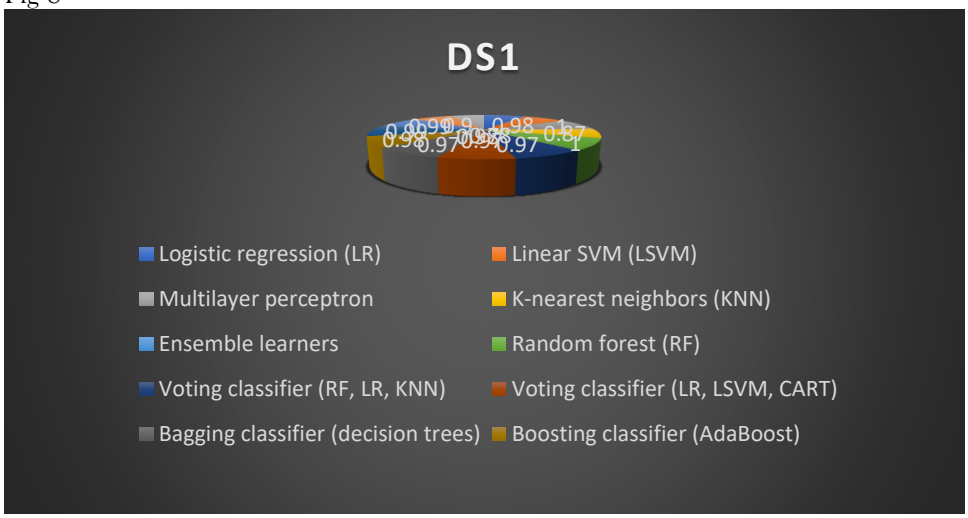

Fig 9

Table 5

	DS1	DS2	DS3	DS4
Logistic regression (LR)	0.98	0.91	0.92	0.87
Linear SVM (LSVM)	0.98	0.32	0.7	0.87
Multilayer perceptron	0.98	0.34	0.95	0.9
K-nearest neighbors (KNN)	0.89	0.23	0.83	0.77
Ensemble learners				
Random forest (RF)	**0.99**	0.32	0.95	**0.91**
Voting classifier (RF, LR, KNN)	0.97	0.88	0.94	0.88
Voting classifier (LR, LSVM, CART)	0.96	0.86	0.92	0.86
Bagging classifier (decision trees)	0.98	**0.94**	0.94	0.9
Boosting classifier (AdaBoost)	0.98	0.92	0.92	0.86
Boosting classifier (XGBoost)	**0.99**	**0.94**	0.95	0.9
Benchmark algorithms				
Perez-LSVM	**0.99**	0.8	**0.96**	0.9
Wang-CNN	0.87	0.67	0.31	0.73
Wang-Bi-LSTM	0.84	0.44	0.35	0.57

International Journal of Environmental Sciences
ISSN: 2229-7359
Vol. 11 No. 21s, 2025
https://theaspd.com/index.php

