***Article Received:** 25 May 2023 **Revised:** 12 September 2023 **Accepted:** 30 September 2023*

Unified Fake News Detection System (UFNDS) Framework

Mr.Raut Rahul Ganpat
Department of Computer Science and Engineering
Dr. A. P. J. Abdul Kalam University, Indore (M.P.)-452010
Indore, India
braut_rahul@yahoo.co.in

Dr.Sonawane Vijay Ramnath
Department of Computer Science and Engineering
Dr. A. P. J. Abdul Kalam University, Indore (M.P.)-452010
Indore, India
vijaysonawane11@gmail.com

Abstract— The deliberate spread of misleading or inaccurate material pose as authentic news is known as "fake news." Its increasing prevalence calls for the creation of practical strategies to recognize and counteract its negative effects on people and society. Previous methods of identifying fake news depended on linguistic signals and stylistic components. However, these methods faced limitations in terms of their applicability and accuracy. To overcome these constraints, this study proposes the utilization of an extended stacking ensemble classification algorithm (ES-ECA), a machine learning technique designed specifically for detecting fake news. By employing this innovative approach, we aim to surpass the existing barriers and enhance our ability to combat misinformation. The ensemble classifier outperformed the individual classifiers, with an accuracy of 75.18% and an F1-score of 81.81%. These findings imply that the suggested algorithm can be utilized to lessen the negative effects of fake news on society and is efficient at identifying it. The EHT-DL model leverages a multi-step approach to effectively detect fake news. It begins with preprocessing steps such as text normalization, special character handling, stemming, stop word removal, tokenization, and lemmatization. This ensures the dataset is clean and ready for subsequent processing. Feature extraction is performed using TF-IDF, N-grams, and word embeddings scores to capture semantic information and word importance. After that, the dataset is divided into training and testing sets and the deep learning model Dl4jMlpClassifier is used to classify the data. To tackle the drawbacks of existing techniques, the EHT-DL model incorporates efficient hyperparameter tuning. It uses both Grid Search and Random Search methods to optimize the Dl4jMlpClassifier's hyperparameters. By using this method, the model's accuracy and capacity to distinguish between authentic and fraudulent news are both improved. The effectiveness of the EHT-DL model is shown by the experimental findings. Standard assessment criteria including accuracy, precision, recall, and F1-score are used to assess the model. In terms of accuracy and efficiency, comparisons with current methods demonstrate the superiority of the proposed model in identifying bogus news (83.27% accuracy, 80.62% precision, 71.57% recall, and 75.63% f1-score). To increase classification accuracy and resilience, OE-MDL combines the phases of optimized deep learning (ODL) and optimized machine learning (OML). An optimized Multilayer Perceptron serves as the Meta classifier in the OML phase, on top of base classifiers such as optimized RandomForest, optimized J48, optimized SMO, optimized NaiveBayes, and optimized IBk. The experimental findings show that the OE-MDL algorithm performs better than other methods with the maximum recall (85.18%), accuracy (84.27%), precision (74.17%), and F1-Score (79.29%), providing a practical means of halting the spread of false information. The framework for the Unified Fake News Detection System (UFNDS) is revealed. The final result of the research works and demonstrates how the three main stages—Deep Learning and Optimized Ensemble Machine (OE-MDL), Efficient Hyperparameter-Tuned Deep Learning Model (EHT-DL), and Enhanced Stacking Ensemble Classification Algorithm (ES-ECA)—are cohesively incorporated into the UFNDS framework. We examine the UFNDS framework's architectural design and show how flexible it is to the always changing problems associated with fake news identification.

Keywords- TF-IDF, Classification, Stacking, Preprocessing, N-Gram

I. INTRODUCTION

The importance of false information has increased in the current social media and internet era. The ability to distinguish between what is real and what is not has gotten harder and harder. thanks to the effortless creation and dissemination of false information. This predicament is distressing because it has the capacity to adversely affect both individuals and society at large. For example, when people rely on misleading data, it can lead to feelings of fear, confusion, and even physical harm. The dedicated efforts of researchers and practitioners have been focused on the development of techniques aimed at detecting false information, commonly known as fake news, with the intention of tackling this pressing issue head-on [3]. These innovative methods strive to identify deceitful content swiftly and effectively, in order to prevent its dissemination and mitigate any potential harm it may cause. One prevalent approach employed in these endeavors is the utilization of linguistic and stylistic indicators, which serve as valuable cues to distinguish between genuine and fraudulent news [4]. It's crucial to remember that there are restrictions on the use of linguistic and stylistic cues to recognize false information. The fact that the attributes employed could not always be trustworthy markers of fake news is one of the primary difficulties [5]. Using linguistic and stylistic signals alone to distinguish between real articles and fake news can be fairly challenging. especially when some fake news articles are crafted with proper syntax and language. In addition, there is a challenge in applying these characteristics to different sets of data, which in turn makes it complicated to develop effective methods for identifying fake news across various scenarios. As a result, a new and improved classification algorithm called ES-ECA has been introduced as an innovative approach in machine learning in order to identify bogus news. The proposed OE-MDL algorithm aims to effectively detect fake news in order to protect individuals, communities, and societies from the negative effects of misinformation. The algorithm is used in a wide range of fields and applications, such as but not restricted to:

• Social media platforms: Recognising and preventing fake news from spreading there, as false information can spread rapidly and reach a wide audience.
• News organisations: Helping media outlets confirm the veracity of news reports and stop false information from unintentionally spreading.
• Online content platforms: By automatically identifying or eliminating bogus news items from online platforms, these platforms assist with content moderation efforts.
• Fact-checking organisations: Increasing fact-checkers' capacity to spot and disprove false information will help them in their mission to give the public accurate information.

- By utilizing deep learning algorithms and effective hyperparameter tuning strategies, the EHT-DL model seeks to increase the precision and effectiveness of fake news identification. Through the optimization of the deep learning model's hyperparameters, the EHT-DL model achieves higher performance in capturing intricate patterns and distinguishing between authentic and fraudulent news.The following is a summary of this paper's contributions:
- The EHT-DL model, an effective deep learning model with hyperparameter adjustments, is introduced for the purpose of detecting false news.
- An extensive pipeline for preparation that guarantees the dataset is clean and prepared for further processing.
- Utilization of word embeddings, N-grams, and TF-IDF scores for effective feature extraction, capturing semantic information, and word importance.

• Using both Grid Search and Random Search approaches to tune the deep learning model's hyperparameters and improve performance.
• A standard assessment metrics-based experimental evaluation of the EHT-DL model reveals its advantage over current methods in precisely and quickly identifying fake news.

Fundamentally, the UFNDS framework is an intricate combination of the three critical phases, OE-MDL, EHT-DL, and ES-ECA. The goal of this research article is to explain how these stages work together harmoniously to produce a comprehensive and cohesive method for identifying false news.

II. RELATED WORKS

This section examines earlier research and strategies that have been put forth to identify and counteract false information. It talks about the several methods that have been employed to recognize and categorize false information. This section aims to give a thorough overview of the present status of research on fake news detection and to draw attention to any shortcomings or gaps in the methods that are currently in use.

A unique framework called UPFD was presented by Dou et al. [6] that uses integrated content and graph modeling to simultaneously capture many signals from user inclinations. Confirmation bias theory states that people are more likely to spread false information if it supports their current opinions or preferences. The researchers offer a method that has been somewhat limited in previous studies for examining user preference for spotting bogus news. The code and data of the researchers are publicly available as a baseline for GNN-based fake news identification.

Shu et al.'s study [7] looked at the relationship between fake news and social media user profiles. Fake news is designed to look like real news, hence the detection efficacy using news content is frequently inadequate. Consequently, a thorough understanding of the relationship between fake news and social media user profiles is required. The study's findings lay the groundwork for more research into the characteristics of social media user profiles and for improving the ability to identify false information.

The association between fake news and social media user profiles was examined in Shu et al.'s study [7]. Because fake news is meant to resemble legitimate, it is often difficult to identify it using news content. As such, a detailed comprehension of the connection between fake news and social media user profiles is necessary. The study's conclusions set the stage for additional investigation into the traits of social media user profiles and for strengthening the detection of misleading material.

The various uses of sentiment analysis in the identification of false news were examined by Alonso et al. [9]. Fake news producers use a variety of artistic devices to increase the impact of their work, and one of them is arousing viewers' emotions. Consequently, sentiment analysis—which determines the degree and polarity of emotions expressed in a text—is used in fake news detection methods, either as the basis of the system or as an add-on. The writers examine the most important aspects and disadvantages as well as the requirements that must be met in the near future, including multilingualism, transparency, the elimination of biases, and the management of multimedia components.

