Political Fake News Detection Farmwork in Social Networks Utilizing Hybrid Deep Learning Algorithms

Raed lrheim Mohmmad
College of Information Technology Engineering, AL-Zahraa University for Women, Iraq

Received: 21 Apr 2026 | Received Revised Version: 20 May 2026 | Accepted: 08 June 2026 | Published: 22 June 2026

Volume 08 Issue 06 2026 | DOI: 10.37547/tajssei/Volume08Issue06-09

Abstract

The speed at which false news spreads across online social media presents major threats to the authenticity of information and the trustworthiness of the public. As such, accurately detecting misinformation remains an ongoing challenge mainly due to the size and high-dimensionality of the textual data; the amount of redundant information found in all forms of texts; and the need for models to take into account semantic and contextual factors. In response to these challenges, this paper introduces a hybrid approach to detect false news based upon Grey Wolf Optimization (GWO); Convolutional Neural Networks (CNNs); and Long-Short Term Memory (LSTMs). Each component performs a unique function and adds to another component's performance. For example, GWO can be used as a meta-heuristic to select relevant features from large amounts of textual data and remove irrelevant or redundant information. Therefore, it can reduce the amount of computation required to train a model and improve its ability to generalize well. Once the relevant features have been identified through GWO, they are passed through CNNs that process the textual data hierarchically and capture local patterns within the textual data. Finally, LSTMs are applied to model sequential dependencies and long-range context within the data that was captured by CNNs. Experiments were performed on three datasets, including the BuzzFeed Political News Dataset, Random Political News Dataset, and the LIAR Benchmark Dataset. The experimental results showed that the proposed model has achieved better accuracy rates than other machine learning methods and optimized methods with accuracy rates of 89.2%, 94.8%, and 96.8% respectively.

Keywords: Fake News, social media, LSTM, CNN, Meta-Heuristic, GWO

Cite This Article: Mohmmad, R. lrheim. (2026). Political Fake News Detection Farmwork in Social Networks Utilizing Hybrid Deep Learning Algorithms. The American Journal of Social Science and Education Innovations, 8(06), 41–58. https://doi.org/10.37547/tajssei/Volume08Issue06-09

1 Introduction

The advent of the Internet has fundamentally changed the way individuals are able to interact and be in contact; how quickly we can get our hands-on information; and most importantly how accessible this information is. The speed at which the Internet transmits data and it's availability/affordability has significantly altered how we consume news today. Today many consumers prefer to receive their news through social media or other digital sources instead of reading it in a newspaper [1]. In turn this has given social media enormous power as a means of disseminating information that shapes public opinion and influences global events. At the same time as social media has become a source of information, there has been a staggering increase in the spread of false information -- particularly the deliberate spreading of fake news (misleading news stories) -- that threatens the very fabric of society by eroding public confidence and stability [2]. Misleading reports such as those published in "fake news" stories often operate from the perspective of the author's agenda rather than objective reporting. These types of publications can manipulate public opinion and ultimately harm society, as exemplified by the psychological impact of COVID-19 misinformation and its influence on voter perception in the 2020 US Presidential Election [3].

Misleading information is spreading rapidly with respect to the speed at which people are getting access to various types of information, specifically social media. There is a great deal of information available online that ranges from simple text messages to image files; numerical data to rumors. As a result, determining what is real and misleading will make identifying misinformation very difficult. Misleading information, generally speaking, includes all forms of false information including, but not limited to, "fake news", rumor, spams, and disinformation. Misleading information may appear in many different formats (textual, visual, numerical, etc.) [5] . Given the broad impact of misleading information on public opinion and democratic processes there has never been a greater need for developing effective methods of detecting misleading information.

The effects that Fake News have had upon both society and economics have been substantial. Those who maliciously wish to create havoc can purposefully distribute false news to harm reputations, to alter consumers' purchasing decisions, and/or to shape voters' choices regarding political elections. Because of this issue's increasing severity, there is great interest among researchers to develop ways to automatically detect fake news through the use of machine learning and artificial neural network (deep) learning models [6], [7] so they may be able to find misleading content through content analysis and contextual cues garnered by social interactions surrounding the news. As such, fake news detection has emerged as a key area of study; it will serve as an important means to ensure information accuracy at a time when the velocity and breadth of information dissemination has never been greater [7], [8].

Although a large amount of literature exists regarding the detection of false news stories, there are some problems with the existing solutions. Most current methods have difficulty dealing with very high dimensional data that include text information as well as redundant features. This makes it difficult for these types of models to be run efficiently while at the same time maintaining acceptable levels of accuracy. A number of previous studies failed to use a combination of semantic feature extraction techniques and contextually aware sequential modeling, which has limited their ability to recognize patterns in local as well as long range dependencies within the content of the news articles. Additionally, because most of the models developed to date do not contain a method of selecting features intelligently this limits the models' ability to generalize or be robust over various data sets [9],[10].

The purpose of this paper will be to design a hybrid fake news detection system using Grey Wolf Optimization (GWO), Convolutional Neural Network (CNN), and Long Short Term Memory (LSTM) network. In this study, GWO was designed for the purposes of feature selection, CNN was used for extracting semantic features from images, and LSTM were used for contextually aware sequence modeling. The main goals of this study are to increase the accuracy of classifying true/false news, reduce the computation required for processing the data and increase the generality and robustness of the classification results when trained over multiple data sets.

1.1. Research Contributions

1. Hybrid GWO-CNN-LSTM Model: The hybrid GWO-CNN-LSTM model unifies a metaheuristic based feature selection method (GWO) with deep learning techniques (CNN, LSTM) in order to improve the detection of false information.

2. Feature Selection via Optimized use of GWO: GWO was used as an optimization technique to reduce the number of selected discriminative features, thereby increasing the efficiency of this process.

3. Learning of Features and their Contexts: Local semantic patterns are extracted by CNN's while LSTMs learn long range contextual relationships which enables the model to accurately classify.

4. Robust Multi-dataset Validation: To demonstrate its potential, the hybrid GWO-CNN-LSTM model has

been evaluated against several benchmarks across three large scale datasets.

2. The Proposed Method

This section outlines our proposed GWO-CNN-LSTM model for detecting false information. We use the combination of (GWO-CNN-LSTM), to handle the three major issues with classifying false information as follows: (i) the high number of dimensions associated with text data, (ii) that many redundant and unnecessary characteristics exist in the data, and (iii) how to capture both semantic and contextual information from the content of news stories. To reduce dimensionality, and improve the overall quality of the preprocessed data; we utilize GWO [11] in the preprocessing phase to identify the most relevant attributes from the document vector representations. These identified attributes are then forwarded to the CNN-LSTM classifier. In this classifier, CNN [12] is utilized to find the most important semantic patterns in the attribute representations; while LSTM [13], is utilized to model sequential and context dependent relationships found in these patterns. The final classification result will be either that it was classified as a piece of false information, or a piece of true information. Therefore, by using GWO-CNN-LSTM, we combine optimization techniques, and deep learning approaches into one pipeline to enhance detection accuracy and generalization performance. Figure 1 explains the proposed GWO-CNN-LSTM Framework for Fake News Detection.

