Journal of Information Technology Management
ISSN-OnLine: 2980-7972
University of Tehran

Enhancing Fake News Detection by Attention-Based BiLSTM and Hybrid Whale-Multi-Verse Optimization

Varalakshmi K.

Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, A.P.-522 502, India. E-mail: varalakshmiprof@gmail.com

Ashok Kumar P. M. *

*Corresponding author, Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, A.P.-522 502, India. E-mail: profpmashok@gmail.com

Journal of Information Technology Management, 2025, Vol. 17, Special Issue, pp.168-197.
Published by the University of Tehran, College of Management
doi: https://doi.org/10.22059/jitm.2025.102975
Article Type: Research Paper

Received: January 17, 2025
Received in revised form: March 03, 2025
Accepted: June 13, 2025
Published online: August 01, 2025

Abstract

The proliferation of fake news, characterized by the dissemination of inaccurate information to deceive audiences, has become a pressing concern in recent times. Traditional approaches to phony news detection, often focused on analyzing Twitter content, are susceptible to noise and variations in input sequences, leading to suboptimal performance. To address these challenges, this study proposes a novel method called Multi-Head Attention-Hierarchical Bidirectional Long Short-Term Memory (MHA-HBiLSTM) Networks. Our approach involves two phases: training and testing, wherein we employ tweet pre-processing techniques such as stemming, punctuation removal, stop-word elimination, URL handling, and Twitter control removal. Features are represented using the Glove word embedding technique for experimental evaluation and comparison. The MHA-HBiLSTM model integrates multi-head attention and hierarchical concepts, allowing meaningful information extraction from Twitter data. Notably, our model utilizes dual-level attention mechanisms and a hierarchical structure, reflecting the inherent hierarchy in documents and prioritizing key material during document representation. The effectiveness of the proposed MHA-HBiLSTM algorithm is evaluated using the Whale & Multi-Verse (W-MVO) Optimizer approach, with tests conducted on Kaggle and FakeNewsNet datasets. Comparative analysis with traditional machine learning approaches and deep learning models demonstrates the superior performance of the MHA-HBiLSTM approach in fake news detection.

Keywords: Bidirectional LSTM, Deep learning, Fake news detection, Hierarchical, Hybrid Op-timization, Information extraction

Introduction

In the last decade, disseminating false information has become a significant issue across digital media and social networking platforms (Castillo, 2011; Shu et al., 2017). Fake news involves spreading inaccurate information under the guise of legitimate news, with far-reaching impacts on electoral polls, financial markets, and the entertainment industry.

Detecting and identifying false news requires extensive analysis of both true and false content (Conroy et al., 2015; Wang, 2017). Data analysts utilize computational intelligence, natural language processing, and advanced machine learning approaches to analyze hidden patterns in social media data (Ciampaglia et al., 2015; Singh & Kumar, 2023; Vosoughi et al., 2018). Machine learning algorithms offer scalability and efficiency in distinguishing genuine from fraudulent news by extracting pertinent features from text (Ruchansky et al., 2017). However, these algorithms face challenges such as limited training data, overfitting, and a lack of interpretability (Potthast et al., 2017; Shu et al., 2017).

Although deep learning models can automatically learn hidden patterns, they are often constrained by large dataset requirements and their black-box nature (Dhiman et al., 2023; Ruchansky et al., 2017; Samadi & Momtazi, 2023). Fake news detection using Recurrent Neural Networks (RNNs), Long Short-Term Memory (LSTM), and Bidirectional LSTM (BiLSTM) models has been shown to outperform traditional methods by capturing temporal dependencies, despite training and interpretability challenges (Adedoyin & Mariyappan, 2022; Bahad et al., 2019; Ma et al., 2018; Naithani et al., 2023; Shikalgar & Arage, 2023; Tanuku, 2022; Trueman, 2021).

Our research reveals that combining various Twitter data sources using both early and late fusion techniques yields notable recognition accuracy. While early fusion integrates feature vectors, late fusion merges analytical outcomes. The inclusion of multi-head attention mechanisms in BiLSTM enhances the model's capability to detect fake news by focusing on multiple perspectives. Hierarchical modeling in BiLSTM further improves comprehension by capturing complex dependencies, thereby facilitating efficient training and managing key textual features.