Sitaula et al. [10] proposed a credibility-based method for detecting bogus news. This study suggests that one important factor in identifying fake news is believability. The researchers show how false news detection systems can perform better when they use the credibility score of a news story, which is calculated using a machine learning-based credibility estimate model. They demonstrate how the inclusion of the believability score greatly improves the capacity of fake news detection systems to distinguish false news items. A sentence-comment co-attention sub-network is presented by

Shu et al. [11] to simultaneously capture explainable top-k check-worthy sentences and user comments for the purpose of detecting fake news. The explainability of fake news identification, according to the researchers, is a vital component that has been overlooked in the study of computational false news detection. Their proposed approach beats seven state-of-the-art fake news detection methods by at least 5.33% in F1-score, and they identify top-k user comments that better explain why a news piece is fake than the baselines by 28.2% in NDCG and 30.7% in Precision. They conduct extensive experiments on real-world datasets to support their findings.

The effectiveness of different algorithms in accurately detecting fake news, along with its true positives and true negatives, is the main topic of Jain et al.'s study [12]. Using two distinct datasets, Kaggle and LIAR, the authors apply count vector and tf-idf vector to four distinct machine learning techniques: Naïve Bayes, Logistic Regression, Random Forest, and XGBoost. According to the findings, XGBoost and count vector produced the best results when it came to predicting false news.

Using language-driven features, Rao et al. [13] provide a Natural Language Processing (NLP) model that extracts grammatical, emotive, syntactic, and readable elements. Since language-level features are highly complicated, the authors take features out of the news material to solve the dimensional problem. They then employ sequential learning using a Dropout layer-based Long Short Term Network Model (LSTM) to improve the detection of bogus news. In comparison to the sequential neural model for false news identification, the suggested Drop out-based LSTM model achieves an accuracy of 95.3% for fake news classification and detection.

The use of data-driven techniques for automatic fake news identification is the main focus of Jwa et al. [14]. The authors examine the link between the news headline and the body content using the Bidirectional Encoder Representations from Transformers model (BERT) to identify fake news. They collect more news information to pre-train their model in order to improve performance. The deep-contextualizing feature of BERT, according to the authors, makes it perfect for this task; it raises the F-score by 0.14 compared to previous state-of-the-art models.

A framework for identifying false news that makes use of data from news articles and social situations is put forth by Raza et al. [15]. Their technology, which is built on Transformer architecture, consists of a decoder that forecasts behavior based on historical observations and an encoder that uses fake news data to acquire valuable representations. To enhance news classification, the authors' approach incorporates many variables from the news content and social surroundings. They also provide a useful labeling method to deal with the issue of label scarcity. Their model can identify bogus news more accurately than the baselines within a few minutes of its transmission (early detection), according to experimental results on real-world data.

III. ES-ECA PHASE

The journey within the UFNDS framework commences with the ES-ECA phase. At this stage, the system employs preprocessing techniques to break down the dataset into individual statements, each ready for analysis. To quantify the importance of features within these statements, the (TF-IDF) measure is employed.

The suggested solution makes use of an improved stacking ensemble technique that combines several base classifiers, such as an upgraded Naive Bayes algorithm, an improved J48 decision tree algorithm, and a refined version of the k-Nearest Neighbors algorithm. Subsequently, a meta-classifier is created using a Random Forest algorithm, boosted through the AdaBoostM1 algorithm. The Random SubSpace algorithm is employed to further enhance performance.

An algorithm for detecting fake news dubbed the Enhanced stacking ensemble classification algorithm (ES-ECA) classifies news stories as "real" or "fake" based on the LIAR dataset. The program cleans and converts the raw data into a format appropriate for machine learning by going through a two-step feature extraction and preprocessing process.

Lowercasing, tokenization, stop word removal, and stemming are used in preprocessing to cut down on the quantity of unique tokens and eliminate words that are frequently used but don't add any value. To extract features from the assertions, "n-grams" must be created. N-grams, which are textual sequences of n words, can be utilized as features in a machine learning model to aid in the classification of the assertions. To determine how significant a term (in this case, an n-gram) is to a document (in this case, a statement) in a corpus (in this case, the LIAR dataset), the method calculates the "term frequency-inverse document frequency" (TF-IDF) of each n-gram in the dataset.

The base classifiers and meta-classifier are then combined using a stacking ensemble method, It uses the training data to train the base classifiers and the base classifier outputs to train the meta-classifier. This approach creates a stacking ensemble by using the RandomSubSpace classifier. Using the stacking method, the base classifiers are integrated into a meta-classifier after being fully trained on the collection of features. The resulting stacked classifier is then used as the input to the RandomSubSpace classifier, which trains an ensemble of classifiers using random subsets of the input features. Finally, the algorithm is used to predict the instances is fake or not in a test dataset.

Algorithm 1: Enhanced stacking ensemble classification algorithm (ES-ECA)

Input	:	LIAR dataset
Output	:	Classification of news articles like "fake" or "real"
		// Preprocessing
1	:	dataset = lowercase(dataset)
2	:	dataset = tokenize(dataset)
3	:	dataset = remove_stop_words(dataset)
4	:	dataset = stem_words(dataset)
		// Feature Extraction
5	:	ngrams = generate_ngrams(dataset)
6	:	tf_idf = compute_tf_idf(ngrams)
		// Divide the dataset into sets for testing and training.
7	:	training_set, testing_set = split_dataset(tf_idf)

8	**:**	write_to_files(training_set, testing_set)
		// Create base classifiers
9	**:**	enhanced_j48_classifier = create_j48_classifier(training_set)
10	**:**	naive_bayes_classifier_(complement) = create_complement_naive_bayes_classifier(training_set)
11	**:**	k_nn_classifier = create_k_nn_classifier(training_set, 5)
12	**:**	k_nn_bagging_classifier = apply_bagging_algorithm(k_nn_classifier)
		// Create meta-classifier
13	**:**	random_forest_classifier = create_random_forest_classifier(500)
14	**:**	ada_boost_classifier = apply_adaboost_algorithm(random_forest_classifier)
		// Combine base classifiers and meta-classifier using stacking ensemble method
15	**:**	base_classifiers = [enhanced_j48_classifier, complement_naive_bayes_classifier, k_nn_bagging_classifier]
16	**:**	meta_classifier = apply_stacking_ensemble_method(base_classifiers, ada_boost_classifier)
17	**:**	stacked_classifier = apply_random_subspace_classifier(meta_classifier)
		// Predict the news instance is fake or real in the testing set
18	**:**	predicted_results = predict_news_instance(testing_set, stacked_classifier)

3.1 Preprocessing:

Preprocessing is the first step in preparing data, during which unprocessed data is cleaned and formatted appropriately for additional study. Before feature extraction and model training, the input dataset of news items is cleaned and transformed using a number of procedures in Algorithm 1's preprocessing stages.

3.2 Feature Extraction:

In machine learning and data analysis, feature extraction is a method used to find and extract a dataset's most crucial characteristics or qualities. Feature extraction aims to convert the unprocessed data into a meaningful set of characteristics that may be utilized for additional research or to construct a prediction model.High-dimensional raw data, or data with numerous variables or features, is frequently utilized in machine learning and data analysis. Because of this, it may be challenging to create correct models or carry out insightful analysis because many of the attributes may be redundant or useless. By determining the key features and lowering the dimensionality of the data, feature extraction can assist in solving this issue [16].

3.2.1 TF-IDF:

Using term frequency-inverse document frequency (TF-IDF) vectors is another popular method in feature extraction for the detection of fake news. This method determines the term frequency (word frequency) of each word in a document and modifies it according to the word's frequency across the full corpus (inverse document frequency).TF-IDF can be used to determine the most significant words or phrases in a document when it comes to the detection of fake news. Words that are uncommon in the entire corpus but frequently appear in false news pieces, for instance, can have a high TF-IDF score, suggesting that they are crucial characteristics for spotting fake news.

Equation displays the formula for determining a term's (word's) TF-IDF scores within a document. (1).

$$\mathbf{TF\_IDF = (TF * IDF)} \qquad (1)$$

Where:

The number of times a term appears in a document is its term frequency, or TF.

IDF is calculated as:

$$\mathbf{IDF = \log(N / n)} \qquad (2)$$

Where:

The number of documents in the corpus that include the phrase is n, and the total number of documents in the corpus is N.

Because it can increase the study's precision and effectiveness, feature extraction is a critical stage in many machine learning and data analysis projects. Feature extraction can assist in removing noise and unnecessary information from the data by decreasing its dimensionality and finding its most significant features. This can enhance the performance of machine learning models and facilitate the discovery of patterns and insights within the data [17].