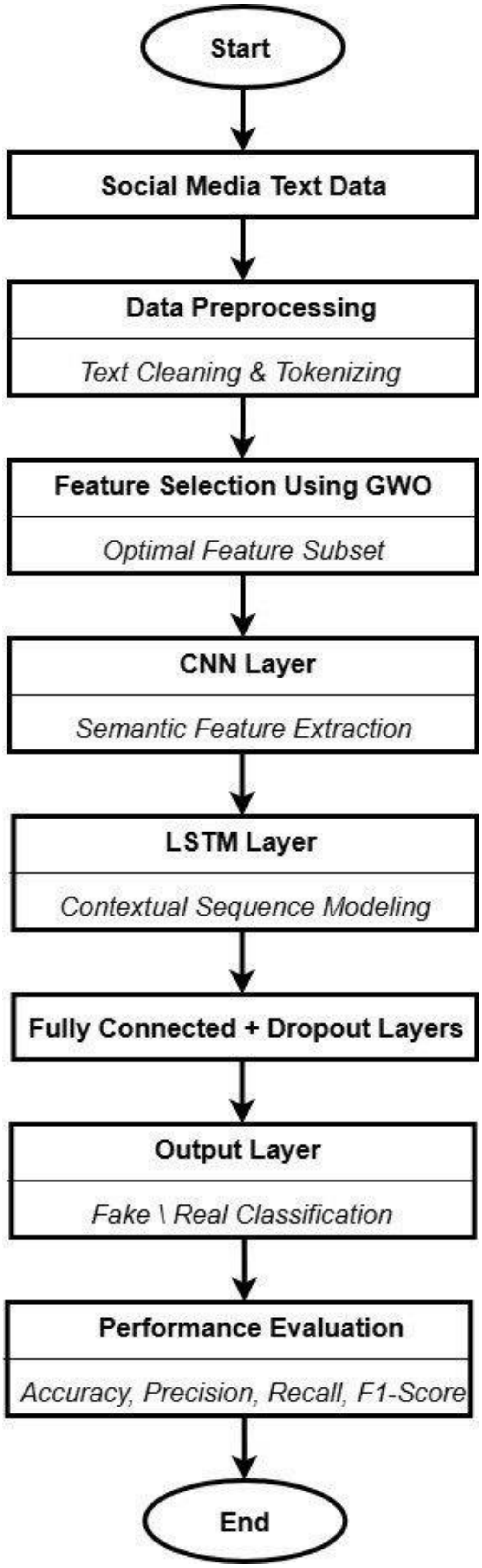


Figure 1: The proposed GWO-CNN-LSTM Framework for Fake News Detection

2.1. Data Preprocessing

The way that text data can be represented has a great influence on the quality of the last model developed through supervised learning and/or deep learning algorithms to classify text. Machine Learning or Deep Learning algorithms cannot process raw text. Raw text must therefore be transformed into some form of numerical representation so that it can be processed by machine learning/deep learning algorithms. Preprocessing is thus viewed as one of the foundational stages in the proposed framework.

Textual information that was collected from Social Media Datasets were first loaded, preprocessed and separated into individual word frequency counts. The resulting processed text was next transformed into Document Vectors, where each row represents a single News Document, and each column represents a separate Feature(s). The transformation changed unstructured Text Data into a Structured Numerical Format that could now be used for Optimization and Classification. Due to the fact that there may be thousands of different Features/Dimensions within the Text Data, Document Vectors tend to have many Dimensions. Thus, Preprocessing does two things; it processes the data so that it is ready to learn, and it creates the environment necessary for Feature Selection. For example, in the Proposed Framework, Document Vectors will be utilized with Grey Wolf Optimizer (GWO), to choose the best possible subset of Dimensional Features. The Optimal Subset of Dimensional Features will then be used as Input to the CNN-LSTM Model for Effective Detection of Fake News.

2.2. Feature selection

In this study, feature selection was implemented within the proposed framework with the goal of removing all irrelevant, redundant and poorly discriminating attributes from the input vector so they could not be used as relevant information for the classification task. Document vectors can include a high amount of features. These additional features increase the processing time associated with machine learning, slow down the learning process of the model and negatively impact the models ability to generalize. Therefore, selecting the optimal number of features is critical to improving both speed and classification accuracy.

The GWO metaheuristics algorithm was employed in this research for feature selection. The GWO is a metaheuristics optimization technique based upon the social structure of gray wolves. Each search agent in the GWO represents a candidate set of features; therefore, it is necessary to evaluate the quality of these agents through the use of a fitness function based on their corresponding classification performance. As a result of iterative application of the best solutions found thus far, the GWO will converge toward an optimum or nearly optimum feature set.

Through the employment of GWO, the proposed methodology efficiently searches for the most discriminant features. Thus, there is no need for a comprehensive search of the complete feature space. Instead, the methodology uses an efficient and intelligent search strategy to find the most relevant features. This leads to a reduction in computational expense, decreases the training time and improves the performance and reliability of the subsequent CNN-LSTM classification model.

2.3. Framework Description

The proposed method has been organized into a two-phase process; i.e., first phase - optimizing feature generation, second phase - applying deep learning techniques for classification as part of the GWO-CNN-LSTM system. As such, the dataset was pre-processed to produce Document Vectors (Features) from the input feature space. The Features obtained in this step were fed into the GWO Algorithm. In the GWO Algorithm, each agent (wolf) is initialized with a random set of wolves. Each wolf will be used to represent a Candidate Subset of Feature(s). The Quality of each Wolf (Agent) is determined by a Fitness Function, whose basis is the Classification Performance.

In this research, a two-stage approach has been developed. The first stage includes the optimization of Features for generating, while the second stage involves applying deep learning methods for classification in a way that it is related to the GWO-CNN-LSTM system. Accordingly, the data was pre-processed to generate Document vectors (Features) from the input feature space. After obtaining the Features in this step, they were inputted into the GWO Algorithm. At this point of time, each agent (wolf) within the GWO Algorithm was randomly assigned a different location. Each wolf would represent a candidate subset of Feature(s). The quality of each wolf (agent), will be represented through a fitness function. It's worth noting that the fitness function represents the Classification Performance.

The fitness of each candidate solution is calculated as follows:

$$Fitness = Acc - \lambda \cdot \frac{|S|}{|F|} \quad (1)$$

Herein, acc denotes the accuracy of classification of samples, s denotes the selected feature subset, f denotes the total number of Features and λ is a weight parameter that determines the trade-off between accuracy and reduction of Features. By doing so we ensure that the selected Features are both informative and compact. During the optimization process, the positions of the agents are updated based on the best current solutions (alpha, beta and delta). The position update mechanism is as follows:

X(t + 1) = (X alpha + X beta + X deltat) / 3 (2).

Where X(t + 1) denotes the updated position after iteration t + 1 and X alpha, X beta and X deltanote the three best solutions. The distance between an agent and optimal solution is calculated as:

D = |c \cdot X p - xt| (3),

And position update is defined as:

$$\vec{X}(t+1) = \vec{X}_p - \vec{A} \cdot \vec{D} \quad (4)$$

Where a = 2ar1 - aand c = 2r2with r1,r2 ∈ [0, 1]. This mechanism balances exploration and exploitation to identify the optimal feature subset. Following feature selection, the optimized Features are entered as inputs to the cnn-lstm classifier. In this stage, the cnn extracts high-level semantic Features via convolution:

$$h_i = \sigma(W \cdot x_{i:i+k} + b) \quad (5)$$

Where w is the convolution filter, xi : i + k is a local input segment and σ is the activation function. Then a pooling operation is applied:

$$h_{max} = \max(h_1, h_2, \ldots, h_n) \quad (6)$$

To retain most informative Features and reduce dimensionality.
The resultant feature maps are reshaped into sequential form and provided to the lstm component that models contextual dependencies using:

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_t, h_t = o_t \cdot \tanh(C_t) \quad (7)$$

Where ct and ht denote cell state and hidden state respectively, and ft, i t and ot represent gate mechanisms. Finally, fully connected layer performs classification of news instances into fake or real categories. This integrated GWO-CNN-LSTM framework enables effective selection of Features, representation learning for semantics, and modeling of contextual relationships therefore leading to improved accuracy, reduced computational complexity and enhanced generalization in detection of fake news.