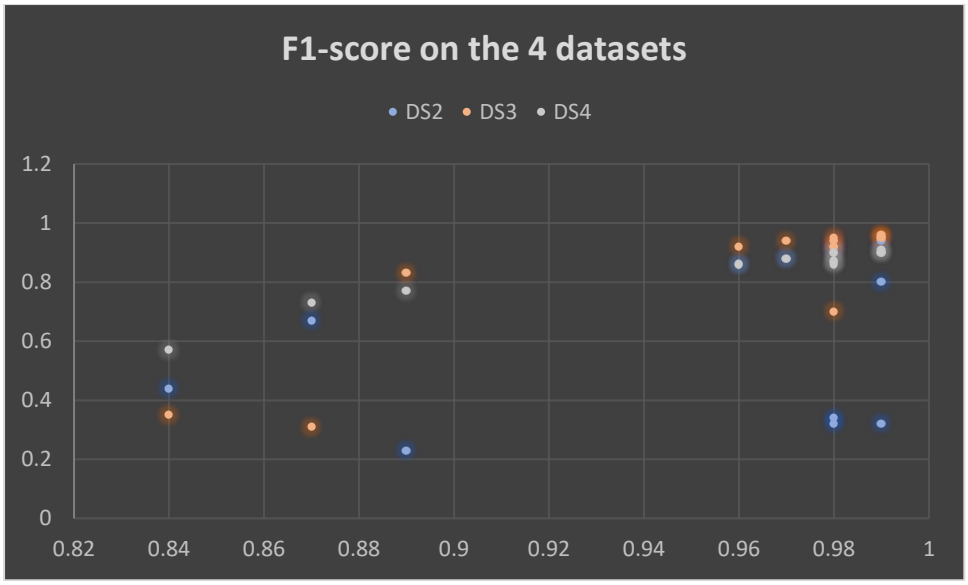

Fig 10
F1-score on the 4 datasets.

Bagging classifiers (decision trees) performed the best with a recall score of 0.942, according to the recall performance criterion. Following on the list is A recall of 0.94 was attained with the XGBoost boosting classifier. Compared to the other benchmark approaches, the Perez-LSVM strategy produced superior results, with a recall score of 0.92. The algorithms maintained their precision-like behavior on F1-score. The boosting classifier (XGBoost) had the highest F1-score of 0.945 out of all the techniques. Bagger classifiers, decision trees, and logistic regression (LR) were the runner-up.

Learning algorithm average performance on all datasets as measured by F1-score, recall, and accuracy is shown in Figure 3. Using alternative measurements does not substantially affect the performance of learning algorithms, with the exception of linear SVM, KNN, Wang-CNN, and Wang-Bi-LSTM.

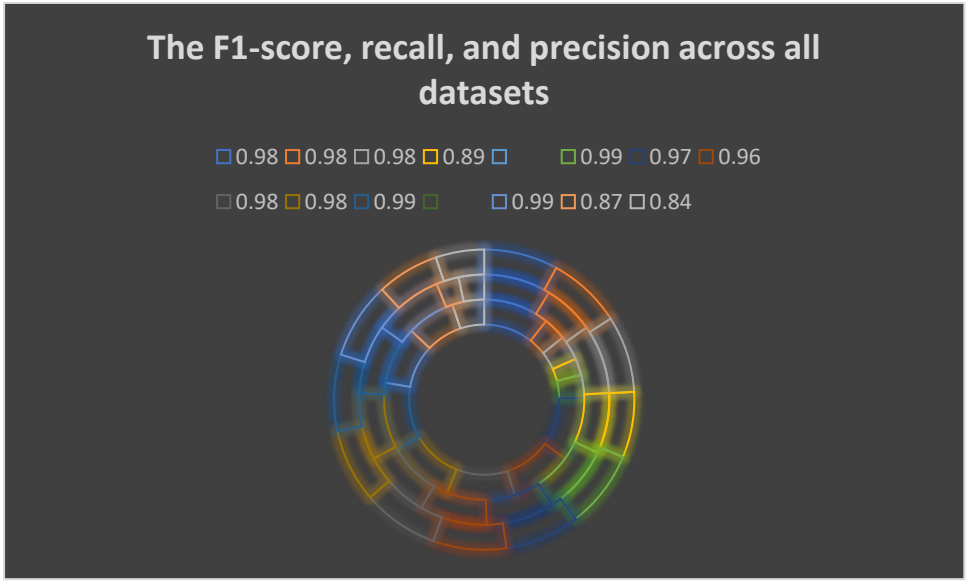

Fig 11
The F1-score, recall, and precision across all datasets.

All performance indicators showed that All of the other learning models were surpassed by the ensemble learner XGBoost. The core concept of XGBoost's improved performance is its ability to detect faults effectively and reduce them with each cycle. If you want to know how XGBoost works, the gist of it is that it uses CART, which combines several weak learners, to give more weight to errors in categorization. Consequently, the model successfully identifies the misclassified points with each iteration, and regularization settings help alleviate the overfitting issue.

