A Deep Learning Ensemble Framework for Fake News Stance Detection and Classification

A. Vaideghy, Thiyagarajan C, R. Sudha, Meena Suguanthi G

[1]*Assistant Professor, Department of Computer Science, PSG College of Arts & Science, vaideghy@gmail.com*
[2]*Associate Professor, Department of Computer Science, PSG College of Arts & Science, pcthiyagu@gmail.com*
[3]*Associate Professor & Head, Department of Computer Applications, PSG College of Arts & Science, ramanugam.sudha@gmail.com*
[4]*Assistant Professor, Department of Management, Karpagam Academy of Higher Education, meenasuguanthi@gmail.com*

The method of stance identification in misleading information is crucial to determining the credibility of the news since it aids in fact-checking by identifying diverse sources' positions on a primary assertion. To address the issues that arose throughout the process of learning ML models, several research projects included ensemble learning into ML models. The fundamental difficulty with models based on deep learning is that it takes a lot of skill and expertise to fine-tune the ideal hyperparameters in order to achieve a global minimal error. However, identifying the best hyperparameters necessitates a time-consuming approach in the field of search, which makes the work tiresome. The development of Ensembles Deep Learning Models (EDLM) for identifying and categorizing bogus news into predefined fine-grained categories is the focus of this work. Convolutional neural networks (CNN), LSTM (Long Term Memory), as well as bi-directional long short-term memory (Bi-LSTM) networks should be used as the foundation for ensemble models at initially. For the final classification, the representations produced by the two aforementioned algorithms are incorporated through a perceptron with multiple layers model (MLP). Experimental analysis demonstrates that our suggested ensemble learner technique outperforms individual learners.

Keywords: Stance detection, fake news, machine learning, hyperparameters, deep learning, Convolutional Neural Network, LSTM, Bi-directional Long Short-Term Memory and Multi-layer Perceptron Model.

1. Introduction

Public opinion on politics is influenced by how news is disseminated on social media. Social networks provide venues where information and articles can be distributed without verification or moderation. The sheer amount and diversity of information uploaded on social media make it difficult to moderate user-generated content. Recent elections are thought to have been influenced by fake news in particular, which is highly partisan-produced content on social media. Recent media attention has focused heavily on the misinformation conveyed by fake news, and current methods entail manual annotation by outside parties to alert consumers that shared content might be false. The fact that there isn't yet a consensus on what constitutes fake news and what standards should be used to determine whether an article is factual or untrue makes it difficult to identify misinformation. As a result, there isn't a single objective that the entire community may work on to compare the many ideas put forth. Up until recently, the adoption of false news evaluations has been somewhat gradual. While it is true that shared tasks may limit the community's attention to a particular task definition and dataset for established shared definition platforms and evaluations by CoNLL shared tasks1 have mostly accelerated progress. In the discipline of Natural Language Processing, or NLP, there has been a lot of interest in automatically recognising false news as a way to reduce the time-consuming and onerous human work of fact-checking. Nevertheless, assessing accuracies of news continue to be complex even in automations. Identifying false news based on other news outlets covering same topics can help initially. This operation is known as stance detection. Finding positions has always been an essential starting point for many activities, such as evaluating online discussions, figuring out the accuracy of news on Twitter, or understanding the reasoning and structure of persuasive writings. Pomerleau and Rao (2017) organised the first Fake News Challenge (FNC-1) for evaluating media coverages of issues for encouraging developments of automated techniques in fake news detections using ML and artificial intelligence (AI) is represented in Fig. 1. This competition drew in roughly fifty teams from academia and business. Their aim was to positions news articles with respect to given titles. There are four possible article viewpoints. It could cover the same ground, align or clash with the headline, or be unrelated. Information on the FNC-1 task, its criteria, data collection, and assessment measures may be found on their official website.

In this age of social networks, news changes quickly, making it challenging to promptly ascertain its accuracy. Consequently, the need for automated techniques to identify fake news has grown. A hybrid neural network architecture that combined advantages of CNN and LSTM was created to handle the aforementioned issues . In this work, we provide an approach that uses attitude labels to automatically classify news stories into four categories: discuss, disagree, and agree or disagree. According on the weight given to headlines' agreement with their appointed bodies, classification is made. The recommended approach is based on findings that publications can be identified as relevant by employing keywords in headlines. The task of automatic relationship discovery among text fragments is known as "stance detection," and it is performed provided an array of article body and headline pairs. Depending on the relationship between the news story's body and headline, the positions taken by the two parties can be categorized as "agree," "disagree," "discuss," or "unrelated."

Advanced ensemble learning-based methods was created in an attempt to enhance effectiveness of false news detections. Three phases make up the model. The TF-IDF approach is used to represent the characteristics in the initial stage. To lessen the resemblance between fake and real news, the semantics as well as contextual meanings of word representations were dropped from the feature sets. Using the n-gram model, new characteristics that are representative are generated. To uncover concealed and more representative features, an ensemble of CNN, LSTM, and the Bi-LSTM machine learning prediction models was created in the second phase.Each binary classifier in the ensemble is capable of predicting how accurate the news is. In other words, deep learning was used to learn the characteristics which represent the new class. These features are taken out from the deep ensemble model's final layer. The multilayer perceptron (MLP), is taught using the deep learning predictors' score outputs in the third stage. The results show that the study's suggested model outperforms the most current models.

This is how the rest of the essay is structured. Section II describes the most current research related to this subject. A synopsis of the information set, the database preparation procedures, and the deep learning model are given in Section III. The paper is concluded with potential future research paths in Section V, which also offers the findings and discussion from Section IV's discussion of the framework's performance evaluation criteria.

2. RELATED WORK

Davoudi et al. created an automated approach called "DSS" for detecting bogus news early where both dissemination branches and opinion networks were used concurrently and dynamically. The FakeNewsNet repository, which contains the most recent iterations of PolitiFact and GossipCop, two well-known industry datasets, is used to evaluate our proposed model. Because fake news is intended to intentionally mislead a broad range of readers, it is challenging to distinguish it from other types of news based alone on its substance. Consequently, other information is needed, such as the social context. Furthermore, in order to mitigate false news' detrimental impact on society, it is imperative that it be identified as soon as feasible.

For the purpose of identifying multifunctional inconsistent semantics for detectable false news, Wu et al. introduced the Multifunctional Integration and Inaccuracy Reasoning (MFIR) model. Extensive testing on three datasets has proven our model's efficacy and improved accuracy by up to 2.8%. However, they struggle with two major problems, including shallow cross-modal feature integration and trouble capturing inconsistent data. Introducing an Artificial Learning optimization strategy for automatic headline categorization in Facebook and Twitter was advised by Setiawan et al. in their study. The research has benefited from the deliberate use of NLP algorithms for social forum fake news findings to distort stories from unreliable sources. The study's remarkable findings are attributed to the classifier—which is powered by a mixture of hybrid support vector machines and achieves 91.23% accuracy—and the document's word content, which functions as an attribute of the extraction approach. On the other hand, scientists are interested in the potential field of online misinformation analysis.

A innovative false news detection technique utilizing the Natural Language Inference (NLI) approach was proposed by Sadeghi et al.. The suggested method takes advantage of a resembling method, which depends on inferring authenticity using an assortment of reliable news, as opposed to using solely statistical aspects of the subject matter or the circumstances surrounding the news. This method makes use of relevant and comparable news that has been published in reliable news sources as supplementary knowledge to determine news items' validity. The trials demonstrate that the suggested strategy achieves 85.58% and 41.31% accuracy in the FNID-Fake News Net and FNID-LIAR datasets, respectively. These results represent absolute improvements of 10.44% and 13.19%, respectively. However, given the abundance of bogus news that is disseminated on online social media, this technique is ineffective. They therefore constantly strive to automate this procedure in order to recognize false information as well as deal with the excessive publication.