The input is social media data in the form of text documents.

The output will be a classification of real vs fake news.

Step 1 – Data Pre-Processing:
i. Reads the raw data set.
ii. Cleans up the document by removing unnecessary characters from the text, separating words for frequency counting.
iii. Frequency count of all words.
iv. Generates document vector for each document.

Step 2 - Feature selection using GWO
v. Sets GWO parameters (a, c, A).
vi. Generates an initial population of wolves.

vii. Treats each wolf as a potential feature subset for our classifier.
viii. Evaluates how well each wolf performs at classifying our documents (fitness function).
ix. Chooses top 3 performing wolves (Alpha = Best, Beta = Second Best, Delta = Third Best).
x. Updates position of each wolf by taking the average of their current positions and positions of the two most successful wolves (Position Update).
xi. Checks whether to stop the process. If so, ends. Otherwise continues with next step.
xii. Picks the best feature set.

Step 3 – CNN-LSTM Classifer
xiii. Lays out inputted features from previous steps through CNN.
xiv. Uses Convolution and Pooling operations to extract semantic features from images.
xv. Passes extracted semantic features through LSTM model that looks for patterns and relationships in sequential and contextual dependencies.
xvi. Adds additional dense layers after LSTM model to further improve performance.
xvii. Adds dropout layers to prevent over-fitting.
xviii. If there are additional layers, repeats above process until all specified layers have been added.

Step 4 – Classification
xix. Presents final classification results using fully connected layer.

1. xx. Outputs classification result (Real News/Fake News).

In detecting fake news, not all features have the same value and each feature has a specific importance. The importance of the features is precisely specified by weighting each of them. For this purpose, the following functions have been used:

1.3.1. Generating Function

Generating functions create a set of candidate feature sets for the first generation (population) in the GWO algorithm; therefore, they provide the start point for the search process. The purpose of GWO is to determine which are the most relevant features with respect to their weights. Therefore, the generating function produces all possible sets of features that will later be assessed. Quality of the starting population influences the effectiveness of subsequent optimizing operations. If there is an effective diversity within the population at its beginning, then it can better assess the search space for highly-informative features for detecting fake news.

1.3.2. Evaluation Function

The evaluation function is used to measure how good each candidate feature subset created by GWO is. The evaluation function uses the classifier's ability to predict the correct class label (accuracy) as its "fitness". Therefore, the candidate subsets that produce higher fitness values will be those that provide the best classifications.

Ultimately, the goal of the evaluation function is to direct the search toward finding feature subsets that enhance the accuracy of detecting fake news. Given that different features may have varying degrees of contribution to the classification task, the evaluation function allows for distinguishing between subsets of features with high contributions to the classification problem from those that have little or no impact on it. Hence, the evaluation function has significant roles in both reducing the number of features and increasing their effectiveness for classification.

1.3.3. Updating the Position of Agents

The positions of the search agent (wolf) during each iteration of the GWO is modified by the current best solution. The modification of position is determined using the Alpha, Beta and Delta wolves. These represent the three best candidate solutions at this point in time. The remaining wolves modify their position relative to these leaders. The ability of the algorithm to explore the Feature Space as well as exploit the best current solution for classification is achieved through this method. In this paper, this step is significant in that it progressively enhances the quality of the selected feature subset(s) while directing the search toward better combinations of textual features. As the number of iterations increase, the algorithm will converge to a subset that increases the accuracy of the CNN-LSTM classifier.

1.3.4. Termination Condition

Termination Condition for GWO Algorithm determines when the GWO algorithm terminates its search for optimal Feature Subset. In this research, termination can occur when the total iteration count reaches a predetermined value (i.e., Maximum Iteration) and/or when the search appears to converge on an optimal solution; and/or when the target Classification Accuracy is obtained. A termination condition allows one to bound the amount of time spent executing the Optimization Process. It also prevents excessive computation from being performed beyond what would be required to find a suitably acceptable feature subset. Once a satisfactory set of Features is identified, they will be transferred to the subsequent Classification Stage.

1.3.5. More Details of the GWO Algorithm in the Proposed Method

GWO as well as other metaheuristics starts by generating randomly the initial set of candidate solution (the population). After defining the control parameters (A, C, and a) and their values, GWO calculates the fitness of each individual. Individuals are ordered based on their fitness and they are classified into three types: alpha, beta, and delta. At each iteration, GWO updates the positions of the wolves, until reaching the last iteration. As explained in this paper, the proposed method has another important stage before updating the parameters. It is named averaging stage, during it, the position of some wolf can be averaged with those of other wolves, depending on the type of wolf; this improves the precision of the research. The averaging operation brings weak solutions closer to better ones and therefore enhances the chances of finding the global optimum characteristics. Finally, the vector "a" plays an essential role in balancing exploration and exploitation through changing over time; it avoids trapping the search in local minima and enables GWO to select relevant features efficiently for detecting fake news.

1.3.6. How to Use CNN-LSTM in the Proposed Method

In the last step of the proposed methodology, once the most significant features have been identified through the GWO algorithm (global optimization), they are sent to a CNN-LSTM classifier for classification. A combination of two architectures, CNN and LSTM (deep learning) has been chosen because both offer several benefits. The CNN will identify relevant local semantic patterns within the feature representation using convolution and feature mapping; it can therefore find structural elements in the input data that could determine if an article was true or false. Then the CNN will send its representations to LSTM to analyze the sequential dependency and contextuality of these structural patterns. LSTMs are specifically developed to retain information across long sequences so they can detect contextual dependencies which might otherwise go unnoticed by the CNN. Each dataset is processed independently as well. For training, 70 % of the data is used while the remaining 30 % is reserved for testing. To avoid overfitting, drop out layers were added to the model. In addition to dense layers, a fully connected output layer was included to perform a classification of the news articles into their respective target categories. Drop out works by removing temporary some units during training. It enhances robustness and reduces the risk of over-training. At the end of this process, the output layer will classify the news articles into one of the target categories. Therefore, the CNN-LSTM classification stage complements the GWO feature selection stage with a highly powerful classification engine combining semantic extraction and contextual analysis.

2. Results and Discussion

This section outlines how the GWO-CNN-LSTM Model will be tested. Testing is performed by running a number of experiments on a Python 3 platform. As part of the testing, the process combines the use of Grey Wolf Optimization for Feature Selection with Deep Learning Classifiers that combine CNN and LSTM architectures.

In order to evaluate whether or not the GWO-CNN-LSTM Model is effective as a classification tool, the classification tool's performance was compared to other nine well known classification tools. They included SMO, J48, Ridors, Gradient Boosted Trees (GBT), Support Vector Machines (SVM), Naive Bayes (NB), Decision Trees (DT), GWO alone and Simplified Swarm Optimization (SSO). The choice of these other classifiers provided an opportunity to compare the proposed method with a wide variety of traditional machine learning techniques, ensemble based techniques and optimization techniques.

Testing of the proposed classifier was carried out on a variety of benchmarking data sets, and the classification performance of the proposed classifier was measured in a multi-faceted way using the four basic performance measures: Accuracy, Precision, Recall and F1-Score [14], [15], [16], [17], [18],[19],[20]. The results from testing are presented and analyzed using a combination of tables, plots and confusion matrix. This allows a fair comparison of the performance of the proposed classifier when compared to the performance of all of the other classifiers being compared.