This study proposes a novel strategy called Hybrid Whale & Multi-Verse Optimization (W-MVO) integrated with Multi-Head Attention-Hierarchical Bidirectional Long Short-Term Memory (MHA-HBiLSTM) to unify data from diverse sources. Lemmatization and the GloVe word embedding model were essential tools for linguistic analysis in this investigation. Lemmatization improved accuracy by reducing words to their base forms, while averaging semantically similar lemmatized terms captured essential meaning more effectively. During

the training and testing phases, a series of preprocessing methods were applied to ensure comprehensive tweet analysis. These included stemming, punctuation removal, stop-word filtering, URL and Twitter control character elimination, and handling extended tweets.

The present study introduces a comprehensive framework that incorporates W-MVO optimization, late fusion, and an extensive evaluation of the model's ability to detect fake news on Twitter. The core contributions of this research are as follows:

a. Development of the MHA-HBiLSTM architecture consisting of the following stages: tweet preprocessing, tweet representation and encoding, coalition layer, attention mechanism, and output layer.

b. Late fusion of textual data, tweet analytics, user demographics, and user roles to enhance fake news detection accuracy.

c. Implementation of the W-MVO technique to optimize attention weights within the MHA-HBiLSTM network.

d. Performance comparison of the proposed MHA-HBiLSTM model with traditional techniques such as k-NN, Naive Bayes, SVM, CNN, LSTM, and BiLSTM using the Kaggle dataset (Ahmed et al., 2017) and the FakeNewsNet dataset (Shu et al., 2020).

The remainder of this paper is as follows: Section 2 discusses the overview of studies related to Deep Learning models, Late Fusion, and the Hybrid approach. Detailed explanations concerning the methodologies deployed for both the W-MVO and MHA-HBiLSTM frameworks, along with the evaluation metrics utilized for model analysis, are presented in Section 3. Section 4 compares traditional machine learning techniques and the proposed MHA-BiLSTM approach in detecting fake news. Finally, Section 5 wraps up the paper, highlighting key findings, offering recommendations, and suggesting avenues for future research.

Literature Review

Deep learning algorithms are increasingly popular in fake news detection (FND) due to their notable achievements across various domains. Unlike traditional models, deep learning reduces the need for manual feature engineering by leveraging its inherent capacity to extract features directly from input news data. Most fake news detection methods are categorized into unsupervised, weakly supervised, or supervised approaches.

In supervised techniques, various labeled datasets and extracted features are employed to facilitate learning and improve model performance. Enhancements are often achieved through fusion strategies, the incorporation of external resources, and the assimilation of heterogeneous data. Recurrent Neural Networks (RNNs) excel in modeling sequential data

and capturing temporal event information (Ma et al., 2015). The Attention Residual Connection (ARC) network captures both long-range dependencies and localized features (Chen et al., 2019). A Generative Adversarial Network (GAN)-based model highlights low-frequency attributes to generate robust representations of fake news (Ma et al., 2019). However, RNNs may prioritize recent elements, sometimes at the expense of critical earlier information.

Convolutional Neural Networks (CNNs) effectively extract essential features and identify inter-feature relationships (Yu et al., 2019). Multitask learning has demonstrated improvements in attitude classification and fake news detection (Wu et al., 2019). The correlation between news credibility and user-level analysis has also been investigated (Shu et al., 2018). Additionally, attention mechanisms that integrate textual content with user metadata have shown promise in enhancing fake news classification (Dong et al., 2018). Attention-based Bidirectional Gated Recurrent Units (Bi-GRU) have further improved contextual understanding by aggregating content from multiple perspectives.

In weakly supervised learning, deep models benefit from partially labeled data. Guacho et al. (2018) utilized tensor-based representations to produce compact news article embeddings. Hu et al. (2019) combined multi-depth Graph Convolutional Network (GCN) modules with attention to derive layered neighbor information. Li et al. (2021) proposed a self-learning mechanism incorporating confidence estimation for semi-supervised detection. Konkobo et al. (2020) emphasized early detection by analyzing user opinions and evaluating source credibility.

Unsupervised anomaly detection methods project standard posts into deep embedding spaces, flagging outliers as possible rumors (Yuan et al., 2020; Zhang & Lee, 2017). Autoencoders have been used to evaluate news authenticity by examining reconstruction errors (Chen, T. et al., 2018). Generative adversarial models have also demonstrated flexibility in unsupervised settings (Chen, X. et al., 2019; Qiu et al., 2019).