The effective detection of bogus news was acknowledged by Dixit et al. . The proposed approach for recognising false news included information pre-processes, feature extractions/reductions and classifications. Tokenization, stop-word removal, and stemming are used in data pre-processes for preparing inputs and subsequently PPCA reduced the characteristics in order to improve accuracy. After the feature has been recovered, it moves on to the classification step, where the LSTM-LF algorithm is applied to precisely classify the headlines as true or phoney. This technique's biggest drawback is reportedly its long execution time.

In order to identify fake news using various datasets, Choudhury&Acharjee compared the classifiers for SVM, Naive Bayes, Random Forest (RF), and Logistic Regression. SVM classifier showed highest accuracy where 97%, and 96% were corresponding scores of Fake Job Posts and Fake News datasets, respectively. SVM, Naive Bayes, RF, and logarithm regressions were considered in fitness functions for GA-based fake news detection systems. In the study's proposed schema, SVM and LR learners achieved 61% accuracy in LIAR datasets, while RF and SVM achieved accuracy scores of 97% for fictitious job lists. Although this method was successful in achieving high resilience and accuracy rates, it was unable to integrate log keywords into machine learning algorithms.

An ensemble ML model based on credibility was used by Ramkissoon & Goodridge for detecting fake news. The legitimacy ensembles combined 2-class neural networks and boosted decision trees' learning capabilities. The ensemble method uses a phony "mixture of experts" approach. A particular version of 2-class logistical regressions were used for the gating model. Legitimacy was validated by employing typical datasets with attributes related to trustworthiness of publishers and traits analysed. The experimental results using four assessment methods demonstrated ensemble ML was best achieving 96.9% accuracy. Furthermore, ensemble approaches for identifying fake news were not incorporated into this approach.

Principal Component Analysis (PCA) and the chi-square are two separate dimensionality reduction techniques that were combined by Umer et al. in their hybrid neural network design. This research recommended utilising techniques for dimensionality reduction to lessen the feature vectors' dimensionality before delivering them to the classifier. The rationale for this study was created using a dataset that was made available by the Fake News Challenges (FNC) website. The dataset consisted of four kinds of stances: agree, disapprove, discuss, and irrelevant. Nontraditional variables were added to PCA and chi-squares for obtaining historical features in identifications of bogus news. The work aimed at determining news article links to headlines. The recommended model improved performances by 4% and 20% for accuracies and F1-scores. Their experimental outcomes showed that, with 97.8% accuracy, PCA outperformed both Chi-squares and other methods. This strategy's primary drawback was assumed its execution, which was incredibly challenging.

In order to facilitate characteristic cross-topic propagation and concurrently frame stance as well as rumour recognition as multistage classification challenges, Li et al. created a hierarchical heterogeneous network by linking postings that contained the same high-frequency terms. The work implemented a multigraph artificial neural network system that can dynamically integrate context's attribute and structural information. This system will enable the periodic updating of node embeddings while simultaneously being affected by stride and rumour detection. In trials on real datasets from Reddit and Twitter, the approach performed much better than other methods for recognitions of rumours and attitudes. The trial results also show that our approach requires less tagged data and is easier to comprehend. The main disadvantage of this approach was its poor overall effectiveness despite very high efficiency rates.

3. PROPOSED METHOD

This proposed strategy uses four stages to identify false information on social media. The evaluation of a news article's headline-focused relative position served as the basis for this study.

The initial phase of the technique involves pre-processing the data set to convert it from an unstructured state to a structured state.

In the second stage, Feature Extraction employing MBDFO is utilized to discover the unidentified characteristics of bogus news and numerous associations between news items.

PSO-based feature selections as the third stage of feature count reduction.

In the final phase, our research creates an EDLM for discovering how to characterize news articles and successfully detect fake news. This study employed datasets from FNC website, which featured four different types of markers: agree, disagree, debate, and irrelevant.

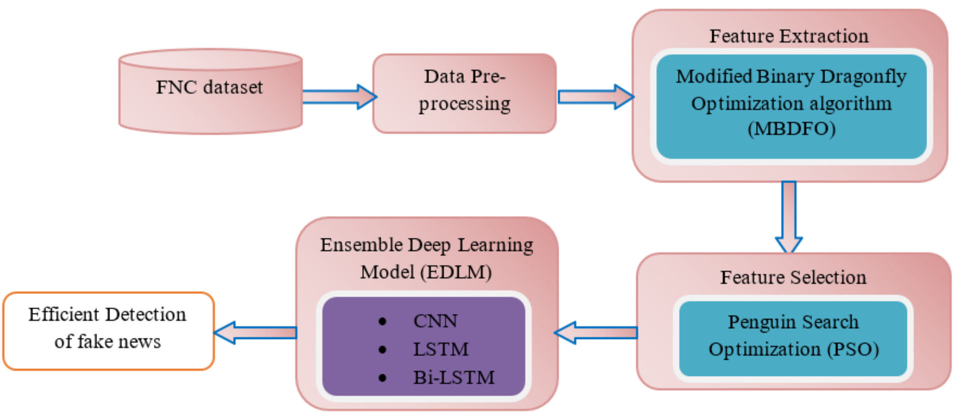

Fig. 1. Fake news detection model

Input Dataset Description

The conventional database for the False Information Challenges was gathered from the official website. 75,385 tagged instances, 2,587 articles bodies, and about 300 headlines make up the FNC dataset. For each allegation, there are 5 to 20 news items. Table 1 shows that of these kinds of headlines, 7.4% are approved, 2.0% are disapproved, 17.7 percent are subject to discussion, and 72.8% are inconsequential. The claims made in the article's body are marked by human participation. These labels' explanations are as follows:

Agree: I agree that every article's title and body should be related.

Disagree: The article's content and headline have no bearing on one another.

Discuss: how little the title of the article and the body resemble one another or are neutral.

Unrelated: The subject of the title was entirely unrelated to the content.

Based on the FNC-1 challenge instructions, the dataset was split into training (972 occurrences) and testing (413 instances). In training data, the ratio of article bodies to headlines is 1: 648 to 683. The test data contains 904 article bodies and about 880 headlines.

Table 1. Dataset statistics.

Dataset	**Headlines**	**Tokens**	**Instances**	**Agree**	**Disagree**	**Discuss**	**Unrelated**
FNC-1	**2,587**	**372**	**75,385**	**7.4%**	**2.0%**	**17.7%**	**72.8%**

Data Pre-Processing

Pre-processing is a typical data mining technique that converts erratic and inaccurate raw data into understandable computer representations. On the FNC-1 dataset, lowercase conversions, stop word removal, tokenization, and stemming were all done using techniques from the Keras as toolkit. Stop words are common words like "of," "the," "and," "an," and so forth that are used in the text but have no real meaning in terms of attributes and are therefore irrelevant to this study. Processing time was decreased by eliminating stopwords, and space was conserved by avoiding the aforementioned unnecessary words. Keywords bearing identical meanings that appear multiple times in the text include games and games. In this situation, presenting the material in a straightforward, universal format is really advantageous. This stemming procedure is performed using Porter's stemmer method from open sourced implementations of NLTK.