Despite its apparent simplicity, logistic regression was able to get the three datasets (DS1, DS2, and DS3) to achieve an average accuracy of about 90%. Using a grid search with several hyperparameters to fine-tune the logistic regression model might explain the high average accuracy. The fact that the logistic regression model achieved a 97% success rate in datasets like DS1—which share comparable authorial styles—is another probable explanation. Logistic regression accuracy falls to 87% on DS4, the combined dataset that incorporates all three sources (and so has more diverse writing styles as well).

4. CONCLUSION

Classifying news stories by hand calls for topic experience in addition to the skill of identifying textual anomalies. The objective of this study was to identify fake news items using ensemble techniques and machine learning models. For the most part, rather than only political news, is covered by the data we retrieved from the Internet, which includes articles from a variety of areas. Finding textual patterns that distinguish false news the project's primary objective is to draw on actual news. Several textual features were extracted from the articles using a LIWC program. The models were subsequently fed the feature set. The learning models were fine-tuned after training in order to achieve optimal accuracy-tuned. A better level of accuracy has been attained by certain models compared to others. In order to compare the algorithms' output, we employed a wide range of performance indicators. Across the board, performance measurements have demonstrated that ensemble learners outperform individual learners.

There are many unsolved issues in the realm of false news detection. For example, determining what elements contribute to the transmission of news is an important first step in limiting the spread of misinformation. Graph theory and machine learning may be used to determine who is disseminating misleading information. The same is true for any future work on real-time video fake news identification.

REFERENCES

1. Douglas, "News consumption and the new electronic media," The International Journal of Press/Politics, vol. 11, no. 1, pp. 29–52, 2006.
2. J. Wong, "Almost all the traffic to fake news sites is from facebook, new data show," 2016.
3. D. M. J. Lazer, M. A. Baum, Y. Benkler et al., "The science of fake news," Science, vol. 359, no. 6380, pp. 1094–1096, 2018.
4. S. A. García, G. G. García, M. S. Prieto, A. J. M. Guerrero, and C. R. Jiménez, "The impact of term fake news on the scientific community scientific performance and mapping in web of science," Social Sciences, vol. 9, no. 5, 2020.
5. D. Holan, 2016 Lie of the Year: Fake News, Politifact, Washington, DC, USA, 2016.
6. S. Kogan, T. J. Moskowitz, and M. Niessner, "Fake News: Evidence from Financial Markets," 2019, https://ssrn.com/abstract=3237763.
7. Robb, "Anatomy of a fake news scandal," Rolling Stone, vol. 1301, pp. 28–33, 2017.
8. J. Soll, "The long and brutal history of fake news," Politico Magazine, vol. 18, no. 12, 2016.
9. J. Hua and R. Shaw, "Corona virus (covid-19) "infodemic" and emerging issues through a data lens: the case of China," International Journal of Environmental Research and Public Health, vol. 17, no. 7, p. 2309, 2020.
10. N. K. Conroy, V. L. Rubin, and Y. Chen, "Automatic deception detection: methods for finding fake news," Proceedings of the Association for Information Science and Technology, vol. 52, no. 1, pp. 1–4, 2015.
11. F. T. Asr and M. Taboada, "Misinfotext: a collection of news articles, with false and true labels," 2019.
12. K. Shu, A. Sliva, S. Wang, J. Tang, and H. Liu, "Fake news detection on social media," ACM SIGKDD Explorations Newsletter, vol. 19, no. 1, pp. 22–36, 2017.
13. S. Vosoughi, D. Roy, and S. Aral, "The spread of true and false news online," Science, vol. 359, no. 6380, pp. 1146–1151, 2018.
14. H. Allcott and M. Gentzkow, "Social media and fake news in the 2016 election," Journal of Economic Perspectives, vol. 31, no. 2, pp. 211–236, 2017.
15. V. L. Rubin, N. Conroy, Y. Chen, and S. Cornwell, "Fake news or truth? using satirical cues to detect potentially misleading news," in Proceedings of the Second Workshop on Computational Approaches to Deception Detection, pp. 7–17, San Diego, CA, USA, 2016.
16. H. Jwa, D. Oh, K. Park, J. M. Kang, and H. Lim, "exBAKE: automatic fake news detection model based on bidirectional encoder representations from transformers (bert)," Applied Sciences, vol. 9, no. 19, 2019.
17. H. Ahmed, I. Traore, and S. Saad, "Detection of online fake news using n-gram analysis and machine learning techniques," in Proceedings of the International Conference on Intelligent, Secure, and Dependable Systems in Distributed and Cloud Environments, pp. 127–138, Springer, Vancouver, Canada, 2017.
18. W. Y. Wang, Liar, Liar Pants on Fire: A New Benchmark Dataset for Fake News Detection, Association for Computational Linguistics, Stroudsburg, PA, USA, 2017.
19. Riedel, I. Augenstein, G. P. Spithourakis, and S. Riedel, "A simple but tough-to-beat baseline for the fake news challenge stance detection task," 2017, https://arxiv.org/abs/1707.03264.
20. N. Ruchansky, S. Seo, and Y. Liu, "Csi: a hybrid deep model for fake news detection," in Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, pp. 797–806, Singapore, 2017.
21. V. Pérez-Rosas, B. Kleinberg, A. Lefevre, and R. Mihalcea, "Automatic detection of fake news," 2017, https://arxiv.org/abs/1708.07104.