Ultimately, the objective of this section is to show that the GWO-CNN-LSTM Classifier performs better than all of the other classifiers being compared in terms of Classification Accuracy and Robustness and Generalization Capability over all of the Data Sets used; as well as showing that it can perform feature selection automatically which will result in improved computational efficiency.

2.1. Dataset

In order to determine how well the suggested GWO–CNN–LSTM framework performed, we conducted experiments with three popular real world fake news datasets that are commonly utilized. We chose these datasets to test our models ability to generalize as well as to demonstrate its robustness by utilizing different distributions of data.

Each dataset was split into two groups; one group contained 70% of the data for use in training the models, and the remaining 30% of the data was set aside for testing. Each dataset was pre-processed and then converted to vector representations (document vectors) prior to optimizing them via GWO based feature selection and finally feeding the resulting optimized document vectors through the CNN-LSTM classifier.

2.1.1 BuzzFeed Political News Dataset

BuzzFeed compiled its BuzzFeed Political News data set from both real and fake political news stories that were released into the public domain during the 2016 U.S. Presidential Election [21]. The time frame of the data collection/analysis was roughly nine months; therefore it represents a reliable benchmark for evaluating various methods of detecting misinformation.

The preprocessed and feature-extracted data set has 38 features; however these were further reduced through use of the GWO algorithm to find an optimal subset of the most informative features to be used with the classifier.

2.1.2 Random Political News Data Set

This random political news data set was developed through the collection of fake news articles from known unreliable sources and real news articles from reliable sources such as Business Insider [22]. As a result, this data set will provide a good testbed for developing effective methods to evaluate the accuracy of a classifier's results based on classifying political news.

With regard to preprocessing, there are 39 features contained within the data set. These features were also optimized using GWO to eliminate redundant features, thereby improving the effectiveness of the learning process within the proposed framework.

2.1.3 LIAR Benchmark Data Set

LIAR is a publicly accessible benchmark data set that is specifically designed for testing models that detect fake news [23] . A total of 12,836 short statements have been included in the data set, representing a variety of realistic situations. After preprocessing, the data set is described using 42 features, which are then subject to GWO-based feature selection prior to being fed into the CNN-LSTM model.

2.2. Implementation tools

A GWO-CNN-LSTM based framework was created using the Python programming language because it is flexible and has many libraries for both Machine Learning (ML) and Deep Learning (DL). This creation uses metaheuristics for optimizing features with DL for creating a single system for the classification.

Both the Development/Execution of the Model and the Environment of the Model were done on Jupyter Notebook & Google Colab, as these two offer very good environments for testing/experimenting/training models. Google Colab was specifically chosen in order to utilize its Cloud Based Infrastructure to train a deep neural network like CNN or LSTM, as well as take advantage of its GPU & TPU for faster processing times.

All three parts of the Model were built utilizing some of the most popular Libraries/Frameworks in both ML and DL; TensorFlow, PyTorch for building the DL, and Matplotlib for visualizing the Experimental Results.

Both Libraries help build/train/performance test your Models efficiently.

Google Colab has several advantages. It allows you to have free access to High Performance Hardware. It also seamlessly integrates with other Cloud Storage Services such as Google Drive. And lastly, you can work collaboratively by having multiple people working on the same project at the same time. Lastly, Google Colab gives you an already configured setup that includes all of the Libraries you will be needing to install, so there is no need to manually install them. Therefore, you are able to reproduce the exact same results whether you run the code locally or remotely. Scientifically, this environment makes it easy to execute this Framework, removes limitations that could come from Computing Resources, and allows for Scalable Experimentation when trying to detect Fake News.

2.3. Evaluation method

The proposed model was evaluated by presenting the results it achieved when applied to a clutter matrix in comparison with nine additional algorithms. As such, prior to describing the comparisons made in this study using the proposed algorithm's results, a description of the confusion matrix will first be provided. The confusion matrix represents one of the simplest and most direct measures that may be employed to determine both accuracy and precision of a model as well as provide an evaluation metric to compare models with regard to classification problems that result in two or multiple classifications.

2.3.1. Confusion Matrix

The confusion matrix [24], an example of which is given in Figure 2, is a two-dimensional table (actual and predicted) and a set of classes in both dimensions. The actual classifications are in the columns and the predictions are in the rows.

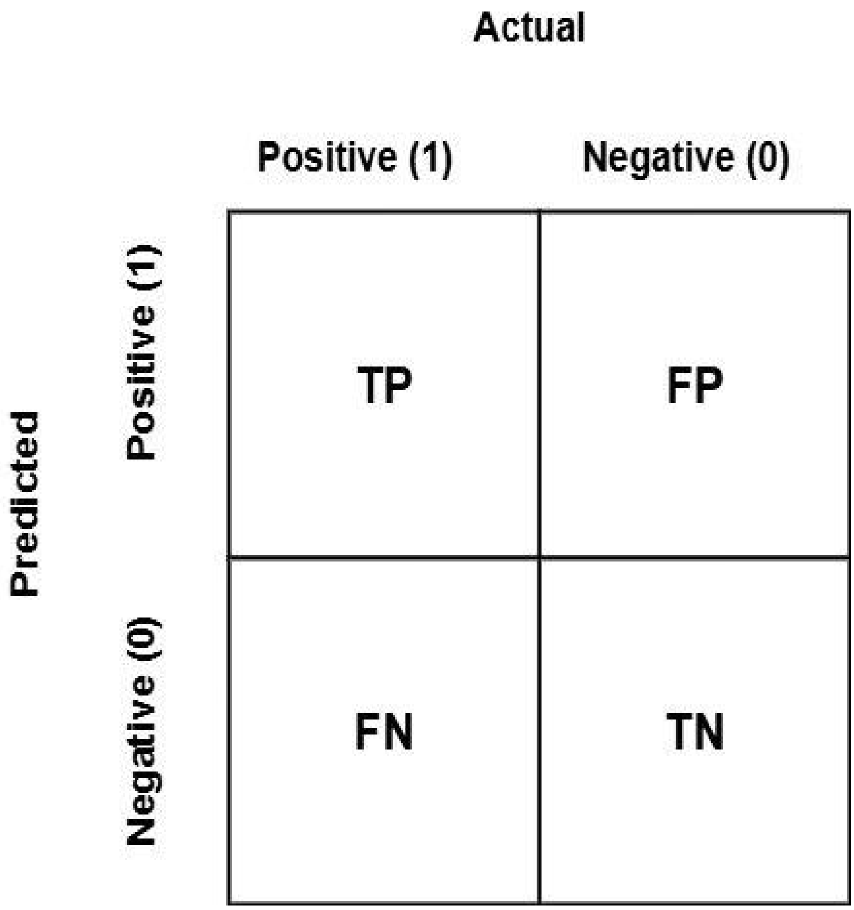


Figure 2. The general representation of the confusion matrix

This Matrix is not an individual Performance Metric, however most Performance Metrics are derived from this Matrix. There are Four (4) Elements within this Matrix: TN, TP, FP & FN, each described below:

TN: Is the Number of Records that were classified by the Algorithm to be Negative and that record was actually Classified as such.

TP: Is the Number of Records that were Classified by the Algorithm as being Positive and that Record was Actually Classified as such.

FP: Is the Number of Records that were Classified by the Algorithm to be Positive and the Record was Actually Classified as Negative.