Algorithm 2: Feature Extraction		
Input	**:**	Preprocessed dataset of news articles
Output	**:**	Extracted features from the dataset
1	**:**	Convert the preprocessed text into numerical vectors using pre-trained word embeddings.
2	**:**	Extract N-grams from the preprocessed text to capture contiguous sequences of n words (e.g., bigrams, trigrams).
3	**:**	For every word in the dataset, calculate its Term Frequency-Inverse Document Frequency (TF-IDF) scores to determine its relative relevance.
4	**:**	Combine the word embeddings, N-grams, and TF-IDF scores to create a comprehensive feature representation for each news article.
5	**:**	To make sure the extracted features have a same scale, normalize them.
6	**:**	Output the extracted features for classification.

3.3 Results and Discussions of the Experiment:

In this part, the Liar dataset is used to assess the ES-ECA algorithm's performance in detecting fake news. Politicians' claims are included in the publicly accessible Liar dataset, which is categorized as truthful, largely true, half true, barely true, false, and trousers on fire. The dataset contains

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
***Article Received:** 25 May 2023 **Revised:** 12 September 2023 **Accepted:** 30 September 2023*

metadata aspects like the speaker's party affiliation and work title in addition to textual features like the statement itself. The Liar dataset is used by the Java implementation of the ES-ECA technique to assess the ensemble's performance. Four assessment metrics—accuracy, precision, recall, and F1-score—are used to assess the algorithm's performance. The percentage of accurate forecasts (true positives and true negatives) relative to the total.

Accuracy = (The sum of the true positives and true negatives, divided by the sum of the false positives, false negatives, and true positives.) **(1)**

The percentage of true positives among all positive forecasts is quantified by precision. The computation is provided by:

Precision = false positives plus true positives, divided by true positives **(2)**

The percentage of true positives relative to all actual positives in the dataset is called recall. It has the following definition:

Recall = true positives divided by (false negatives plus true positives) **(3)**

A balanced measure between precision and recall is provided by the F1-score, which is a harmonic mean of the two. It is computed as follows:

F1-score = 2 * recall * precision / (recall + precision) **(4)**

The assessment metrics offer a numerical gauge of the algorithm's efficacy in identifying false information. Using the same parameters for comparison, the performance of each participant classifier is also assessed independently. Table 1 presents a comparison of classifier performance based on f1-score, accuracy, precision, and recall.

Table 1: Classifier Performance Comparison Using F1-Score, Accuracy, Precision, and Recall Metrics

Metrics	**J48**	**NB**	**KNN**	**RF**	**ES-ECA**
Accuracy	20.07	20.68	17.43	21.29	75.18
Precision	20.60	20.93	18.70	21.29	75.98
Recall	20.07	20.68	17.43	21.29	88.62
F1-Score	16.85	20.30	11.59	16.80	81.81

Furthermore, Figure 1 shows the pictorial diagram of the performance comparison of five different classifiers, namely J48, NB, KNN, RF, and ES-ECA, on a dataset.

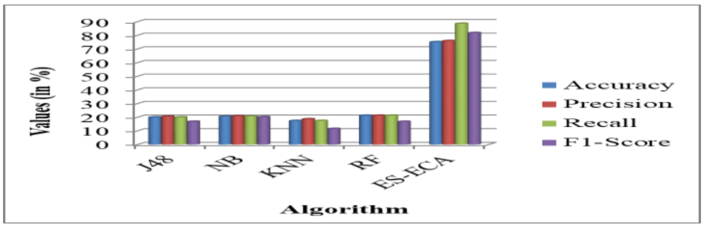


Figure 1: Classifier Performance Comparison Using F1-Score, Accuracy, Precision, and Recall Metrics

As seen in Figure 1, the ES-ECA classifier achieved an accuracy of 75.18%, precision of 75.98%, recall of 88.62%, and F1-score of 81.81%, outperforming the other classifiers in every evaluation metric. The other classifiers, on the other hand, obtained poorer recall, accuracy, precision, and F1-Scores; KNN scored the lowest across the board in all evaluation criteria. According to these findings, the ES-ECA classifier is the most suitable option for the dataset at hand and is capable of accurately identifying bogus news.

IV. EHT-DL PHASE

Following the ES-ECA phase, the focus shifts to the EHT-DL phase. This phase leverages a multi-step approach to detect fake news effectively. It initiates with preprocessing steps that include text normalization, handling special characters, tokenization, stop word removal, stemming, and lemmatization, ensuring that the dataset is pristine for subsequent processing.

Feature extraction in EHT-DL incorporates word embeddings, N-grams, and TF-IDF scores to capture semantic information and word importance. The robust Dl4jMlpClassifier deep learning model then takes the lead in classification after the dataset is carefully split into training and testing sets.

Effective hyperparameter adjustment is integrated into the EHT-DL phase to further improve the system's precision. The Dl4jMlpClassifier's hyperparameters are carefully adjusted using Grid Search and Random Search approaches to find the best combinations that maximize performance. This method improves the phase's precision and capacity to distinguish between real and fraudulent news.

The EHT-DL model follows a series of steps to detect fake news effectively:

- **Preprocessing:** The model begins by preprocessing the dataset of news articles. It converts the text to lowercase to ensure consistency. Special characters and punctuation are handled appropriately. The dataset is then tokenized, splitting it into discrete terms or symbols. Stop phrases, such as common words with little semantic meaning, are removed. Words are stemmed and lemmatized to reduce variations and unify related terms. These preprocessing steps ensure the dataset is clean and prepared for further processing.
- **Feature Extraction:** The EHT-DL model uses a number of feature extraction methods. The preprocessed text is transformed into numerical vectors that represent word relationships and semantic information using word embeddings. In order to capture local word contexts, N-grams—contiguous sequences of n words, such as bigrams and trigrams—are retrieved as supplementary features. Based on a word's frequency and rarity, TF-IDF (Term Frequency-Inverse Document Frequency) scores are calculated to determine how important a word is in the dataset. Through the capture of both local and global information, these feature extraction approaches improve the representation of text data.
- **Classification:** To facilitate model training and assessment, the dataset is split into training and testing sets. To identify

news items as genuine or fraudulent, the EHT-DL model applies a deep learning model called Dl4jMlpClassifier, which was created especially for classification tasks. The parameters and architecture of the model are optimized for efficient classification.

- **Hyperparameter Tuning:** Effective hyperparameter adjustment is incorporated into the EHT-DL model to maximize the performance of the Dl4jMlpClassifier and get around the drawbacks of previous methods. To repeatedly investigate various combinations of hyperparameters, such learning rate and the number of hidden units, it employs Grid Search and Random Search algorithms. The optimal choices that result in higher performance are found by assessing the model's performance with various hyperparameter setups.
- **Model Training and Deployment:** By using the training data, the EHT-DL model teaches the Dl4jMlpClassifier the patterns and traits that distinguish between fake and authentic news. After being trained, the model can be used to determine if new articles are fraudulent or not.
- **Model Evaluation:** Accuracy, precision, recall, and F1-score are among the common assessment measures used to assess the trained EHT-DL model. These metrics shed light on the model's effectiveness and precision in distinguishing between fake and authentic news.
  The EHT-DL model can be applied in various domains and platforms where the detection of fake news is crucial. It can be utilized in social media platforms to identify and mitigate the spread of misinformation. Online news portals can integrate the EHT-DL model to provide more reliable news sources. Fact-checking organizations can leverage the model's capabilities to enhance their verification processes. Essentially, any system or platform that deals with the dissemination of information can benefit from the EHT-DL model for effective fake news detection.

When it's necessary to precisely identify and distinguish between authentic and fraudulent news, the EHT-DL model is employed. It can be employed in real-time scenarios where the timely identification of misinformation is critical. For example, during election campaigns, the EHT-DL model can help identify and counteract fake news that might influence voters. In crises, such as natural disasters or public health emergencies, the model can help separate reliable information from false rumors, preventing panic and facilitating effective responses. The EHT-DL model is also valuable in situations when the dissemination of false information can have serious negative effects on society, such as controversial events or sensitive political developments.

Overall, the novelty of the EHT-DL model lies in its integration of deep learning, efficient hyperparameter tuning, multi-step preprocessing, a combination of feature extraction techniques, and its focus specifically on the challenge of identifying false news. These elements contribute to its unique approach and advancements in tackling the challenges of identifying and mitigating the dissemination of false information. Algorithm 3 explains the detailed proposal for the EHT-DL model.

Algorithm 3: EHT-DL: An efficient hyperparameter-tuned deep learning model for identifying fraudulent news.