Following the completion of the pre-processes outlined above, characters in subtitles were reduced to 372 words. The headline was converted into a word vector with the help of the Keras as library's tokenizer function. Following pre-processing, word embedding (word2vec) was used to map words and texts to groups of vectors. Finally, 5,000 unigrams based on story headlines and body content are compiled into a dictionary. Every headline must be at least as lengthy as it is at its longest. Zero padding is applied to headlines that are shorter than the longest length.

Word Embedding: The Google tool "Word2vec" is utilized to increase the range of words in newspaper articles to 300 dimensions enabling the purpose of embedding in the preprocessing. It is adequate to represent the similarities and contrasts between words using the 300-dimension semantics. The written content is digitized and made into a countable unit at the same time. The evaluation of the embedded LSTM model is built on the semantics.In text classification or natural language modeling, word embedding is a common technique. Text must first be digitized before processing. We increase the text's dimensions to match the 300 dimensions found within the News from Google dataset (Word2vec, n.d.) because digitized texts lose their originality of meanings. Each of the words in the news is first tokenized, and then once the dimensions are increased, each word's dimensional values are modified.

Grammar Analysis: We believe that the sentence patterns in an article can provide useful information during the grammar analysis during preprocessing. Actually, the sentence depths—i.e., mean and Q25—are different between true and false news. The news is examined in light of grammatical conventions. After determining the complexity of each of the sentences in a piece of content, we additionally look at the distributions of sentence depth, taking into account other elements like mean and Q25.

Mean: The average length of each article's sentences is referred to as The Mean. In addition to having a deeper mean than actual news, false news also has a mean distribution with a wider range of values than real news. The following equation represents the average depth of a sentence in an article.

$$Mean = \frac{\sum_{i=1}^{n} depth_i}{n} \tag{1}$$

where n is how many sentences make in an article.

Q25: This term implies article's 25th percentile sentence depths. Fake news have deeper Q25 distributions than actual news, in addition to having a deeper Q25 than real news Q25 of sentences' depths in articles are depicted below.

$$Q25 = 25thpercentileo\cancel{\sigma}\cancel{f}Sort_{asc}(depth) \tag{2}$$

Where Sortasc is the sentence-depth-based ascending sort algorithm. Additionally, dimensionality reduction approaches are given the functions of word embedding and grammar analysis (mean and Q25).

Dimensionality Reduction Methods

Through feature extractions or choices, the number of characteristics within text classification can be decreased. In feature selection processes, the other features are not taken into consideration, only the most significant and pertinent attributes are kept. Alternatively, feature extractions modify initial vectors to produce new vectors with distinct characteristics which are the base for feature reductions. Reducing features improving performances in terms of execution speeds. Text classification outcomes are significantly impacted by feature reductions. Consequently, it is essential to pick the appropriate selection approach for dimensionality reduction. The text classifier's scalability may be improved by utilising the two-dimensionality reduction techniques MBDFO and PSO. In order to deal with Feature Selection (FS) problems, the MBDFO algorithm uses many methods to adjust the parameters of its five main coefficients.

Feature selection Using PSO

The goal of the suggested opinion mining system is to improve the sentiment analysis of medical tweets. The structure and work flow of the PSO-based feature selection method is depicted in Fig. 2. Following pre-processing of the data, feature descriptors are used to extract the features. The PSO method is then used to choose the features. To evaluate the categorization accuracy in false news stance identification, three classifiers are lastly used. In this section, The procedure for choosing PSO features is explained. PSO is dependent on how penguin groups use ice holes to fish for food. The penguins split up into defined counts of groups, and they randomly searche for fishes until their oxygen reserves are depleted. When the oxygen returns, they seek again until they find a sufficient number of fish. After that, they contrast the food's locations with those of other groups to determine which is best for hunting. The best feature selection is made possible by this approach, which is used with PSO. Initial oxygen amount, penguin population counts, other parameters are defined. Subsequently, penguins are split up into smaller groups, and each group travels on its own towards an area where food is available. To adequately handle the choice of features problem, PSO mapping must be carried out. The randomly generated samples of penguins reply sets are features, and categories for feature subsets are selected. Each feature's fitness is computed, and the optimal values get the best result. The alternatives advance in the direction of the top choices. This quantity is represented as:

$$\mathcal{X}_{new} = \mathcal{X}_{old} + rand \times (\mathcal{X}_{localbest} - \mathcal{X}_{localold}) \quad (1)$$

Where $\mathcal{X}_{old}$is the earlier local best solution, $\mathcal{X}_{old}$is the preceding solution, $\mathcal{X}_{new}$is the most recent solution, and $\mathcal{X}_{new}$is any number between [0, 1]. The oxygen reserve of each penguin is modified using Eq. (2) following each plunge.

$$\mathcal{O}_j^i(t+1) = \mathcal{O}_j^i(t) + (\mathcal{of}(\mathcal{X}_{new}) - \mathcal{of}(\mathcal{X}_{old}) \times |\mathcal{X}_{new} + \mathcal{X}_{old}|) \quad (2)$$

$\mathcal{O}_j^i(t+1)$ represents latest oxygen reserves, $\mathcal{O}_j^i(t)$ implies prior oxygen reserves, and $\mathcal{of}$ stands for defined objective functions.

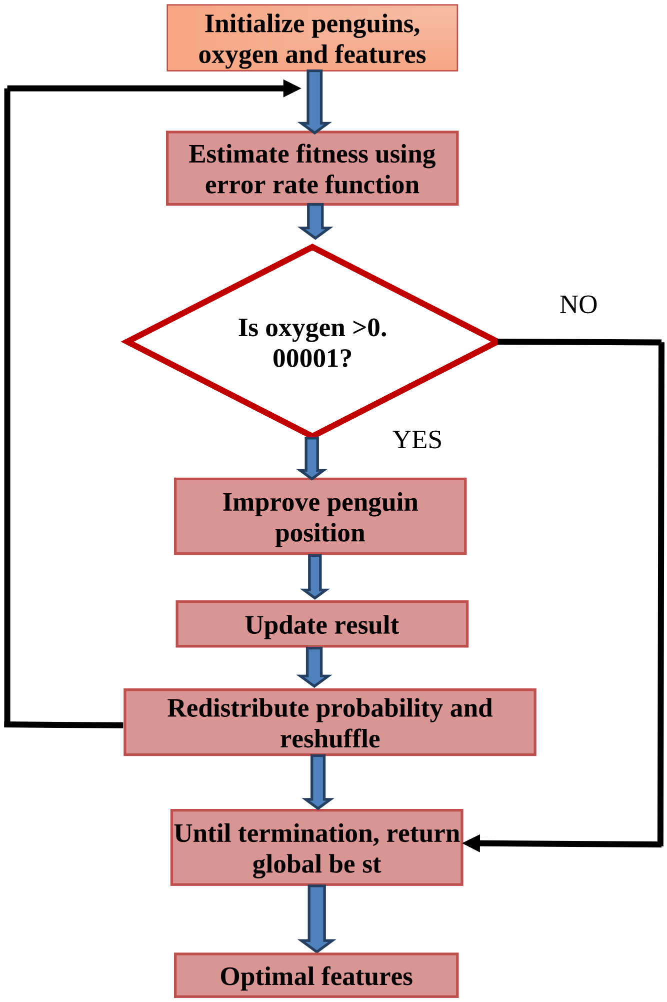

Fig 2. Architecture of PSO Feature Selection based opinion mining Framework

The collective membership of penguins and the quantity of eaten Fish (QEF) are both updated in a similar manner. Utilizing the error rate function, the QEF is represented as the food's energy content.

$$\mathcal{QEF}^{i}(t+1) = \mathcal{QEF}^{i}(t) + \sum_{j=1}^{n}\left(\mathcal{O}_j^i(t+1) + \mathcal{O}_j^i(t)\right) \tag{3}$$

By calculating the possibility P_iof associating to the group i, an association modification of penguins is carried out.

$$P_i(t+1) = \frac{\mathcal{QEF}^{i}(t)}{\sum \mathcal{QEF}^{i}(t)} \tag{4}$$

The world's finest characteristics are returned after the upgrade procedures are finished. The following details the entire PSO feature selection process.