22. P. Bühlmann, "Bagging, boosting and ensemble methods," in Handbook of Computational Statistics, pp. 985–1022, Springer, Berlin, Germany, 2012.
23. H. Ahmed, I. Traore, and S. Saad, "Detecting opinion spams and fake news using text classification," Security and Privacy, vol. 1, no. 1, 2018.
24. Kaggle, Fake News, Kaggle, San Francisco, CA, USA, 2018, https://www.kaggle.com/c/fake-news.
25. Kaggle, Fake News Detection, Kaggle, San Francisco, CA, USA, 2018, https://www.kaggle.com/jruvika/fake-news-detection.
26. J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," Journal of Machine Learning Research, vol. 13, pp. 281–305, 2012.
27. T. M. Mitchell, The Discipline of Machine Learning, Carnegie Mellon University, Pittsburgh, PA, USA, 2006.
28. N. Cristianini and J. Shawe-Taylor, An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods, Cambridge University Press, Cambridge, UK, 2000.
29. T. Hofmann, B. Schölkopf, and A. J. Smola, "Kernel methods in machine learning," The Annals of Statistics, vol. 36, no. 3, pp. 1171–1220, 2008.
30. V. Kecman, Support Vector Machines-An Introduction in "Support Vector Machines: Theory and Applications", Springer, New York City, NY, USA, 2005.
31. S. Akhtar, F. Hussain, F. R. Raja et al., "Improving mispronunciation detection of arabic words for non-native learners using deep convolutional neural network features," Electronics, vol. 9, no. 6, 2020.
32. Ruta and B. Gabrys, "Classifier selection for majority voting," Information Fusion, vol. 6, no. 1, pp. 63–81, 2005.
33. B. Gregorutti, B. Michel, and P. Saint-Pierre, "Correlation and variable importance in random forests," Statistics and Computing, vol. 27, no. 3, pp. 659–678, 2017.
34. L. Breiman, J. Friedman, R. Olshen, and C. Stone, Classification and Regression Trees, Springer, Berlin, Germany, 1984.
35. R. E. Schapire, "A brief introduction to boosting," IJCAI, vol. 99, pp. 1401–1406, 1999.
36. M. Dos Santos, R. Sabourin, and P. Maupin, "Overfitting cautious selection of classifier ensembles with genetic algorithms," Information Fusion, vol. 10, no. 2, pp. 150–162, 2009.
37. T. Chen and C. Guestrin, "Xgboost: a scalable tree boosting system," in Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 785–794, San Francisco, CA, USA, 2016.
38. T. Hastie, S. Rosset, J. Zhu, and H. Zou, "Multi-class adaboost," Statistics and its Interface, vol. 2, no. 3, pp. 349–360, 2009.
39. L. Lam and S. Y. Suen, "Application of majority voting to pattern recognition: an analysis of its behavior and performance," IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans, vol. 27, no. 5, pp. 553–568, 1997.