Input	:	Dataset of news articles (with labels indicating whether they are fake or not) Hyperparameters (e.g., learning rate, numHiddenUnits)
Output	:	The deep learning model that has been taught to detect bogus news
		/* Preprocessing */
1	:	Convert the dataset to lowercase.
2	:	Handle special characters and punctuation.
3	:	Tokenize the dataset using whitespace and regular expressions.
4	:	Remove stop words from the dataset.
5	:	Stem words in the dataset.
6	:	Perform lemmatization on the dataset.
7	:	Utilize parallel processing to optimize efficiency.
		/* Feature Extraction */
8	:	**Word Embeddings:** Convert the preprocessed text into numerical vectors using pre-trained word embeddings.
9	:	**N-grams:** Extract contiguous sequences of n words as features (e.g., bigrams, trigrams).
10	:	**TF-IDF:** To determine a word's importance, calculate its Term Frequency-Inverse Document Frequency (TF-IDF) scores.
		/* Classification */
11	:	Divide the dataset into training and testing sets (train/test split).
12	:	**Model Architecture:** Use the Dl4jMlpClassifier deep learning model for classification.
13	:	**Hyperparameter Tuning:** Make use of methods like Grid Search and Random Search to optimize the Dl4jMlpClassifier model's hyperparameters. Set base classifier options. **Apply Random Search:** Randomly set hyperparameters (e.g., learningRate and numHiddenUnits). Modify options with new hyperparameters. Create and evaluate Dl4jMlpClassifier. Update best options if performance improves. **Apply Grid Search:** Define a set of learning rate values (e.g., [0.01, 0.05, 0.1]) and numHiddenUnits values (e.g., [10, 50, 100]). Modify the previous best options with new hyperparameters. Create and evaluate Dl4jMlpClassifier.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
***Article Received:** 25 May 2023 **Revised:** 12 September 2023 **Accepted:** 30 September 2023*

		Update best options if performance improves. The tuned hyperparameters are the best choices that came from the hyperparameter tuning procedure.Create Dl4jMlpClassifier with these final best options.
14	**:**	**Model Training:** Use the training set of data to train the Dl4jMlpClassifier model.
15	**:**	**Model Deployment:** Use the trained model to determine whether or not new articles (testing data) are fraudulent.
16	**:**	**Model Evaluation:** Assess the trained model using measures like accuracy, precision, recall, and F1-score on the testing data.

4.1 The training, implementation, and assessment of the Dl4jMlpClassifier model:

The processes of training a deep learning model named Dl4jMlpClassifier, deploying the trained model to categorize new articles as fake or not, and assessing the model's performance using a variety of metrics are referred to as Dl4jMlpClassifier model training, deployment, and assessment in EHT-DL.To create a fake news detection system that works in EHT-DL, Dl4jMlpClassifier model training, deployment, and assessment are required.

Accurate classification of news articles is made possible by the training phase, which teaches the model to recognize patterns and correlations in the labeled dataset. In order to apply the trained model to unknown data for real-world classification tasks, deployment is required. Evaluation aids in evaluating the model's efficacy and performance in differentiating between false and authentic news.

In EHT-DL, Dl4jMlpClassifier model training, deployment, and evaluation work as follows:

- Training and testing sets are separated from the labeled dataset.
- The architecture and parameters of the Dl4jMlpClassifier deep learning model are specified, including the number of layers, activation functions, regularization strategies, learning rate, and numHiddenUnits.
- The model is initialized with the defined architecture and parameters.
- The extracted features and labels are used to train the model on the training set.
- To maximize the model's performance, hyperparameter tuning is carried out. This entails using methods like Grid Search and Random Search to modify hyperparameters, including learning rate and numHiddenUnits. In order to determine the optimal configuration that maximizes the performance of the model, it methodically investigates various combinations of hyperparameters.
- The trained model is validated using the training set to monitor performance and prevent overfitting.
- Until the model performs satisfactorily on the training set, validation and hyperparameter tuning are repeated.
- The trained model that has been hyperparameter-tuned is used to identify bogus news articles from unseen datasets.

The accuracy, precision, recall, and F1-score are among the common assessment measures used to assess the model's performance.Following feature extraction, Dl4jMlpClassifier model training, deployment, and assessment are commonly employed in EHT-DL. The Dl4jMlpClassifier model is trained on labeled training data, used for real-world classification tasks, and its performance is assessed once the features have been retrieved from the preprocessed text data.
Advantages of Dl4jMlpClassifier model training, deployment, and evaluation used in EHT-DL include:

- **Accuracy:** Because it is a deep learning model, the Dl4jMlpClassifier model has the ability to identify intricate patterns and relationships in the data, which could result in a greater detection accuracy for fake news.
- **Flexibility:** Better adaptation to the issue at hand is made possible by the ability to modify the model's architecture and parameters in accordance with the particular needs and features of the dataset.
- **Evaluation Metrics:** In order to enable comparisons and benchmarking against other models, the evaluation process gives objective metrics, such as accuracy, precision, recall, and F1-score, to assess the performance of the trained model.
- **Scalability:** Large datasets can be handled using deep learning models like Dl4jMlpClassifier, which can also be scaled to handle higher data volumes when needed.
- **Hyperparameter Tuning:** The Dl4jMlpClassifier model can be tuned for improved performance by maximizing the model's accuracy and capacity for generalization by determining the best possible combination of hyperparameters.

The Dl4jMlpClassifier model refers to a specific type of deep learning model used in the EHT-DL algorithm for fake news detection. Dl4jMlpClassifier stands for "DeepLearning4j Multi-Layer Perceptron Classifier," where DeepLearning4j (DL4j) is a deep learning library for Java and One kind of neural network architecture is the Multi-Layer Perceptron (MLP).
The MLP is a feedforward neural network model made up of artificial neurons or units, which are numerous layers of interconnected nodes. An input layer, one or more hidden layers, and an output layer are usually its components. Every neuron in one layer is linked to the neurons in the layer below it. Weights are used to represent the connections between neurons and are modified during training to uncover underlying patterns in the data.
The Dl4jMlpClassifier model in EHT-DL utilizes the MLP architecture for classification tasks, specifically for detecting fake news. It takes the extracted features from the preprocessed text data as input and learns to classify articles as either fake or not. The model's architecture, During the training phase, certain

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
***Article Received:** 25 May 2023 **Revised:** 12 September 2023 **Accepted:** 30 September 2023*

parameters are defined, such as the total number of layers and the number of units in each layer (numHiddenUnits).

To train the Dl4jMlpClassifier model, the labeled dataset is used, where the features extracted from the preprocessed text are paired with their corresponding labels indicating whether they are fake or not. To decrease the discrepancy between the model's predicted outputs and the true labels, an optimization approach, such as stochastic gradient descent (SGD), is used throughout the training process.

Throughout the training phase, the Dl4jMlpClassifier model modifies the weights of its neurons according to the training cases, progressively enhancing its capacity for precise prediction-making. Using validation approaches, the model's performance is continuously assessed to track its development and avoid overfitting.

The model can be used after it has been trained. to classify new articles as fake or not by feeding them through the network and obtaining the predicted output. The deployed model is capable of handling unseen data and making predictions in real time.

All things considered, the Dl4jMlpClassifier model is an effective deep learning model that uses the MLP architecture to help the EHT-DL algorithm detect bogus news. In order to accurately forecast whether news stories are real, it learns to identify patterns and relationships in the extracted features. The training, deployment, and assessment of the Dl4jMlpClassifier model are depicted in Algorithm 4.

Algorithm 4: Dl4jMlpClassifier model training, deployment, and evaluation

Input	**:**	Extracted features and labeled dataset (training data)
Output	**:**	Trained classification model, fake news detection, and Evaluation metrics
1	**:**	Divide the labeled dataset into sets for testing and training.
2	**:**	Use the Dl4jMlpClassifier deep learning model for classification.
3	**:**	Specify the Dl4jMlpClassifier model's architecture and parameters, such as the number of layers, activation functions, regularization strategies, learning rate, and numHiddenUnits.
4	**:**	Initialize the Dl4jMlpClassifier model with the defined architecture and parameters.
5	**:**	Utilizing the extracted features and matching labels, train the Dl4jMlpClassifier model on the training set.
6	**:**	Optimize the Dl4jMlpClassifier model's performance by adjusting the hyperparameters, such as learning rate, and numHiddenUnits.
7	**:**	Validate the trained model using the training set to monitor its performance and prevent overfitting.
8	**:**	Iterate steps 5-7 until the model achieves satisfactory performance on the training set.
10	**:**	Deploy the hyperparameter-tuned trained Dl4jMlpClassifier model to classify the news articles in the unseen dataset as either real or fake.
11	**:**	Use common assessment metrics, like accuracy, precision, recall, and F1-score, to assess the model's performance.

Overall, Dl4jMlpClassifier model training, deployment, and evaluation are integral parts of EHT-DL for identifying false news. They involve training a deep learning model, deploying it for real-world classification, and evaluating its performance. These processes enable the model to learn from labeled data, make predictions on unseen articles, and assess its effectiveness using evaluation metrics. The advantages include accuracy, flexibility, evaluation metrics, scalability, and the capability to perform hyperparameter tuning, which enhances the model's performance and adaptability. Dl4jMlpClassifier serves as a valuable component of the overall EHT-DL system.