1. Initialize

2. Examine the previously edited tweet data.
3. Predetermined features (number of penguins)
4. Produce arbitrary penguin populations in clusters.;
5. Determine each penguin's mistake rate (fitness function);
6. Where i=1 and n=generations;
7. For each penguin i ∈ P do
8. Set each penguin's oxygen reserve to zero.
9. While oxygen reserves are not depleted, (until 0.00001)
10. Take a random action.
11. Using Eq. (1), relocate the penguin location.
12. Update the neighborhood's top solution;
13. Use Eq. (2) to update the oxygen backup;
14. End while
15. End for
16. Using Eq. (3), update the total amount of devoured fish in the holes
17. Use Eq. (4) to update group membership.
18. Rearranges penguin probabilities in levels and holes
19. Discard the groupings with no participants;
20. Update most significant solution;

End

The feature subsets selected through this procedure are thought to be the finest ones. The total quantity of feature subsets is reduced by the information gain measure by ranking the chosen features. The classification is carried out using EDLM after the best feature subsets have been chosen. These classifiers' classification procedure is improved by the PSO. The investigational findings are used to verify how well the classifiers work.

Proposed method EDLM

As previously said, we propose a multilayered supervised neural network (LSTM) as a reconstruction technique with CS utilizing deep learning theory in order to gain superior reconstruction results. We recorded pressure data over a time-sequenced region of the human body. Additionally, the pressure readings within this matrix are related to the time sequence if the pressure array is rewritten as a single-column vector.With the help of an LSTM network, we can better recreate the original signal based on the relationship between the sampling time sequence. In the suggested method, the data vector y of dimension M 1 serves as the LSTM network's input and$\mathcal{y}$ of size $M\times 1$, is a CS measurement vector. We can also substitute Φx for $\mathcal{y}$ in formula (2), where $\Phi$is the evaluation matrix with dimensons$M \times N$. In this case, the random Gaussian matrix is used. Regardless of identically spaced (i.i.d) arbitrary Gaussian matrix $\Phi$is used in this situation. In a Gaussian matrix, elements i and j are random variables that are independent that follow the following distribution:

$$n\varphi_{i,j} \sim N\left(0, \frac{1}{n}\right) \tag{5}$$

The components of formula (5) follow a Gaussian distribution with an expectation of 0 and a variance of $\frac{1}{n}$. When using adaptable pressure array sensors to determine the pressure data of a specified area of the model, we may only use select columns as measurement points; the data for the remaining observable points are all zero. The acquired matrix has a sizable number of zero elements. Additionally, this generated matrix is thin. Thus, the matrix x is sparse. The vector x_r of size $N \times 1$ that the multilayer LSTM network produces contains the initial format of the data along with the recovery outcome.The initial signal x's length is N. Here, we compute the loss function using the mean squared error (MSE).

$$MSE_g = (1/n) \sum_{i=1}^{n} (detection_i - Original_i)^2 \quad (6)$$

here$detection_i$ is the outcome of the suggested LSTM algorithm's detection. $Original_i$is the data that came through the human body originally. Formula (7) can also be used to calculate formula (6).

$$MSE = \left(\frac{1}{n}\right) \sum_{i=1}^{n} (x_{r-i} - x_i)^2 \quad (7)$$

where the component of the reconstruction result x_{r-i} is $x_r . x_i$.A component of the original signal$(x_1, x_2, \ldots\ldots x_{n-1}, x_n)$is said to be known as called x_{r-i}. The loss that we determine for the reconstruction outcome is MSEg. Here we built up our multilayer LSTM by D_{train} training set.The dimension vector of CS determined by formula is D_{train}=$\{(y^{(1)}, x^{(1)}), (y^{(2)}, x^{(2)}), \ldots\ldots, (y^{(p)}, x^{(p)})\}$,$y^{(\mathrm{p})}$

$$\left(O^{(p)}\right) = x^{(p)} \quad (8)$$

Where $y^{(\mathrm{p})}$ is the CS measurement of $x^{(p)}$. The original data is $O^{(p)}$. In pairs, we add measurement vectors and the original data to the training set. The test set's $y^{(q)}$is utilized as the data source of the multilayered LSTM network to produce the reconstructed$x_r^{(q)}$, which is then compared to $x_r^{(q)}$the test set's true value $x^{(p)}$. This is done following that the multifunctional LSTM network has been trained through the training set $Dtrain$.

We suggest feeding each one of these relations into a different network layer in order to make sure that deep networks can comprehend these relationships. Afterward, immediately following extracting all the relationships, we group them together using the five-column attributes that contain data on the speaker's total number of credit history entries. Additionally, we also incorporate a unique feature vector that we suggest be constructed from the count history data. The vector that follows is a number with a length of five that represents each of the five counts history columns; one of the digits is set to '1' (according to whichever section has the highest count), and the other four are set to '0'.

Bi-LSTM: The networks containing LSTM components which analyze sequences of words in both ways, across one side to the other in addition to from right to left, are known as bidirectional LSTMs. Considering post padding by zeros, the maximum possible input length for each statement in this model is 50 (the average statement length is 17, the maximum is sixty-six, while only fifteen percent of the experimental data had a length larger than 50).The maximum allowable length of an input sequences is 5, 20, and 25 for attributes such statement type, speaker's occupation, and context, respectively. After that, separate Bi-LSTM networks with 50 neural units in each direction for each individual embedded inputs are fed. Each Bi-LSTM network's output is then sent to a highly dense structure of 128 neurons using the activation function "ReLU." LSTM is trustworthy when using a standard RNN approach, but it uses a variety of techniques to figure out the hidden state, which fixes the RNN's problem but cannot handle long-distance dependency. The LSTM method uses a series of identical memory modules with three gates as illustrated in Fig. 3. Associated LSTM unit state values for words are provided below using text vectors of features FV and words as an example.