FN: Is the Number of Records that were Classified by the Algorithm to be Negative and that Record was Actually Classified as Positive.
In order to Evaluate the Proposed Method and Compare it with the 9 Algorithms Previously Mentioned, Accuracy, Precision, Recall, and F-Criteria will be Utilized. These Criteria are Defined Below.

2.3.2. Evaluation Criteria

In this paper, four criteria have been used to evaluate the proposed GWO-CNN-LSTM framework these are Accuracy, Prediction, F1-Score and Recall.

i.Accuracy:

Accuracy is the number of correct predictions received from the model over the total number of predictions used in classification problems and is calculated from equation 8.

Accuracy=(TP+TN)/(TP+FP+FN+TN) (8)

The accuracy criterion is represented in Figure 3by the confusion matrix.

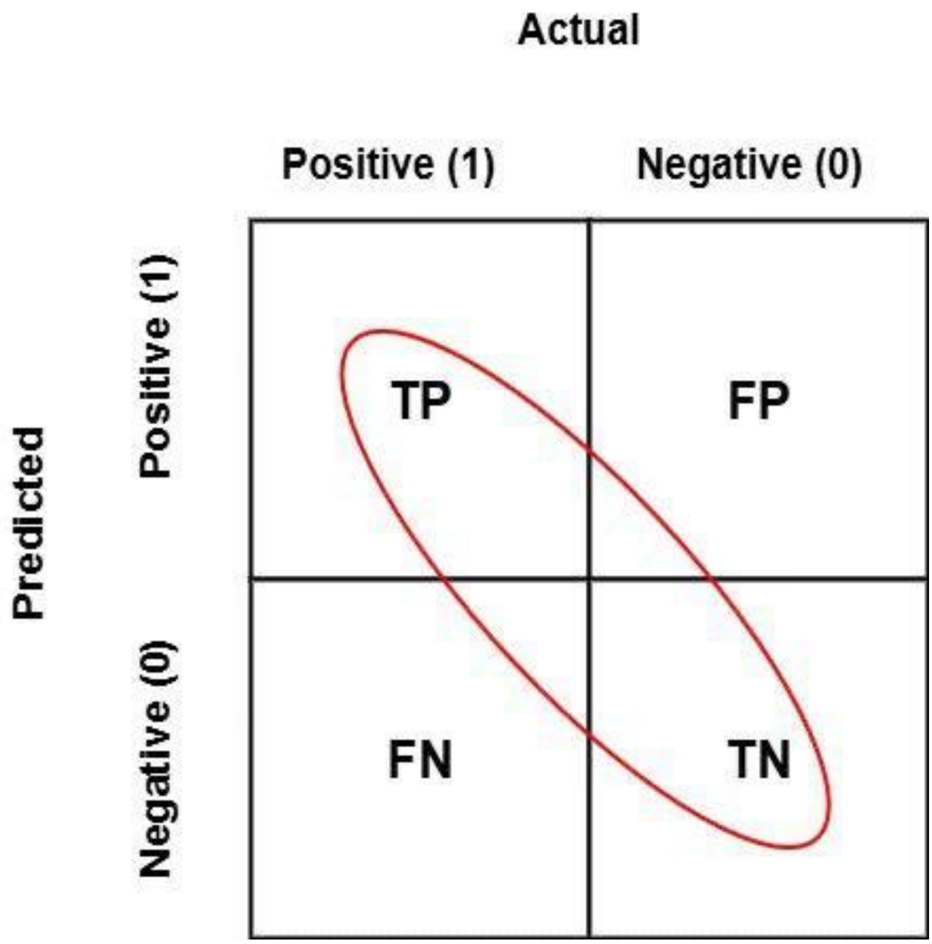


Figure 3: The representation of Accuracy metric using a confusion matrix.

ii. Precision:

The confusion matrix can also be used to describe accuracy. This metric determines how much fake or real news is correctly predicted by the proposed model. To calculate accuracy, equation 9 is used.

Precision=TP/(TP+FP) (9)

The Precision criterion is represented in Figure 4 by the confusion matrix.

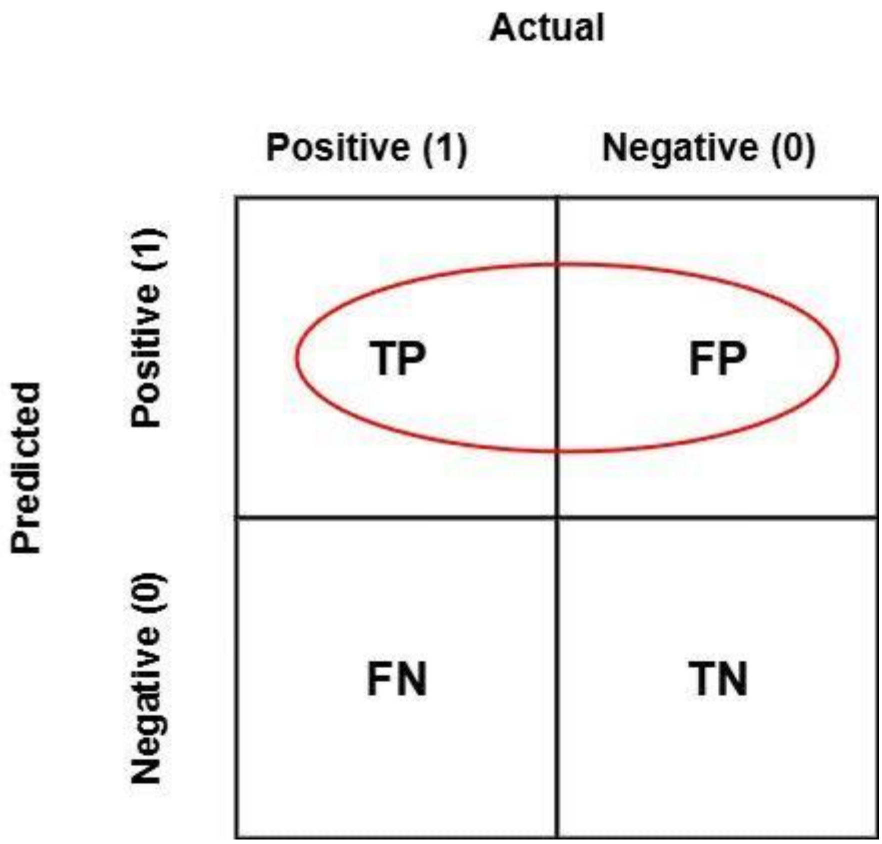


Figure 4: The representation of Precision metric using a confusion matrix.

iii. Recall:

Readback indicates how much of the fake news was predicted as fake news by the classification model. Equation 10

is used to calculate this measure.

Recall=TP/(TP+FN) (10)

The Recall criterion is represented in Figure 5 by the confusion matrix.