4.3 Advantages of the EHT-DL model:

The EHT-DL model for fake news detection offers several key advantages in combating the spread of misinformation. Through the application of deep learning methodologies and effective hyperparameter adjustments, the EHT-DL model enhances accuracy, efficiency, and adaptability. From improved accuracy and efficient feature extraction to automated hyperparameter tuning and robustness to evolving fake news techniques, the EHT-DL model provides a comprehensive solution for identifying and combating fake news in the modern information age. The key advantages are:

1. **Improved accuracy:** The EHT-DL model reduces false positives and false negatives by improving the accuracy of fake news detection through the use of deep learning techniques and hyperparameter tuning.
2. **Efficient feature extraction:** The model employs word embeddings, N-grams, and TF-IDF scores to capture semantic information and word importance, enabling it to effectively represent the text data.
3. **Hyperparameter optimization:** In order to find the optimal combinations of hyperparameters and enhance performance, the EHT-DL model uses hyperparameter tuning approaches including Grid Search and Random Search.
4. **Flexibility:** The model's adaptability to diverse settings and false news kinds stems from its ability to be applied to a wide range of datasets and domains.
5. **Preprocessing steps:** Essential preparation tasks like text normalization, stop word removal, stemming, and lemmatization are carried out by the EHT-DL model to enhance the quality of the data and the classification process that follows.
6. **Scalability:** The model can handle large datasets and can be parallelized to optimize efficiency by leveraging parallel processing techniques.
7. **Generalization:** The EHT-DL model has the potential to generalize well to unseen data by learning complex patterns and representations from the training set.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
Article Received: *25 May 2023* ***Revised:*** *12 September 2023* ***Accepted:*** *30 September 2023*

8. **Automated hyperparameter tuning:** The incorporation of hyperparameter tuning eliminates the need for manual parameter selection, saving time and effort in finding optimal configurations.
9. **Comparative analysis:** The model allows for easy comparison with existing techniques, providing insights into its superiority and advantages over other approaches.
10. **Real-time deployment:** Once trained, the EHT-DL model can be deployed for real-time classification of news articles, enabling quick detecting and reducing false information in situations where time is of the essence.

4.4 Results and Discussions of the Experiment:

This section uses the Liar dataset to assess the EHT-DL model's efficacy in detecting fake news. Politicians' claims are categorized as truthful, mostly true, half true, barely true, false, and trousers on fire in the publicly accessible Liar dataset. It contains both textual elements, such as the statement itself, and metadata elements, like the party affiliation and job title of the speaker. The performance of the Java-implemented EHT-DL model is assessed using the Liar dataset.

The efficacy of the algorithm is evaluated using four metrics: accuracy, precision, recall, and F1-score for each.The ratio of accurate predictions—true positives and true negatives—to the total number of forecasts is known as accuracy. It is computed with the following formula:

Accuracy = (The sum of the true positives and true negatives, divided by the sum of the false positives, false negatives, and true positives.) **(1)**

The percentage of true positives among all positive forecasts is quantified by precision. The computation is provided by:

Precision = false positives plus true positives, divided by true positives **(2)**

The percentage of true positives relative to all actual positives in the dataset is called recall. It has the following definition:

Recall = true positives divided by (false negatives plus true positives) **(3)**

A balanced measure between precision and recall is provided by the F1-score, which is a harmonic mean of the two. It is computed as follows:

F1-score = 2 * recall * precision / (recall + precision) **(4)**

These assessment measures provide numerical information about how well the program detects false news. Furthermore, using the same measures for comparison, each participant classifier's performance is assessed independently. A comparison of the classifier performance measured by accuracy, precision, recall, and F1-score is shown in Table 2.

Table 2: Comparing Classifier Performance Using F1-Score, Accuracy, Precision, and Recall Metrics

Metrics	J48	NB	KNN	RF	Dl4jMlpClassifier	EHT-DL model
Accuracy	20.02	22.37	18.94	20.54	21.29	83.27
Precision	20	21.13	26.17	25.7	20.8	80.62
Recall	20.02	22.37	18.94	20.54	21.29	71.57
F1-Score	16.93	19.83	13.01	17.03	20.07	75.83

Furthermore, Figure 2 shows the pictorial diagram of the performance comparison of six different classifiers, namely J48, NB, KNN, RF, Dl4jMlpClassifier, and EHT-DL model, on a dataset.

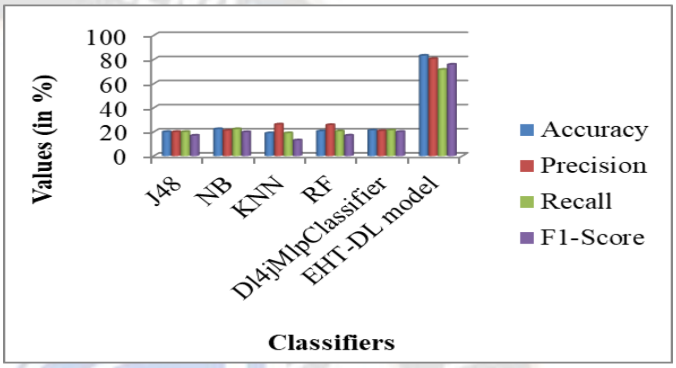


Figure 2: Comparing Classifier Performance Using F1-Score, Accuracy, Precision, and Recall Metrics

The EHT-DL model stands out as the best-performing model among the classifiers.

Accuracy: With an accuracy of 83.27%, the EHT-DL model outperforms other classifiers with accuracies ranging from 18.94% to 22.37%). Accuracy quantifies the proportion of correctly identified cases and the overall correctness of the model's predictions. The EHT-DL model appears to perform better than the other classifiers in accurately differentiating between fake and authentic news, based on its higher accuracy.

Precision: Out of all cases projected as positive, precision measures the percentage of accurately predicted positive instances (false news in this context). The precision of 80.62% achieved by the EHT-DL model is higher than that of J48, NB, RF, and Dl4jMlpClassifier. This suggests that the EHT-DL model is better able to recognize and categorize instances of fake news.

Recall: The percentage of accurately anticipated positive instances among all actual positive instances is called recall, which is often referred to as sensitivity. With a recall of 71.57%, the EHT-DL model outperforms the recalls of J48, NB, RF, and Dl4jMlpClassifier. This suggests that the likelihood of false negatives can be decreased since the EHT-DL model can efficiently catch a larger number of real fake news events.

F1-Score: The F1-score offers a fair assessment of a model's performance since it is the harmonic mean of precision and recall. The F1-score of 75.83% is attained by the EHT-DL model, surpassing that of J48, NB, RF, and Dl4jMlpClassifier. This suggests that the EHT-DL model performs better overall

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
***Article Received:** 25 May 2023 **Revised:** 12 September 2023 **Accepted:** 30 September 2023*

in the detection of bogus news by achieving a better trade-off between precision and recall.

The EHT-DL model performs best because it incorporates an efficient hyperparameter tuning approach, utilizing both Grid Search and Random Search techniques. By systematically exploring various combinations of hyperparameters, the EHT-DL model identifies the optimal settings that maximize its performance. This procedure aids the model's ability to recognize intricate patterns and distinguish between instances of legitimate and fraudulent news.

Furthermore, to guarantee that the dataset is appropriately ready for classification, the EHT-DL model makes use of a multi-step methodology that includes preprocessing processes (such as text normalization, tokenization, and feature extraction utilizing word embeddings, N-grams, and TF-IDF scores). By taking a thorough approach, the model is able to extract word importance and semantic information, which improves feature representation and improves the identification of fake news.

The experimental findings show that in terms of accuracy, precision, recall, and F1 score, the EHT-DL model performs better than the other classifiers. Its superior performance can be attributed to the effective hyperparameter tuning process and the incorporation of preprocessing steps that improve feature extraction and representation. Overall, the EHT-DL model exhibits effectiveness in combating the challenges of fake news detection, outperforming existing techniques.

V. OE-MDL PHASE

Having laid the foundation with ES-ECA and refined it with EHT-DL, the UFNDS framework proceeds to the OE-MDL phase. Here, the limitations of existing techniques are addressed with a series of enhancements.

Lowercase conversion, tokenization, stop word removal, word stemming, lemmatization, and spell checking are examples of preprocessing techniques. Furthermore, the ability to generate n-grams and compute term frequency-inverse document frequency (TF-IDF) scores is utilized to collect subtle signals that differentiate authentic news from fraudulent news.To achieve better classification accuracy and resilience, OE-MDL takes one step further by combining the phases of optimized deep learning (ODL) and optimized machine learning (OML).