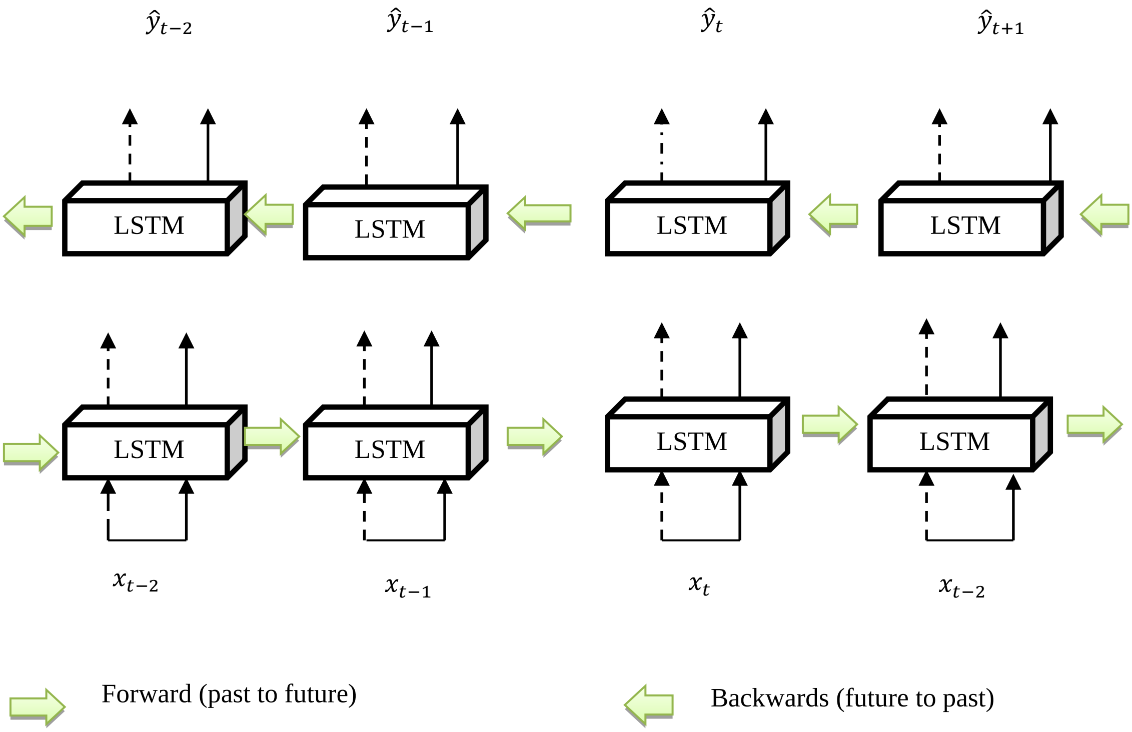

Fig. 3. The structure of three-time steps relating based BiLSTM

The following list contains the specialized estimate functions, where Σ represents sigmoid functions while ⨀ implies dot multiplications. The outputs of sigmoid functions, which simultaneously receive data from previous hidden states and current inputslie in the interval [0,1]. It should be discarded in higher proportions the closer it gets to 0, and retained in bigger proportions the closer it gets to 1. A forget gate is defined by the F_{gt}:

$$Fgt_{\mathrm{t}} = \Sigma\left(W_{Fgt} * [hs_{\mathrm{t}-1}, x_{\mathrm{t}}] + b_{Fgt}\right)$$

The input gate Igt_t is used to update the state of the cell. Send the data relevant to the hidden state of the previous layer, together with the current input data, to the sigmoid function first. The value can be adjusted from 0 to 1 to indicate which information requires updating. One indicates importance, whereas zero indicates unimportance. Moreover, the current input data and the knowledge of the hidden state of the layer before it are sent to the tanh function, which creates a new candidate value vector (cvv). Tanh's output value is multiplied by the Sigmoid's output value, at the very least. Depending on the sigmoid's conclusion value, the data that is essential and must be retained in the tanh final result will vary.

$$\mathrm{Igt}_{\mathrm{t}} = \Sigma\left(W_{Igt} * [hs_{\mathrm{t}-1}, x_{\mathrm{t}}] + b_{Igt}\right)$$

$$\widetilde{cvv}_{\mathrm{t}} = \tan hs(W_{cv} * [hs_{\mathrm{t}-1}, x_{\mathrm{t}}] + b_{cv})$$

To determine the cell state, use the current stateCst. First, the cell's state from the previous layer is multiplied point by point by the forgetting vector. If it has been split by a value that is close to zero, it must be removed in its new condition. Next, update the cell state with the newly learned information by the neural network by gradually adding this number to the input gate's resultant value. At this point, the altered cell state is obtained.

$$Cst_{\mathfrak{t}} = Fgt_{\mathfrak{t}} \cdot Cst_{\mathfrak{t}-1} + Igt_{\mathfrak{t}} \times \widetilde{cvv_{\mathfrak{t}}}$$

The output gates O gt are used to compute probabilities of next concealed states. Previously entered info is hidden and passing current inputs and previous concealed states through sigmoid functions, the most recent cell states are passed to tanh functions.The outputs of tanh and sigmoid outputs are multiplied to determine what concealed states should include. The freshly generated hidden state and the newly formed cell state are then moved to the next time step, when the hidden state will be used as the current cell's outcome.

$$Ogt_{\mathfrak{t}} = \Sigma\left(W_{Ogt} * [hs_{\mathfrak{t}-1}, x_{\mathfrak{t}}] + b_{Ogt}\right)$$

$$hs_{\mathfrak{t}-1} = Ogt_{\mathfrak{t}} \cdot \tan hs(Cst_{\mathfrak{t}})$$

Where the LSTM weights W_{Fgt}, W_{Igt}, $W_{cv}$and $W_{Ogt}$are concerned. The variance of the LSTM is represented byb_{Fgt}, b_{Igt}, b_{cv}andb_{Ogt}. The hidden state at time t is denoted by$hs_{\mathfrak{t}}$. The tangent hyperbolic function is called tanhs, and Σsigmoid is the activation function. Periodical data from the series make up the LSTM, which is insufficient.

When this data is accessible into the future, it might be highly useful for a series of actions. Onward and backwards LSTM layers make up the bidirectional LSTM. The technique for this method is as follows: forward layers retain sequences' existing data, while backward layers hold new sequence data. Comparable output layers are connected to these layers anf when this information becomes available in future, it could assist multiple actions. Bidirectional LSTMs are made of forward and reverse LSTM layers and uses the following techniques: forward layers store old sequence data, while backward layer store newly created sequence data. These layers are connected to a comparable output layer.The use of the ordered content information to its fullest extent is another important aspect of this model. Assume that word embedding $w_{\mathfrak{t}}$ is the input at time $\mathfrak{t} - 1$, resulting in $\overrightarrow{hs}_{\mathfrak{t}}$for the forward hidden unit and $\overleftarrow{hs}_{\mathfrak{t}-1}$, for the backward hidden unit's simulation outcome. Finally, the results obtained using both a hidden unit and a backward unit at time t are provided below:

$$\overrightarrow{hs}_{\mathfrak{t}} = \mathfrak{L}\left(w_{\mathfrak{t}}, \overrightarrow{hs}_{\mathfrak{t}-1}, cvv_{\mathfrak{t}-1}\right)$$

$$\overleftarrow{hs}_{\mathfrak{t}} = \mathfrak{L}\left(w_{\mathfrak{t}}, \overrightarrow{hs}_{\mathfrak{t}-1}, cvv_{\mathfrak{t}-1}\right)$$

where$\mathfrak{L}(\cdot)$is the LSTM hidden layer's hidden layer task. It is clear that $\mathfrak{H}$ is the total amount of hidden layer cells because the inward input vector is $\overrightarrow{hs}_{\mathfrak{t}} \in R^{1\times\mathfrak{H}}$, whereas the corresponding backward output vector is also $\overleftarrow{hs}_{\mathfrak{t}} \in\leftarrow R^{1\times\mathfrak{H}}$, These vectors must be integrated to obtain the text feature.

$$\mathfrak{H}_t = \overrightarrow{hs}_{\mathfrak{t}} \parallel \overleftarrow{hs}_{\mathfrak{t}}$$