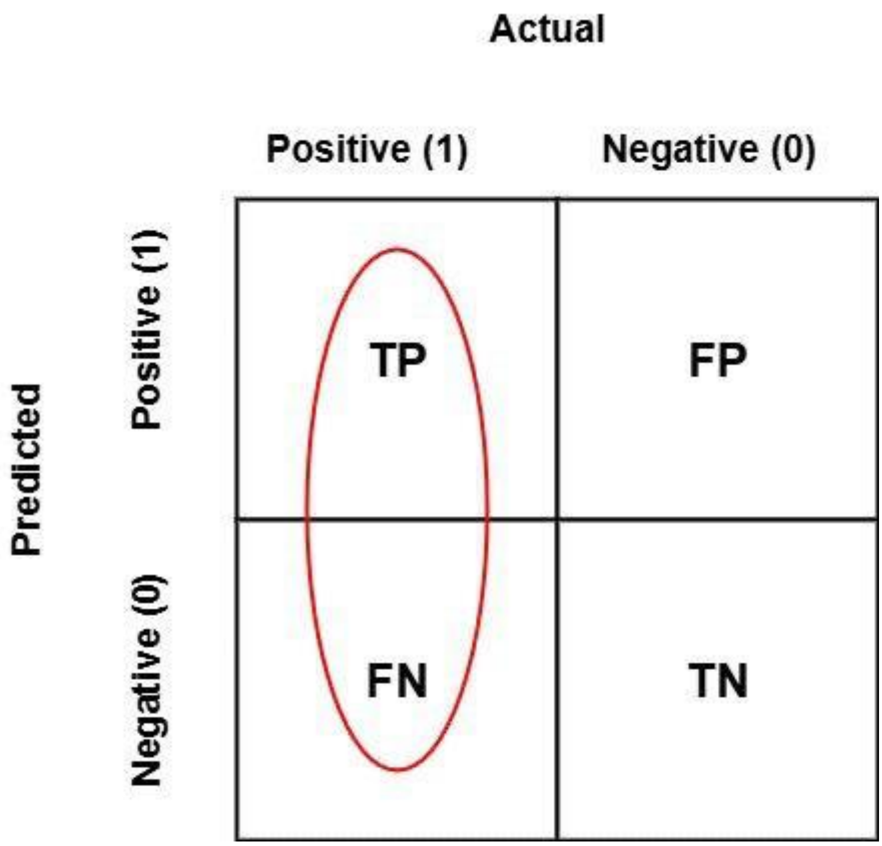


Figure 5: The representation of Accuracy metric using a confusion matrix.

iv. F1-Score:

There is another important parameter called F-criterion which is used to evaluate the performance of classifiers and is obtained by combining two parameters: precision and recall. "F-criterion" is defined as follows:

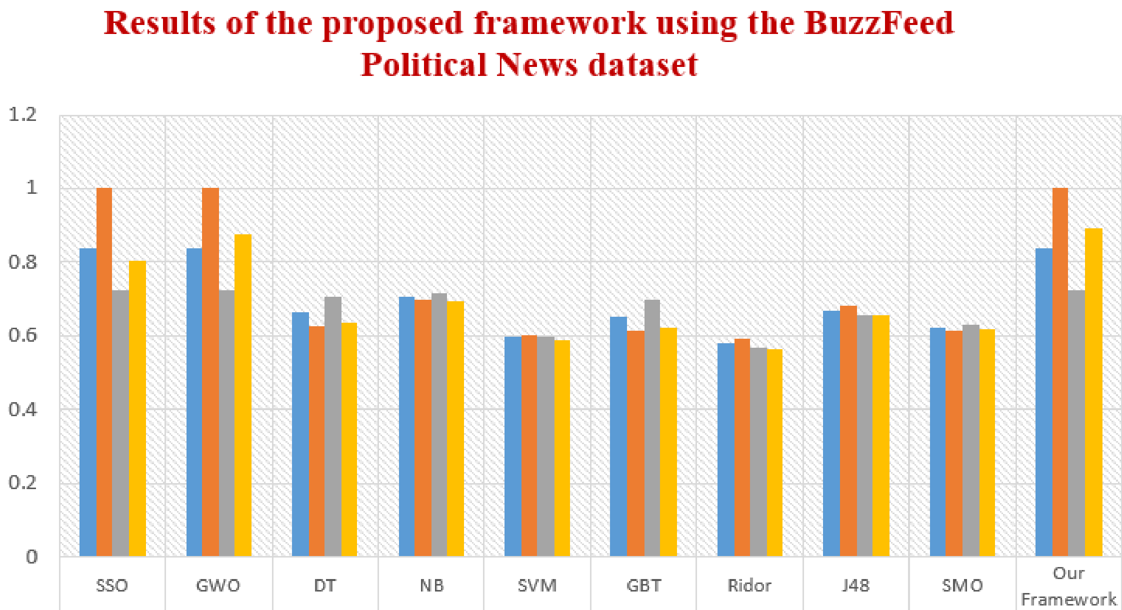
F1-Score = (2*Recall*Precision)/(Recall+Precision) (11)

The values of these four criteria for the proposed method and the 9 algorithms being compared are given below.

2.4. Results of the proposed method for BuzzFeed political news

The results of implementing the proposed model on the BuzzFeed political news dataset and comparing it with the 9 algorithms being compared are given in Table 1.

Table 1: Results of implementing and comparing the proposed method for the BuzzFeed political news dataset



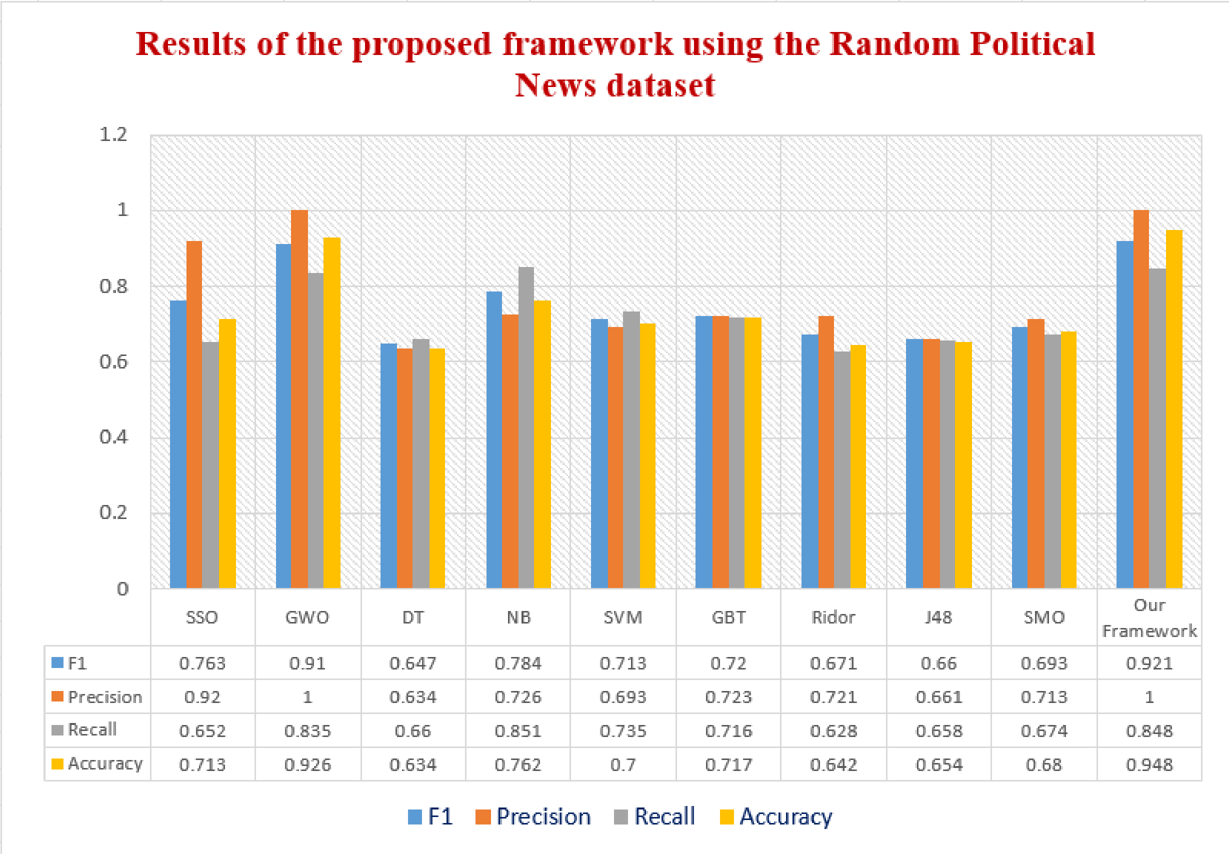
	SSO	GWO	DT	NB	SVM	GBT	Ridor	J48	SMO	Our Framework
F1	0.839	0.838	0.664	0.706	0.597	0.653	0.579	0.668	0.622	0.839
Precision	1	1	0.626	0.697	0.6	0.614	0.592	0.681	0.616	1
Recall	0.723	0.722	0.707	0.716	0.595	0.698	0.567	0.655	0.629	0.722
Accuracy	0.803	0.875	0.634	0.696	0.59	0.621	0.562	0.655	0.619	0.892