In OML, an optimized Multilayer Perceptron serves as the Meta classifier, and base classifiers including optimized RandomForest, optimized J48, optimized SMO, optimized NaiveBayes, and optimized IBk are layered beside it. An AdaBoostM1 boosting classifier uses this stacked classifier as its classifier, while a bagging classifier uses it as its base.

In ODL, a Dl4jMlpClassifier is employed as the base for a bagging classifier, which becomes the classifier for an AdaBoostM1 boosting classifier. To combine OML and ODL classifiers, a blending classifier with weighted voting is employed to make predictions on the training set. The trained blending classifier plays a pivotal role in determining the authenticity of news articles within the testing set.

The OE-MDL algorithm, through the use of optimised ensemble techniques that combine machine learning and deep learning methodologies, offers a thorough and efficient methodology for fake news detection. The suggested OE-MDL algorithm is described in Algorithm 4.

Algorithm 4: (OE-MDL) for Recognizing False News

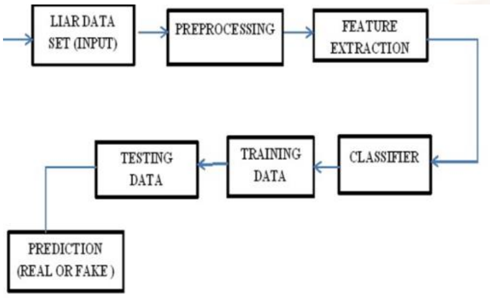
Input	:	LIAR dataset
Output	:	News articles are categorized as "true," "half-true," "false," "barely true," "pants-fire," and "mostly true."
		// Preprocessing Phase
1	:	Convert dataset to lowercase
2	:	Set the dataset to tokens.
3	:	Eliminate stop words from the dataset.
4	:	Stem words in the dataset
5	:	Perform lemmatization on the dataset
6	:	Apply spell check and correction to the dataset
		// Feature Extraction Phase
7	:	Generate n-grams from the dataset
8	:	Determine the n-grams' word frequency-inverse document frequency (TF-IDF).
		// Split dataset into training and testing sets Phase
9	:	Divide the TF-IDF dataset into sets for testing and training.
10	:	Create files with the training and testing sets.
		/* Optimized Machine Learning Phase */
11	:	**Stacking Classifier:** • Optimized Random Forest, Optimized J48, Optimized SMO, Optimized Naive Bayes, and Optimized IBk are used as base classifiers. • Optimized Multilayer Perceptron is used as the meta classifier. • Combine the base classifiers and meta classifier in the stacking classifier.
12	:	**Bagging Classifier 1:** • Set the stacking classifier as the classifier for the bagging classifier.
13	:	**Boosting Classifier 1:** • Set the bagging classifier as the classifier for the AdaBoostM1 classifier. • This ensemble classifier is referred to as the Optimized Machine Learning (OML) classifier.
		/* Optimized Deep Learning Phase */
14	:	**Bagging Classifier 2:** Set the Dl4jMlpClassifier as the classifier for the bagging classifier.
15	:	**Boosting Classifier 2:** • Set the bagging classifier as the classifier for the AdaBoostM1 classifier. • This ensemble classifier is referred to as the Optimized Deep Learning (ODL) classifier.
		/* Optimized Ensemble Machine and Deep Learning Phase*/

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
***Article Received:** 25 May 2023 **Revised:** 12 September 2023 **Accepted:** 30 September 2023*

16 : **Blending Classifier:**
- Combine the OML classifier and ODL classifier using weighted voting.

17 : **Training and Prediction:**
Using the training set, train the blending classifier. Afterward, use the learned classifier to the testing set to make predictions about bogus news.

5.1 Architecture of (OE-MDL)



5.2 Results and Discussions of the Experiment:

This part primarily evaluates the OE-MDL algorithm's capacity to identify false news. A dataset called the Liar dataset is used for this assessment. The Liar dataset is an accessible collection of official comments by politicians that have been meticulously labelled with truth-indicating information. These labels comprise a number of categories, including "true," "mostly true," "half true," "barely true," "false," and "pants on fire," to denote different levels of correctness in claims.

In addition to the statements' actual textual content, the Liar dataset also contains other metadata elements. Additional details about the statements are provided by these metadata characteristics, including the speaker's party membership and employment position. The dataset attempts to capture a full picture of the statements made by politicians by combining both textual and metadata elements.Specifically, this assessment procedure makes use of the Java programming language's implementation of the OE-MDL algorithm. It uses the Liar dataset to assess how well the algorithm performs when used as an ensemble model. An ensemble model improves the overall forecast accuracy and robustness by combining several separate models or algorithms.

The OE-MDL algorithm's performance is evaluated using four assessment metrics: accuracy, precision, recall, and F1-score. Accuracy is a crucial metric that indicates the proportion of precise forecasts the algorithm generates. In order to determine the true positives (instances accurately identified as false) and true negatives (instances successfully recognized as genuine), the total number of predictions made is compared. Higher accuracy numbers indicate better performance. It is defined as follows:

Accuracy = (The sum of the true positives and true negatives, divided by the sum of the false positives, false negatives, and true positives.) **(1)**

The percentage of true positives among all positive forecasts is quantified by precision. The computation is provided by:

Precision = false positives plus true positives, divided by true positives **(2)**

The percentage of true positives relative to all actual positives in the dataset is called recall. It has the following definition:

Recall = true positives divided by (false negatives plus true positives) **(3)**

A balanced measure between precision and recall is provided by the F1-score, which is a harmonic mean of the two. It is computed as follows:

F1-score = 2 * recall * precision / (recall + precision) **(4)**

The assessment metrics offer a numerical gauge of the algorithm's efficacy in identifying false information. Using the same parameters for comparison, the performance of each participant classifier is also assessed independently. Table 1 presents a comparison of classifier performance based on f1-score, recall, accuracy, and precision.

Table 3: Classifier Performance Comparison Using F1-Score, Accuracy, Precision, and Recall Metrics

Metrics	RF	J48	SMO	Naive Bayes	IBk	MLP	Dl4jMlp Classifier	OE-MDL
Accuracy	20.82	20.77	19.12	19.12	19.36	19.45	20.16	84.27
Precision	27.56	21.17	17.9	17.72	26.39	18.73	19.98	74.17
Recall	20.82	20.77	19.12	19.12	19.36	19.45	20.16	85.18
F1-Score	23.72	20.97	18.49	18.39	22.34	19.09	20.07	79.29

Additionally, following Figures displays a visual picture that compares the performance of eight distinct classifiers on a dataset: RF, J48, SMO, Naive Bayes, IBk, MLP, Dl4jMlpClassifier, and OE-MDL

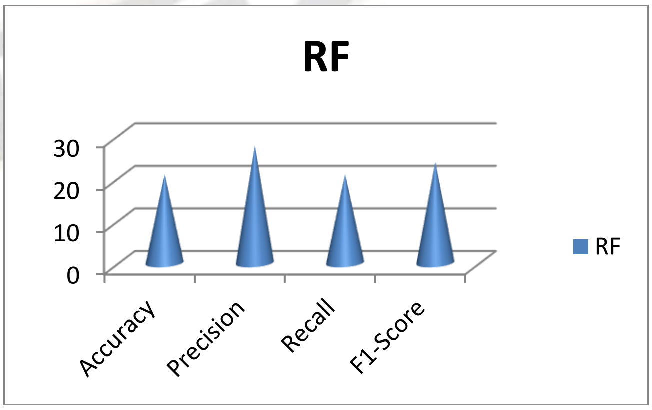


Figure 3: Classifier Performance for RF

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
***Article Received:** 25 May 2023 **Revised:** 12 September 2023 **Accepted:** 30 September 2023*