CNN: The pooling and convolution properties of CNN have been effectively employed by numerous experimenters over the past few years to uncover hidden features in both texts and images. A MaxPooling layer will choose the largest input among the convoluted inputs after using a convolution layer with a $n \times m$kernel size (in which m-size of word embedding). Speaker, party, and state attributes are embedded using a pre-trained n-dimensional FNC, and the embedded inputs are subsequently fed into different Convlayers. Different credit history counts, fictitious speaker remarks, and a feature we proposed built utilizing the financial history counts are all fed directly into different Conv layers. In order to extract features, a CNN is employed, and in order to extract multiple features, one dimension rather convolution operation with various size filters is applied $\mathbb{w} \in R^{\mathfrak{H} \times k}$. To get the k-largest component of the feature map, P^{k-max}, which can better represent the relationships of long-range elements in the sequence, let's additionally apply dynamic k-max-pooling . The formulation of the j-thmapping of features is

$$P_j^{k-max} = \mathfrak{L}\left(ReLU\langle w_{\mathfrak{t}}, \widehat{\mathcal{X}}\rangle_T + b\right)$$

When T is the Trace/Frobenius inner product, $w_{\mathfrak{t}}$is the width of the filter with $h_j$height, and b is bias, $\mathfrak{L}$ is the k-max pooling operation. The final written structure TC representation is determined by

$$TC = \mathfrak{c}_s\left(P_1^{k-max}, P_2^{k-max}, \dots, P_m^{k-max}\right)$$

where$m$ is the total number of filters and $\mathfrak{c}_s$is the conversion procedure and concatenation process. Let's concatenate the representations of the content network and the structure network for the fusion layer, followed by a completely linked layer. The likelihood that an event is a rumor using a softmax function is as follows:

$$\mathcal{X} = \mathfrak{c}(TC, \mathfrak{H}_t)\text{and}Y = \Sigma\left(\sigma\!\!\!/\!f(\mathcal{X})\right)$$

Where in Y is the prediction outcome showing either the occurrence is a rumor, $\sigma\!\!\!/\!f$ represents the completely connected layer, and Σ represents the sigmoid function. The textual content network is represented by TC, and the cross-entropy loss function is used, i.e.

$$Loss = \sum_{i=1}^{m} Y_i \log \hat{Y}_i + (1 - Y_i) \times \log\left(1 - \hat{Y}_i\right)$$

whereY_i and $\hat{Y}_i$ stand for the genuine label and prediction outcome of the i-th event, respectively. Let's use the dropout method in the outermost layer to prevent overfitting and Adam as the optimizer to hasten convergence during training.

Ensemble deep learning Model

The MLP, which is produced at this step and is most frequently employed by investigators for regression and classification studies, is created and described in Fig.4. The six deep predictors' outputs' extracted features q were sent into the MLP as its input. Input, two hidden and output layers with twelve, thirty-two, sixteen, and six neurons encompass five layers of MLP for multi-class classifications using the ReLu activation and SoftMax functions.

The ideal counts of neurons and hidden layers were determined via trial and error, and those values were selected. In the earlier stage, the input features for the MLP classifier are extracted using the deep learning model that has been developed. If $P(cl)$ stands for the likelihood of correctly predicting a given class cl (such as bogus news), then

$$P(cl) = \sum_{i=1}^{n} w_i * x_i + \theta$$

where the letters w_i, x_i, and θ stand for the neuron's weight, the preceding layer's output that corresponds to it, and weights of deep learners as they learn from preceding phases, respectively. Based on the results of MLP-based deep learning classifiers, weightsw_i are learned. Each classifier makes a contribution to the weight calculations, which results in the $P(cl)$.The sigmoid function is used to determine the ultimate classification score $S(cl)$for the new class.

$$S(cl) = \frac{1}{P(cl) + e^{P(cl)}}$$

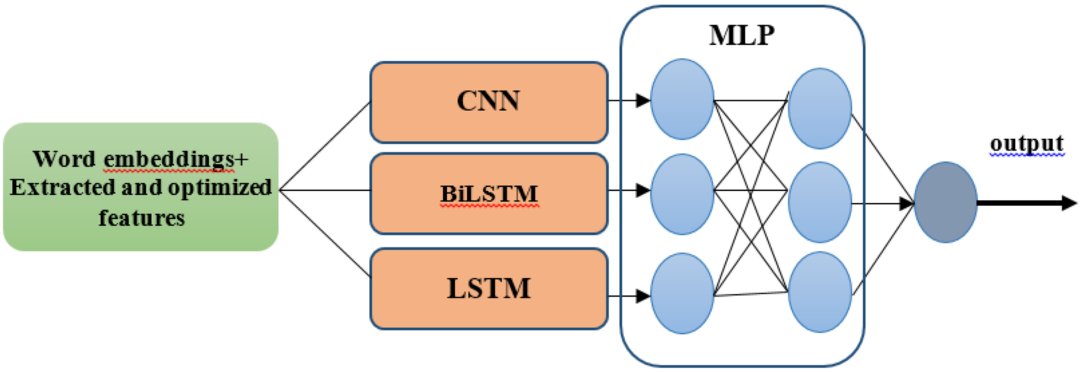

Fig. 4. Steps of Ensemble Various Network Structures

In the initial stage, CNN, LSTM, and BiLSTM is with attention layer are independently trained on the meticulously preprocessed data. Then, by initializing weights, each prediction made by the previously mentioned models is summarized. We used the Xavier weight initialization method in this case. By assuring fine-grained classification, this in turn reduces the total misclassification error rate. The improved effectiveness in the shallow ensemble network over the individual deep networks will be shown in the succeeding sections, which will comprise of the experimental data.

Experimental Results and Discussion

A series of tests using the FNC-1 datasets presented below are used to assess the effectiveness of the proposed model. This section outlines these trials and compares the outcomes to those obtained using alternative cutting-edge methods. To identify bogus news, multiple data sets were recently made public. The availability of an extensive data set for training models is one of the essential requirements for using neural networks. In this study, the deep models were trained using a dataset from Kaggle that included a number of documents. This system's performance is assessed using the provided dataset, and it is contrasted with more contemporary methods like CSI (Capture, Score, and Integrate), CNN

(Convolution Neural Network), LSTM (Long Short-Term Memory), and Bi-LSTM. The following section presents a collection of experiment-specific assessment metrics based on true positive (TP), false positive (FP), true negative (TN), and false negative (FN) rates. The first performance statistic, called accuracy, measures the proportion of relevant incidents among those that were returned. Recall is the percentage of occurrences that are pertinent and tuned-in, and it is the second performance metric. The metrics of recall and accuracy are both important for assessing the effectiveness of a prediction technique, considering the reality that they typically have opposing characteristics. The F-measure is produced because these two metrics can be combined using the same weights. The proportion of accurately foretold occurrences to all predicted occurrences is used to determine the reliability aspect of the final outcome metric.

By dividing the correctly obtained positive observations by all the projected positive observations, precision is calculated.

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

The ratio of correctly anticipated positive observations to all observations determines sensitivity or recall.

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

A weighted mean of recall and precision is produced by the F-measure. As a result, it makes use of misleading results and false negatives.

$$F1\,Score = 2 * \frac{(Recall * Precision)}{(Recall + Precision)}$$

Positives and negatives are used to calculate accuracy as shown below:

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

Table 2. Comparative numerical Analysis outcomes of suggested and existing Techniques on FNC-1 dataset

Performance Metrics	**CSI**	**CNN-LSTM**	**ELM**	**MBDFO-ELM**	**PSO and EDLM**
Accuracy	71.41	97.2	98.24	98.64	98.75
Precision	69	97	98	99.24	99.35
Recall	74	91	93.1	98.14	98.69
F-measure	71.41	93.9	95.67	99.21	99.48

Table 2 summarizes the results of the efficacy assessment involving the suggested and contemporary approaches for the provided FNC-1 dataset. The table demonstrates that the proposed PSO plus EDLM approach had the greatest detection accuracy when compared to existing false news detection techniques.