As the results in the table show, the GWO-CNN-LSTM method has the highest accuracy value, 0.892, on the BuzzFeed political news dataset. In this dataset, the worst accuracy of 0.562 is obtained using Ridor. In terms of accuracy, the SSO algorithm has the highest accuracy among the 9 algorithms, while again, Ridor has the lowest accuracy. In the recall metric, the SSO, GWO, and GWO-CNN-LSTM methods have the highest performance with an accuracy value of 1. In terms of F-criterion, the highest value is obtained by the SSO and GWLSTMNN algorithms (0.839) while the lowest value is obtained by Ridor (0.579).

5-4- Results of the proposed method for random political news

For this dataset, the results of running the proposed method and the 9 algorithms compared are shown in Table 2.

Table 2: Results of running and comparing the proposed method for the random political news dataset

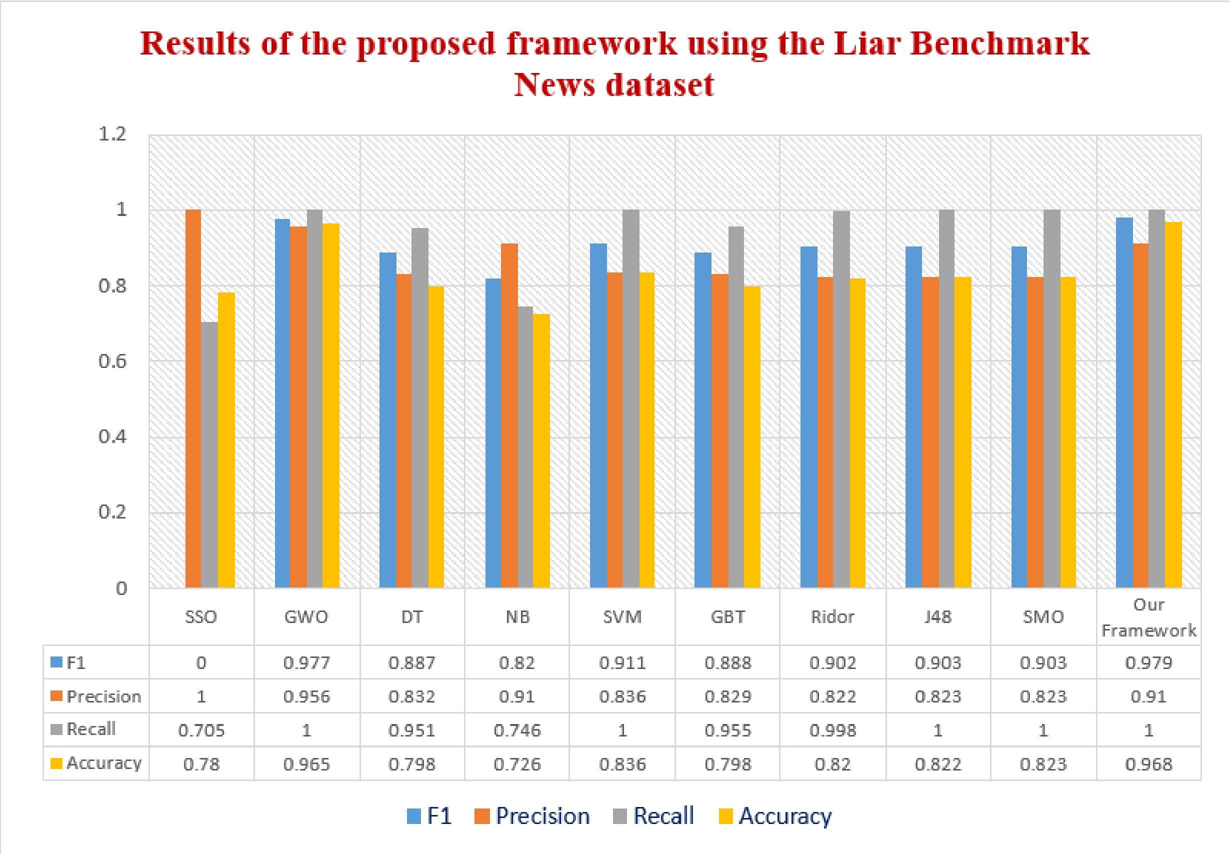


As can be seen in Table 2, the proposed GWO-CNN-LSTM method for the random political news dataset similar to the results obtained from the BuzzFeed political news dataset provides the best results in terms of all evaluation criteria except recall values in this dataset due to the proposed efficient fitness function that does not consider only one objective. The lowest values for precision, accuracy and f-score are obtained by Decision Tree.

6-4- Results of the proposed method for Liar Benchmark

Similar to the previous dataset, the proposed method was also implemented on the Liar Benchmark dataset and its results in comparison with 9 other algorithms are listed in Table 3.

Table 3: Results of implementation and comparison of the proposed method on the Liar Benchmark political news dataset



As shown in Table 3, the proposed GWO-CNN-LSTM framework for the Liar Benchmark dataset gives the best results in terms of all the evaluation indicators presented except accuracy in this set due to the proposed accepted subjectivity that simultaneously manages different efficiency. The highest accuracy value is obtained from the proposed SSO (1.000). The lowest accuracy and f-score are obtained by Naïve Bayes. In terms of accuracy, the lowest value is obtained by Ridor (0.822).

3. conclusion, limitations and future works

3.1. Conclusion

Fake news has been spread rapidly by means of social networks; it is therefore imperative to develop an efficient and reliable system for detecting fake news. A new Hybrid GWO-CNN-LSTM method has been developed to solve the above problems.

In recent years, many researchers have investigated various machine learning algorithms to detect fake news, but they are often limited by their own characteristics such as dealing with big data, redundant features and lack of consideration of semantic and context information.

A hybrid model which includes a combination of Grey Wolf Optimizer (GWO) algorithm and a deep learning architecture of Convolutional Neural Network (CNN) for extracting semantic features from images and Long Short-Term Memory (LSTM) network for modeling the context sequences was developed.

This integrated approach provides better classification results than other methods mentioned earlier at the same time reduces the complexity of computation. Experimental results demonstrated that the proposed method achieved higher accuracy and F1 scores than some traditional and optimization-based methods.

3.2. Research Limitations

Although there are many positive aspects to this work, there are also a number of limitations that need to be acknowledged. The *first* is that the proposed model is based solely on text and therefore can have limited utility in real world settings when fake news frequently contains multimedia elements like video and image. The *second* limitation is that while GWO has improved feature selection it adds additional computational cost during the optimization process (which can negatively impact scalability) with very large data sets. The *third* limitation is that how well the model performs will depend heavily upon the quality and variety of the training data used to train the model. Further validation of the models ability to generalize to unknown or domain specific data would likely be required. The fourth limitation is that the architectural design does not include attention mechanisms and inclusion of these would improve both the interpretability and contextual understanding of the model.