Long Short-Term Memory (LSTM) networks, a well-known deep learning variant, address the limitations of capturing long-term dependencies by introducing "short-term memory" gates. Attention mechanisms in deep networks act as bridges between encoder and decoder components, enabling the model to focus selectively on informative segments. Attention-based LSTM models that incorporate Twitter profile metadata have the potential to significantly enhance fake news detection (Long et al., 2017).

Multi-head attention is often favored over single-head attention in Bidirectional LSTM (Bi-LSTM) models due to its ability to capture diverse sequence characteristics, model inter-token dependencies, increase expressiveness, and reduce overfitting risk, thereby improving stability and prediction reliability. Optimization of attention weights using the Whale Optimization Algorithm (WOA) has been shown to enhance model performance and

interpretability. Nevertheless, classical WOA faces limitations such as poor convergence in complex tasks, early stagnation, and constrained parameter flexibility.

This study proposes a novel hybrid approach that integrates the Multiverse Optimization (MVO) technique with WOA to improve fake news classification. Additionally, late fusion is applied to combine multiple information sources, leading to performance gains. The proposed WOA-MVO hybrid strategy provides advantages such as balanced exploration and exploitation, accelerated convergence, enhanced diversity preservation, and increased optimization effectiveness. By merging late fusion with this hybrid optimization strategy, the model successfully addresses complex classification problems and yields superior results.

Methodology

Proposed Multi-Head Attention: Bidirectional Long Short-Term Memory

Overview

This section introduces a novel Multi-Head Attention-Hierarchical Bidirectional Long Short-Term Memory (MHA-HBiLSTM) model for identifying syntactic and semantic relationships in detecting fake news on Twitter. The attention weights in the proposed MHA-HBiLSTM architecture are optimized using the Hybrid Whale and Multiverse Optimization (W-MVO) algorithm. The detection framework operates in two phases: the learning phase and the testing phase.

During the learning phase, tweets undergo preprocessing steps, and GloVe embeddings are utilized to convert them into numerical representations. The MHA-HBiLSTM model is subsequently trained to optimize attention weights between words within each sentence, guided by the W-MVO technique. This stage relies on labeled data for supervised learning. In the testing phase, incoming Twitter sentences are similarly preprocessed and transformed into numerical vectors using the same embedding approach. The pre-trained MHA-HBiLSTM model then analyzes these vectors to detect potentially false or misleading content.

Figure 1 illustrates the overall architecture of the MHA-HBiLSTM model, showcasing the stages of Twitter data preprocessing and fake news prediction.

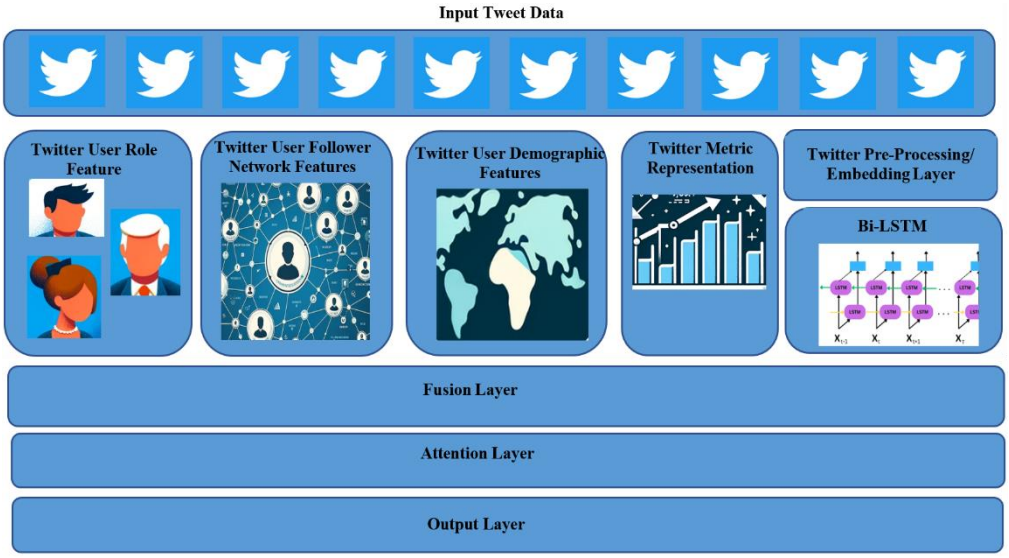


Figure. 1. Proposed Multi-Head Attention-Hierarchical BiLSTM (MHA-HBiLSTM)

Preliminary Tweet Processing:

The initial phase of tweet preprocessing involves essential steps aimed at noise reduction and structuring data for analytical purposes. In the proposed framework for fake news detection on Twitter, multiple text preprocessing and feature extraction strategies were implemented to improve the accuracy and reliability of the analysis.