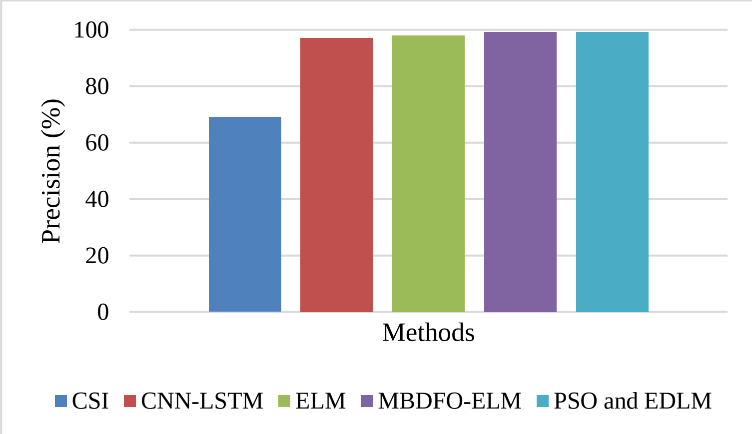

Fig. 5. Comparative values of Precision for the suggested and existing fake news detection models

Fig. 5. presents a comparison of the two suggested and existing false news detection methods in terms of precision. One may infer, after examining all the data, that PSO is more effective for severe dimensionality reduction since it significantly improves accuracy. Outperforming all other models, the offered model has an accuracy of 99.35%. The results show that the proposed PSO and EDLM approach performs more precisely than the currently in use categorization methods. By including the pre-processed data, PSO and EDLM perform better than other approaches.

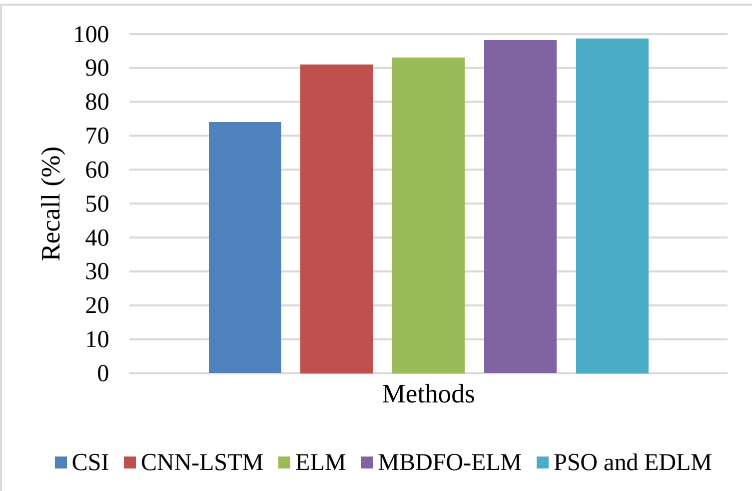

Fig. 6. Comparative Recall values of the suggested and existing fake news detection models

Fig. 6. displays the findings of a recall comparison analysis between proposed and current false news detecting techniques. The offered method is guaranteed to successfully classify any news as true or false by virtue of the statistical significance. The findings demonstrate that the proposed PSO and EDLM models have a 98.14% recall rate in comparison to the existing models. Comparing the recall rates of the various existing methods—CSI, CNN-

LSTM, ELM, and MBDFO-ELM—shows that the suggested work can detect phony stances more accurately than the alternatives, with recall rates of 74%, 91%, 93.1%, and 98.14%, respectively.

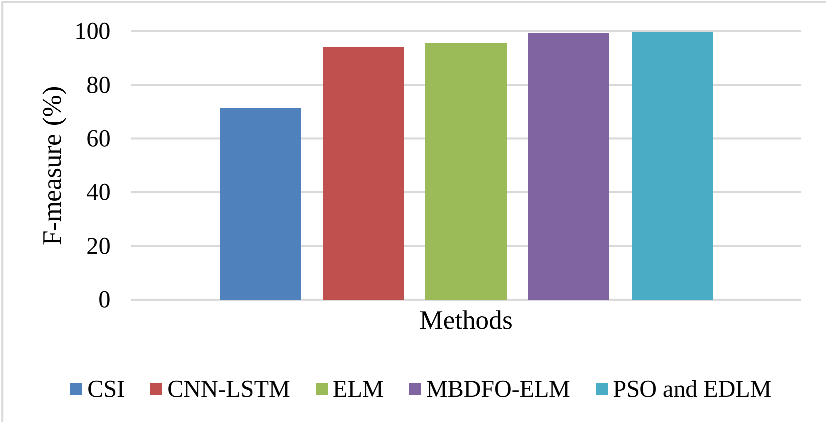

Fig. 7. Comparative F-measure values of the suggested and existing fake news detection models

The findings of an analysis of comparison of the most recent and recommended fake news detecting algorithms are shown in Fig. 7. in terms of F-measure. It is possible to assume that F1-score, accuracy, and recall have significantly improved. The suggested PSO-based selection of features method also significantly reduces the time required to complete a forecast. According to the results, the suggested PSO and EDLM technique works better than the currently used classification algorithms, with the F-value values of 99.48%. The EDLM often trains networks more quickly than CSI, CNN-LSTM, ELM, and MBDFO-ELM. It also has an efficient automatic feature extraction mechanism, which raises the f-measure value.

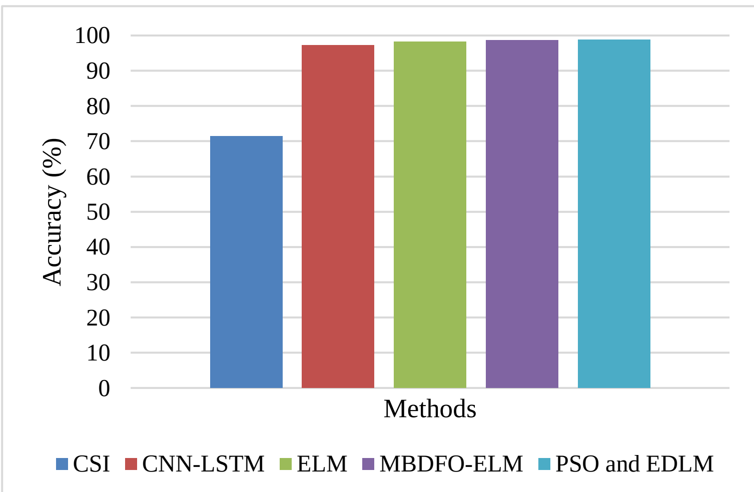

Fig. 8. Accuracy comparison results between the proposed and existing fake news detection model

The findings demonstrate that the proposed PSO and EDLM paradigm has a 98.14% recall rate in comparison to the existing models. The suggested work can provide superior fake stride detection outcomes than the existing approaches, as evidenced by recall rates of 71.41%, 97.2%, 98.24%, and 98.64%, respectively, while comparing the accuracy rates of the current techniques CSI, CNN-LSTM, ELM, and MBDFO-ELM. Fig. 8. displays the results of a comparison analysis into the effectiveness of the current and suggested fake news detecting strategies. The results show that accuracy is much lower (98.75%) whenever the characteristics are used without data cleaning or preparation. The dataset itself is likely to have a lot of distracting, redundant, and discontinuous data, according to this sign. The results show that the suggested EDLM approach can achieve higher accuracy values than current classification strategies.