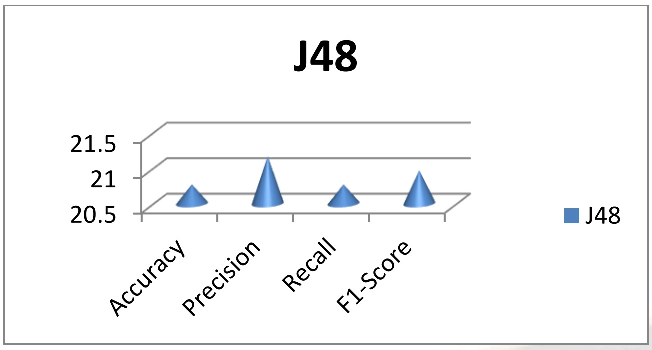


Figure 4: Classifier Performance for J48

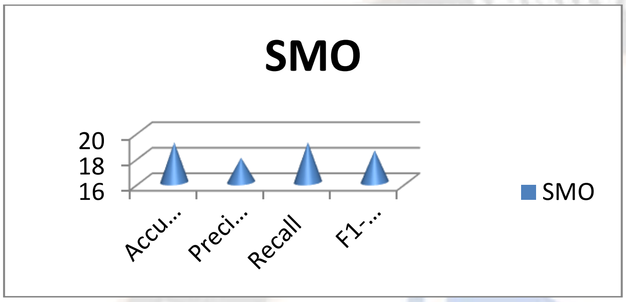


Figure 5: Classifier Performance For SMO

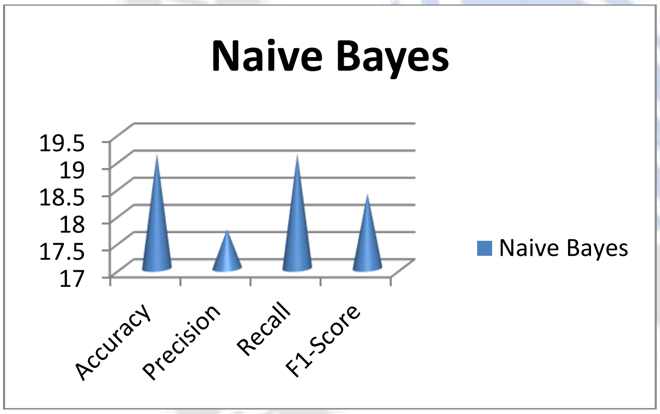


Figure 6: Classifier Performance For Naïve Bayes

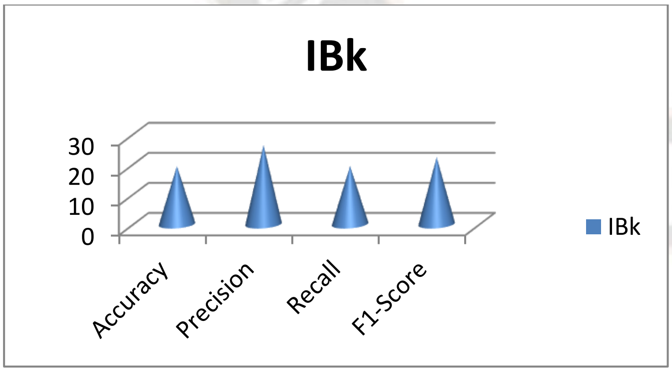


Figure 7: Classifier Performance For IBK

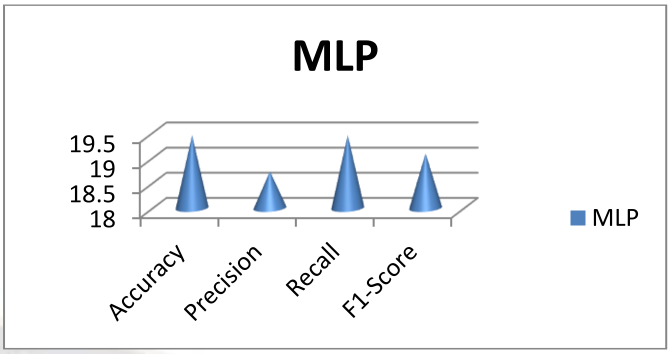


Figure 8: Classifier Performance For MLP

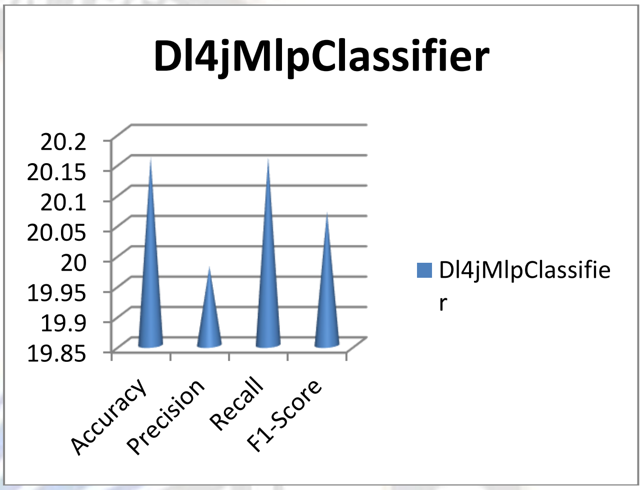


Figure 9: Classifier Performance For MLP

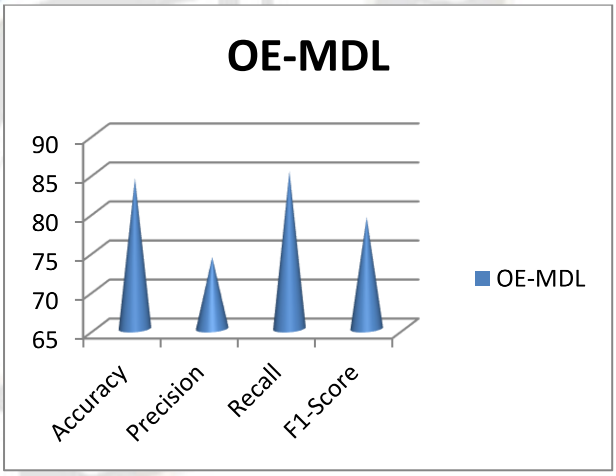


Figure 10: Classifier Performance For OE-MDL

OE-MDL is the algorithm that performs the best across several measures out of all the algorithms that were compared. It had the best accuracy, achieving an astounding 84.27% accuracy rate. This indicates that 84.27% of the cases classified properly when OE-MDL was applied, demonstrating a high degree of overall prediction accuracy. Moreover, the maximum precision value of 74.17% was likewise shown by OE-MDL. The precision metric quantifies the percentage of accurately identified positive cases among all the cases that were expected to be positive. OE-MDL demonstrated the capacity to reduce false positives, guaranteeing that a sizable percentage of the

cases predicted as positive were in fact accurate, with a precision score of 74.17%.
OE-MDL achieved the greatest recall value of 85.18% in terms of recall, which quantifies the percentage of correctly categorized positive occurrences out of the actual positive instances. This means that there were few false negatives because OE-MDL was able to accurately detect positive instances without missing many of them.Furthermore, with an F1-Score of 79.29%, OE-MDL obtained the highest score. The F1-Score provides an overall evaluation of an algorithm's performance by taking into account both precision and recall in a balanced manner. OE-MDL demonstrated its capacity to strike a good balance between recall and precision with an F1-Score of 79.29%, pointing to a great overall performance.

It is clear that the OE-MDL algorithm fared better than all other algorithms in the comparison given the continuously high ranks in accuracy, precision, recall, and F1-Score. These findings suggest that, of the algorithms examined, OE-MDL is the most appropriate for the particular classification problem at hand.

VI. COMBINING PHASES

This section explores the algorithmic implementation of the UFNDS Framework and provides it. The purpose of the UFNDS Framework is to effectively identify false news and evaluate the reliability of fresh news sources. The detailed algorithm that drives the UFNDS, from data preprocessing to the ultimate prediction, is described in this section. The UFNDS Framework Algorithm 6.1 employs a methodical approach to ascertain the authenticity of news data. It incorporates several stages, each of which adds to the process of making decisions.

Algorithm 6.1: UFNDS Framework

Input : Liar dataset (for training)
New news data (for prediction)

Output : Final_Result (shows if the recently released news data is phony or not)

1 : **Preprocessing (new news data):**
- Use preparation techniques such as text normalization, special character handling, tokenization, stop word removal, stemming, and lemmatization to clean and prepare the new news data.

2 : **Feature Extraction (new news data):**
- To represent semantic information and word importance, extract features such as word embeddings, N-grams, and TF-IDF scores from the preprocessed fresh news data.

3 : **ES-ECA Phase Classification (Liar dataset):**
- Train the ES-ECA phase using the Liar dataset.
- To categorize news articles as real or fraudulent, this phase uses a stacking ensemble technique that combines many base classifiers.

4 : **EHT-DL Phase Classification (Liar dataset):**
- Use the Liar dataset to train the EHT-DL phase.
- Deep learning algorithms are used in this step to classify news.

5 : **OE-MDL Phase Classification (Liar dataset):**
- Use the Liar dataset to train the OE-MDL phase.
- To categorize news stories, this stage integrates deep learning and machine learning techniques.

6 : **ES-ECA Phase Prediction (Liar dataset, new news data):**
- To determine whether the new news data is real or fraudulent, apply the trained ES-ECA phase.

7 : **EHT-DL Phase Prediction (Liar dataset, new news data):**
- To predict whether the new news data is authentic, apply the trained EHT-DL phase.

8 : **OE-MDL Phase Prediction (Liar dataset, new news data):**
- To determine whether the new news data is authentic or fraudulent, apply the trained OE-MDL phase.