Text preprocessing began with tokenization, which breaks down tweets into meaningful tokens. Stop words, commonly used terms that carry limited semantic value, were removed to reduce noise and enhance linguistic clarity. URLs were excluded as they rarely contribute to semantic understanding. Similarly, non-alphanumeric characters, including special symbols and emojis, were eliminated to simplify the textual input and remove irrelevant elements.

To capture deeper contextual relationships between words, word embeddings were applied using the pre-trained GloVe model implemented via the Gensim package. These embeddings allowed the model to understand the semantic proximity and syntactic roles of words in various contexts, thus aiding in more accurate fake news classification.

In addition to textual data, tweet-level metrics such as impressions and engagement rates (see Equation 1) were computed to measure the visibility and interaction associated with each tweet. These metrics helped identify behavioral patterns commonly linked to fake news dissemination. Furthermore, user demographic features, including age, gender, geographic location, and stated interests, were integrated into the model, as these characteristics often influence the likelihood of spreading or believing fake news. Finally, an analysis of follower interaction dynamics was performed to evaluate user credibility. Attributes such as follower count, follower-to-following ratio, and interaction behavior provided valuable insight into trustworthiness and potential for misinformation spread.

$$\text{Engagement rate} = \frac{\text{Total number of engagement actions}}{\text{Total number of impressions}} * 100 \quad (1)$$

The model also incorporated Twitter's verification status as an important indicator of user authenticity. The presence of the blue verification badge served as a proxy for account legitimacy, as verified profiles are typically associated with reputable individuals or organizations. Additionally, user roles and affiliations—such as corporations, media outlets, influencers, celebrities, and political figures—were considered critical in assessing the credibility of shared content. High-profile accounts are generally less prone to spreading misinformation compared to lesser-known or unverified profiles. Together, these features contribute to a robust framework for detecting fake news on Twitter by integrating both content-based indicators and behavioral user attributes.

Hierarchical Bidirectional Long Short-Term Memory (HBLSTM)

The Long Short-Term Memory (LSTM) network (as shown in Figure 2) is a specialized form of Recurrent Neural Network (RNN) designed to capture sequential dependencies while mitigating the vanishing gradient issue. The LSTM cell operates as shown in Equations 2 through 6, where the input gate (Equation 2), forget gate (Equation 3), cell state update (Equation 4), output gate (Equation 5), and final hidden state computation (Equation 6) are defined mathematically:

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + b_i) \quad (2)$$

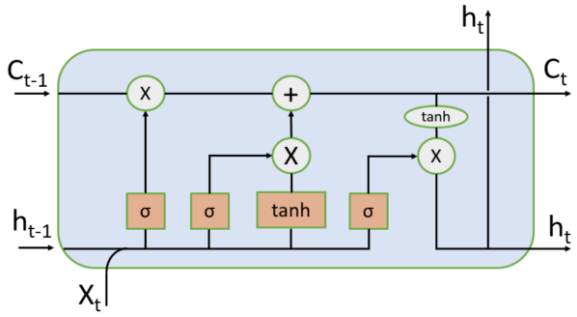
$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + b_f) \quad (3)$$

$$c_t = f_t \cdot c_{t-1} + i_t \cdot \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \quad (4)$$

$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + b_o) \quad (5)$$

$$h_t = o_t \cdot \tanh(c_t) \quad (6)$$

Where x_t is the input vector at time t, h_t is the current hidden state, h_{t-1} is the previous hidden state, c_{t-1} and c_t are the memory cell states, σ is the sigmoid activation function, $tanh$ is the hyperbolic tangent function, and $tanh$ are weight matrices and bias vectors. LSTMs outperform standard RNNs by selectively retaining or discarding past information, enabling them to capture long-range dependencies effectively.