4. CONCLUSION AND FUTURE WORK

Manually classifying news involves a thorough understanding with the ability to spot text oddities. In this work, we investigated the problem of identifying fake news articles using machine learning models and ensemble approaches. Furthermore, the majority of the news was covered by the data utilised in this study, which was derived from FNC news items rather than specifically classifying political material. Several issues in false news identifications are unresolved and require further research. For example, a critical first step in limiting the spread of fake news is recognising the critical elements involved in the news-dissemination process. It is possible to identify the primary actors behind the spread of false information by using EDLM techniques. The suggested model uses CNN, LSTM, and Bi-LSTM for this task, and the method known as PSO is used to choose the right features. Outcomes show that the outcomes are much improved when data preparation is used. Results from the suggested EDLM technique are better than those from the CSI, CNN-LSTM, ELM, and MBDFO-ELM models, with a success rate of 98.75%. The proposed model performs better when compared to current cutting-edge approaches. Additionally, k-fold cross-validation demonstrates the model's robustness. In the future, the model will be put to the test on sizable and intricate datasets, and it will be looked into if an ensemble of deep learning and machine learning algorithms can enhance performance.

References

1. Hunt Allcott and Matthew Gentzkow. (2017). social media and fake news in the 2016 election. Technical report, National Bureau of Economic Research, doi:10.1257/jep.31.2.211
2. Dominic DiFranzo and Kristine Gloria-Garcia. (2017). Filter Bubbles and Fake News. XRDS 23(3), 32–35. doi:10.1145/3055153
3. Alex Heath. (2016). Facebook is going to use snopes and other fact-checkers to combat and bury 'fake news'. http://businessinsider.com/ facebook-will-fact-check-labelfake-news-in-news-feed-2016-12. [Online; accessed 01-June-2017].
4. Konstantinovskiy, L., Price, O., Babakar, M. and Zubiaga, A. (2018) ''Towards automated factchecking: Developing an annotation schema and benchmark for consistent automated claim detection,'' arXiv:1809.08193. [Online]. Available: http://arxiv.org/abs/1809.08193,

doi:10.1145/3412869
5. Lazer, D. M. J., Baum, M. A., Benkler, Y., Berinsky, A. J., Greenhill, K. M., Menczer, F. Metzger, M. J., Nyhan, B., Pennycook, G., Rothschild, D., Schudson, M., Sloman, S. A., Sunstein, C. R., Thorson, E. A., Watts, D. J. and Zittrain, J. L. (2018) ‘‘The science of fake news,’’ Science, 359 (6380), 1094–1096, doi:10.1126/science.aao2998
6. Borges, L., Martins, B., and Calado, P. (2019) ‘‘Combining similarity features and deep representation learning for stance detection in the context of checking fake news,’’ J. Data Inf. Qual., 11 (3), 1–26, doi:10.1145/3287763.
7. Sridhar, D., Foulds, J., Huang, B., Getoor, L. and Walker, M., (2015) ‘‘Joint models of disagreement and stance in online debate,’’ in Proc. 53rd Annu. Meeting Assoc. Comput. Linguistics 7th Int. Joint Conf. Natural Lang. Process., vol. 1. Beijing, China, Jul., 116–125. [Online]. Available: https://www.aclweb.org/anthology/P15-1012, doi:10.3115/v1/P15-1012.
8. Lukasik, M., Srijith, P. K., Vu, D., Bontcheva, K., Zubiaga, A. and Cohn, T., ‘‘Hawkes processes for continuous time sequence classification: an application to rumour stance classification in twitter,’’ in Proc. 54th Annu. Meeting Assoc. Comput. Linguistics, vol. 2. Berlin, Germany, Aug. 2016, 393–398. [Online]. Available: https://www.aclweb.org/anthology/P16-2064, doi:10.18653/v1/P16-2064
9. Stab, C. and Gurevych, I. ‘‘Parsing argumentation structures in persuasive essays,’’ Comput. Linguistics, vol. 43, no. 3, pp. 619–659, Sep. 2017. [Online]. Available: https://www.aclweb.org/anthology/J17-3005, doi:10.1162/COLI a 00295
10. Pomerleau, D. and Rao, D (2017). Exploring How Artificial Intelligence Technologies Could be Leveraged to Combat Fake News. Accessed: Oct. 29, 2019. [Online]. Available: http://www.fakenewschallenge.org/.
11. Umer, M., Imtiaz, Z., Ullah, S., Mehmood, A., Choi, G. S., & On, B. W. (2020). Fake news stance detection using deep learning architecture (CNN-LSTM). IEEE Access, 8, 156695-156706, doi:10.1109/ACCESS.2020.3019735.
12. Davoudi, M., Moosavi, M. R., &Sadreddini, M. H. (2022). DSS: A hybrid deep model for fake news detection using propagation tree and stance network. Expert Systems with Applications, 198, 116635, doi:10.1016/j.eswa.2022.116635.
13. Wu, L., Long, Y., Gao, C., Wang, Z., & Zhang, Y. (2023). MFIR: Multimodal fusion and inconsistency reasoning for explainable fake news detection. Information Fusion, 101944, doi:10.1016/j.inffus.2023.101944.
14. Setiawan, R., Ponnam, V. S., Sengan, S., Anam, M., Subbiah, C., Phasinam, K., &Ponnusamy, S. (2021). Certain investigation of fake news detection from facebook and twitter using artificial intelligence approach. Wireless Personal Communications, 1-26, doi:10.1007/s11277-021-08720-9.
15. Sadeghi, F., Bidgoly, A. J., &Amirkhani, H. (2022). Fake news detection on social media using a natural language inference approach. Multimedia Tools and Applications, 81(23), 33801-33821, doi:10.21203/rs.3.rs-107893/v1.
16. Dixit, D. K., Bhagat, A., & Dangi, D. (2022). Automating fake news detection using PPCA and levy flight-based LSTM. Soft Computing, 26(22), 12545-12557, doi:10.1007/s00500-022-07215-4.
17. Choudhury, D., &Acharjee, T. (2023). A novel approach to fake news detection in social networks using genetic algorithm applying machine learning classifiers. Multimedia Tools and Applications, 82(6), 9029-9045, doirg/10.1007/s11042-022-12788-1.
18. Ramkissoon, A. N., &Goodridge, W. (2022). Enhancing the Predictive Performance of Credibility-Based Fake News Detection Using Ensemble Learning. The Review of Socionet work Strategies, 16(2), 259-289,doi:10.1007/s12626-022-00127-7.
19. Umer, M., Imtiaz, Z., Ullah, S., Mehmood, A., Choi, G. S., & On, B. W. (2020). Fake news

stance detection using deep learning architecture (CNN-LSTM). IEEE Access, 8, 156695-156706, doi:10.1109/ACCESS.2020.3019735

20. Li, C., Peng, H., Li, J., Sun, L., Lyu, L., Wang, L., & He, L. (2021). Joint stance and rumor detection in hierarchical heterogeneous graph. IEEE Transactions on Neural Networks and Learning Systems, 33(6), 2530-2542, doi:10.1109/TNNLS.2021.3114027
21. Vaideghy, A., &Thiyagarajan, C. (2023). Detection of Fake News Stance Employing Swarm Intelligence Based Feature Extraction and Ensemble Deep Learning. International Journal of Intelligent Systems and Applications in Engineering, 11(9s), 385-399.