3.3. Future Works

Although the new proposed model exhibits high quality results in many cases, there are a number of ways that researchers can expand upon it to increase both how well it performs and how applicable it is. One area of expansion includes developing methods of using the most recent advancements or combinations (hybrid) of metaheuristics in order to develop more efficient methodologies of selecting features from datasets and potentially avoiding suboptimal solutions. Additionally, by utilizing some of the newest Deep Learning architectures that include applications of Attention Mechanisms and/or Transformers (i.e., for detecting contextual relationships in Textual Data), the potential exists to greatly improve the model's capabilities in capturing complex contextual associations in textual data. The inclusion of Multimodal data types (including but not limited to: Textual, Visual and Audio/Visual), will enable the model to better address real-world Fake News Scenarios. Finally, another way to build on this model involves investigating Optimize techniques to enable Real-Time Detection of Fake News and to utilize the Model with Large-Scale Deployment, especially considering Social Media platforms which require Rapid Analysis due to their ever-changing nature.

References

1. Tufchi, S., Yadav, A., & Ahmed, T. (2023). A comprehensive survey of multimodal fake news detection techniques: advances, challenges, and opportunities. *International Journal of Multimedia Information Retrieval*, *12*(2), 28.
2. 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.
3. Algara, C., Amlani, S., Collitt, S., Hale, I., & Kazemian, S. (2024). Nail in the coffin or lifeline? Evaluating the electoral impact of COVID-19 on President Trump in the 2020 election. *Political behavior*, *46*(1), 277-305.
4. Priya, B. A., Gayathri, D., Maheshwari, B., Nikhitha, C., Sravanam, D., Yadav, D. B., & Saravanan, M. (2026). Fake news detection using natural language processing. *International Journal of Computer Technology and Electronics Communication*, *9*(2), 541-550.
5. Monday Simon, N. S. S., Abe, G. T., Ashraf Ishaq, L. E., Adda, S. A., & Zaku, V. S. (2026). Machine Learning Approaches for Combating Fake News on social media: Approaches, Advancement, Taxonomy and Future Directions.
6. Papageorgiou, E., Varlamis, I., & Chronis, C. (2025). Harnessing Large Language Models and deep neural networks for fake news detection. *Information*, *16*(4), 297.
7. Kumar, S., Perwej, Y., Siddiqui, F., Shukla, A., & Akhtar, D. N. (2025). A Data-Driven Framework for Fake News Detection Via Web Scraping and Machine Learning Approach. *International Journal of Innovative Science and Research Technology (IJISRT), ISSN-2456-2165*, *10*(6), 1391-1404.
8. Mishra, S., Shukla, P., & Agarwal, R. (2022). Analyzing machine learning enabled fake news detection techniques for diversified datasets. *Wireless Communications and Mobile Computing*, *2022*(1), 1575365.
9. Lawal, M. M., & Abdulrauf, A. (2025). Fake News Detection Using Bi-LSTM Architecture: A Deep Learning Approach on the ISOT Dataset. *Journal of Computing Theories and Applications*, *3*(2), 104-114.
10. Jain, M. K., Gopalani, D., & Meena, Y. K. (2025). UcConvoNet: unifying customised features for deep CNN-based fake news detection. *Multimedia Tools and Applications*, 1-32.
11. Chaudhari, A., & Shrivastava, H. (2025, November). A Metaheuristic Approach to Fake News Detection using Tuned Grey Wolf Optimized SVM. In *2025 3rd DMIHER International Conference on Artificial Intelligence in Healthcare, Education and Industry (IDICAIHEI)* (pp. 1-6). IEEE.
12. Asraa, S. A., Haddad, A. A. A., Hameed, R. S., & Taha, M. S. (2025). An Accurate Model for Text Document Classification Using Machine Learning

Techniques. *Ingenierie des Systemes d'Information*, *30*(4), 913.

13. Abdulghani, S. F., Shtayt, B. A., Taha, M. S., & Hashim, M. M. (2025, September). Effective Knowledge Graph Representation for Cybersecurity Using AI-Based X Data and Named Entity Relation Technique. In *International Conference on Cybersecurity and Artificial Intelligence Strategies* (pp. 71-89). Cham: Springer Nature Switzerland.

14. Ismael, B. M., Ngadi, A. B., Taha, M. S., & Sharif, J. B. M. (2026, February). Non-dominated sorting genetic algorithm for channel assignment in multiple radio interfaces with multiple channels. In *AIP Conference Proceedings* (Vol. 3393, No. 1, p. 060041). AIP Publishing LLC.

15. Sideek, S. M. M., Ali, N. S., Abed, W. Y., & Taha, M. S. (2026). An Improved Anomaly-based Intrusion Detection System for IoT Applications using Machine Learning Methods. *Pertanika Journal of Science & Technology*, *34*(1).

16. Ahmed, A. S., Abood, I. N., & Taha, M. S. (2026). Adaptive Multi-objective Optimization for Obstacle-aware Wireless Sensor Network Deployment: A Comparative Analysis of State-of-the-Art Algorithms. *International Journal of Intelligent Engineering & Systems*, *19*(2).

17. Jabbar, N. K., Naderan, M., & Taha, M. S. (2025). HybridIoMT: A Dual-Phase Machine Learning Framework for Robust Cybersecurity in Internet of Medical Things. *International Journal of Intelligent Engineering & Systems*, *18*(4).

18. Obaid, A. L., Haddad, N. M., & Taha, M. S. (2024, November). DL-SCDDS: Accurate Skin Cancer Detection and Diagnosis Scheme Based on an Improved Convolutional Neural Networks Model. In *International Human-Centered Technology Conference* (pp. 201-214). Cham: Springer Nature Switzerland.

19. Hashim, M. M., Ajeel, N. K., Alhamdane, H. J., Herez, A. H., Taha, M. S., & Ali, A. H. (2020, July). Based on Competitive Marketing: A New Framework mechanism in Social Media. In *IOP Conference Series: Materials Science and Engineering* (Vol. 881, No. 1, p. 012121). IOP Publishing.

20. Ismael, B. M., Ngadi, M. A., Sharif, J. B. M., & Taha, M. S. (2025). Multi-Agent Reinforcement Learning for User-Router Assignment in Multi-Radio Multi-Channel Wireless Mesh Networks. *International Journal of Intelligent Engineering & Systems*, *18*(8).

21. Jain, M. K., Gopalani, D., & Meena, Y. K. (2024). Confake: fake news identification using content based features. *Multimedia Tools and Applications*, *83*(3), 8729-8755.

22. Fathin, M. A., Sibaroni, Y., & Prasetyowati, S. S. (2024). Handling Imbalance Dataset on Hoax Indonesian Political News Classification using IndoBERT and Random Sampling. *J. Media Inform. Budidarma*, *8*(1), 352.

23. Mushtaq, A., Iqbal, M. J., Ramzan, S., Yaqoob, S., Asif, A., & Haq, I. U. (2024). Fake news prediction and analysis in LIAR dataset using advanced machine learning techniques. *Bulletin of Business and Economics (BBE)*, *13*(1).

24. Hanoon, F. S., ˡevik, M., & Taha, M. S. (2024, May). A fault classification for defective solar cells in electroluminescence imagery based on deep learning approach. In *AIP Conference Proceedings* (Vol. 3097, No. 1, p. 050006). AIP Publishing LLC.