9 : **Final_Result (Majority Voting):**
- The outcomes of the ES-ECA, EHT-DL, and OE-MDL stages should be combined by a majority vote method.
- Based on the consensus of these phases, Final_Result shows whether the new news data is authentic or fraudulent.

The goal of the UFNDS Framework algorithm is to categorize news items as authentic or fraudulent. Preprocessing fresh news data involves a number of operations, including stemming, lemmatization, tokenization, special character handling, text normalization, and stop word removal. Following preprocessing, the data is subjected to features extraction, such as word embeddings, N-grams, and TF-IDF scores, to represent its semantic content and word importance. The Liar dataset is then used to train the three classification stages (ES-ECA, EHT-DL, and OE-MDL). EHT-DL uses deep learning methods, OE-MDL mixes machine learning and deep learning, and ES-ECA uses a stacking ensemble strategy to merge numerous base classifiers. The training phases are utilized to forecast the veracity of fresh news info. The Final_Result, which indicates whether the new news data is real or phony depending on the consensus of these stages, is the outcome of the algorithm's final combination of their results by majority vote.

6.2 Architecture of Unified Fake News Detection System (UFNDS)

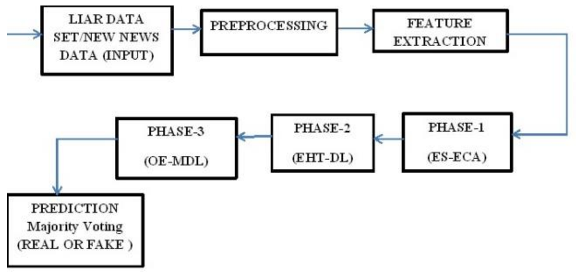


Figure 11. Architecture of Unified Fake News Detection System (UFNDS)

6.3 Benefits of the UFNDS architecture

The UFNDS Framework has various benefits when it comes to identifying and categorizing fake news.

1. **High Accuracy:** UFNDS can identify fake news with a high degree of accuracy by integrating numerous classification phases and employing ensemble approaches like as majority voting and stacking. This thorough technique lessens false negatives and positives.
2. **Semantic Understanding:** To extract features from news articles that capture word importance and semantic meaning, UFNDS employs feature extraction techniques including word embeddings and TF-IDF scores. This makes it possible for the algorithm to comprehend the text's context and meaning, which enhances its capacity to distinguish between real and bogus news.
3. **Versatility:** The framework can adapt to different kinds of data and news sources because it integrates a variety of machine learning and deep learning approaches. It is adaptable and adjustable to particular languages or domains.
4. **Consensus-Based Decision Making:** When deciding whether or not a news piece is authentic, UFNDS uses a majority vote process. This consensus-based method takes into account several viewpoints, which improves the categorization findings' dependability.
5. **Scalability:** The system is appropriate for real-time or high-throughput applications, including monitoring social media or news feeds, because it can be scaled to efficiently handle a large amount of news articles.
6. **Reduced Bias:** UFNDS lessens the possibility of bias that could be present in a single classification model by integrating several stages and methodologies. This aids in obtaining equitable and well-rounded classification outcomes.
7. **Continuous Improvement:** In order to keep up with changing fake news strategies and patterns, UFNDS can be updated and retrained on a regular basis using fresh data.
8. **Interpretable Results:** Transparency and accountability can be aided by UFNDS, which can offer insights into the reasons behind a news article's classification as real or fake based on the techniques and base classifiers selected for each phase.
9. **Comprehensive Approach:** Numerous feature extraction and preprocessing strategies are covered by the framework, along with a variety of classification approaches. This all-encompassing method raises the possibility of successfully recognizing different types of bogus news.
10. **Applicability to Real-World Scenarios:** The purpose of UFNDS is to tackle the real-world problems of detecting fake news, where news story authenticity might have important social and political ramifications.

In summary, the UFNDS Framework provides an effective and versatile approach to identify false news by utilizing an array of methods to enhance precision, dependability, and flexibility in response to changing tactics employed by false news outlets.

6.3 Summary

The UFNDS framework comes to light as a ray of hope, providing a comprehensive fix for the dangerous issue of fake news. It gives people and communities the ability to consume digital material with greater assurance about its integrity. Future developments in the crucial area of false news detection are paved as the nuances of each phase are explored, clarifying the benefits of integration.

References:

[1] Ahmed, H., Traore, I., & Saad, S. (2018). Detecting opinion spams and fake news using text classification. Security and Privacy, 1(1), e9.
[2] Tran, T., Valecha, R., Rad, P., & Rao, H. R. (2020). Misinformation harms: A tale of two humanitarian crises. IEEE Transactions on Professional Communication, 63(4), 386-399.
[3] Shu, K., Sliva, A., Wang, S., Tang, J., & Liu, H. (2017). Fake news detection on social media: A data mining perspective. ACM SIGKDD explorations newsletter, 19(1), 22-36.
[4] Seddari, N., Derhab, A., Belaoued, M., Halboob, W., Al-Muhtadi, J., & Bouras, A. (2022). A hybrid linguistic and knowledge-based analysis approach for fake news detection on social media. IEEE Access, 10, 62097-62109.
[5] Shahid, W., Li, Y., Staples, D., Amin, G., Hakak, S., & Ghorbani, A. (2022). Are you a cyborg, bot or human?—A survey on detecting fake news spreaders. IEEE Access, 10, 27069-27083.
[6] Dou, Y., Shu, K., Xia, C., Yu, P. S., & Sun, L. (2021, July). User preference-aware fake news detection. In Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (pp. 2051-2055).
[7] Shu, K., Zhou, X., Wang, S., Zafarani, R., & Liu, H. (2019, August). The role of user profiles for fake news detection. In Proceedings of the 2019 IEEE/ACM international conference on advances in social networks analysis and mining (pp. 436-439).
[8] Singhal, S., Kabra, A., Sharma, M., Shah, R. R., Chakraborty, T., & Kumaraguru, P. (2020, April). Spotfake+: A multimodal framework for fake news detection via transfer learning (student abstract). In Proceedings of the AAAI conference on artificial intelligence (Vol. 34, No. 10, pp. 13915-13916).
[9] Alonso, M. A., Vilares, D., Gómez-Rodríguez, C., & Vilares, J. (2021). Sentiment analysis for fake news detection. Electronics, 10(11), 1348.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
***Article Received:** 25 May 2023 **Revised:** 12 September 2023 **Accepted:** 30 September 2023*

[10] Sitaula, N., Mohan, C. K., Grygiel, J., Zhou, X., & Zafarani, R. (2020). Credibility-based fake news detection. Disinformation, misinformation, and fake news in social media: Emerging research challenges and Opportunities, 163-182.
[11] Shu, K., Cui, L., Wang, S., Lee, D., & Liu, H. (2019, July). defend: Explainable fake news detection. In Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining (pp. 395-405).
[12] Jaina, S., Patelb, R., Guptac, S., & Dhootd, T. FAKE NEWS DETECTION USING SUPERVISED LEARNING METHOD.
[13] Rao, G. P. C., & Narasimha, V. B. (2021). Feature-Based Learning Model for Fake News Detection and Classification.
[14] Jwa, H., Oh, D., Park, K., Kang, J. M., & Lim, H. (2019). exbake: Automatic fake news detection model based on bidirectional encoder representations from transformers (bert). Applied Sciences, 9(19), 4062.
[15] Raza, S., & Ding, C. (2022). Fake news detection based on news content and social contexts: a transformer-based approach. International Journal of Data Science and Analytics, 13(4), 335-362.
[16] K. Ashok, Rajasekhar Boddu, Salman Ali Syed, Vijay R. Sonawane, Ravindra G. Dabhade & Pundru Chandra Shaker Reddy (2022) GAN Base feedback analysis system for industrial IOT
networks, Automatika, DOI: 10.1080/00051144.2022.2140391
[17] Vijay Sonawane et al. (2021). A Survey on Mining Cryptocurrencies. Recent Trends in Intensive Computing, 39, 329.
[18] Sonawane, V. R., & Rao, D. R. (2015). An Optimistic Approach for Clustering Multi-version XML Documents Using Compressed Delta. International Journal of Electrical and Computer Engineering, 5(6).
[19] Kharade, K.G. et al. (2021). Text Summarization of an Article Extracted from Wikipedia Using NLTK Library. In: Singh, M., Tyagi, V., Gupta, P.K., Flusser, J., Ören, T., Sonawane, V.R. (eds) Advances in Computing and Data Sciences. ICACDS 2021. Communications in Computer and Information Science, vol 1441. Springer, Cham. https://doi.org/10.1007/978-3-030-88244-0_19
[20] Sonawane, V. R., & Halkarnikar, P. P. Web Site Mining Using Entropy Estimation. In 2010 International Conference on Data Storage and Data Engineering.