X_t: Input Vector
h_t: Present network's output
h_{t-1}: Past LSTM unit's output
c_{t-1}: Previous Memory unit

c_t: Current Memory unit
×: Element-wise multiplication
+: Element-wise Addition
tanh: Hyperbolic tangent

Figure. 2-Long Short-Term Memory (LSTM) neural network

Extending LSTM, the Bidirectional Long Short-Term Memory (BiLSTM) model enhances sequential learning by processing information in both forward and backward directions, providing richer contextual information. The forward and backward states are calculated using Equations 7 and 8, respectively, while Equation 9 shows their concatenation to form the final hidden representation:

$$fh_t = LSTM_{forward}(\mathrm{x_t}, fh_{t-1}) \tag{7}$$

Simultaneously, the backward LSTM captures dependencies from future steps:

$$bh_t = LSTM_{backward}(\mathrm{x_t}, bh_{t+1}) \tag{8}$$

The final hidden representation in BiLSTM is obtained by concatenating these states:

$$h_t = [fh_t, bh_t\,] \tag{9}$$

Building upon BiLSTM, the Hierarchical Bidirectional Long Short-Term Memory (HBLSTM) model (as shown in Figure 3) introduces hierarchical layers that extract information at multiple levels of granularity. At each layer, forward and backward LSTM units process input sequences, capturing bidirectional dependencies. The Hierarchical BiLSTM (HBiLSTM) further improves this architecture by introducing multiple BiLSTM layers. Each hierarchical layer refines the feature representation from the previous layer as defined in Equation 10:

$$H^{(l)} = BiLSTM\left(H^{(l-1)}\right) \tag{10}$$

where $H^{(l)}$ is the hidden representation at layer l, and $H^{(l-1)}$ is the previous layer's output.

By combining bidirectional processing with hierarchical structures, $HBLSTM$ significantly enhances the model's ability to capture both local and global dependencies. This makes it highly suitable for fake news detection, where understanding complex textual patterns and context is crucial.

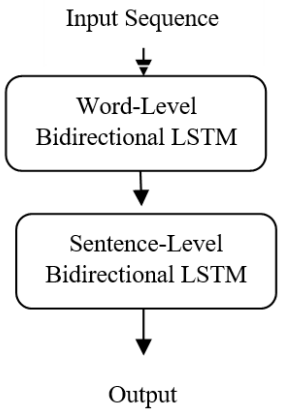


Figure 3. Structure of Hierarchical Attention

This hierarchical technique allows the framework to extract local and global context data, which is helpful for jobs with hierarchical structures. HBLSTM improves the model's capacity to record long-term relationships and performance in tasks such as text categorization and sentiment evaluation by integrating bidirectional processing and hierarchical structures.

Fusion Layer

The most recent integration approach in misinformation classification involves combining features from multiple sources into a unified attribute vector. These sources include tweet content and Twitter metadata, such as follower count and retweet frequency. Various techniques are used for extracting textual features, including bag-of-words, term frequency-inverse document frequency (TF-IDF), and word embeddings. In this study, the GloVe embedding technique is employed to represent the attribute vector with rich contextual semantics.

The final unified feature vector is obtained by concatenating representations derived from the Bi-LSTM outputs, textual features, and social media attributes. This combined vector is then passed through a fully connected layer followed by an attention mechanism, which helps the model focus on the most informative features for tweet classification. The integration of both linguistic and social network information enhances the model's ability to detect fake news.

However, the resulting fused feature vector may be high-dimensional, requiring careful design of the model architecture and optimization of the training process to mitigate computational challenges. The late fusion strategy adopted in this study involves feature-level concatenation across multiple modalities—namely, textual data and social media features. This strategy provides a comprehensive representation that preserves the semantic relationships captured by the GloVe model. The concatenated vector is input into the subsequent layers of the model for further processing and classification. By integrating

heterogeneous sources of information, the model achieves a more holistic understanding of the data, thereby improving its accuracy in identifying fake news.

However, the enormous dimensionality of the concatenated feature vector might provide computing issues. Thus, thorough optimization of the model's architecture and training methods is required for efficient and successful categorization. Overall, the late fusion strategy with feature concatenation effectively improves the performance of false news classification models. (as seen in Figure. 4)

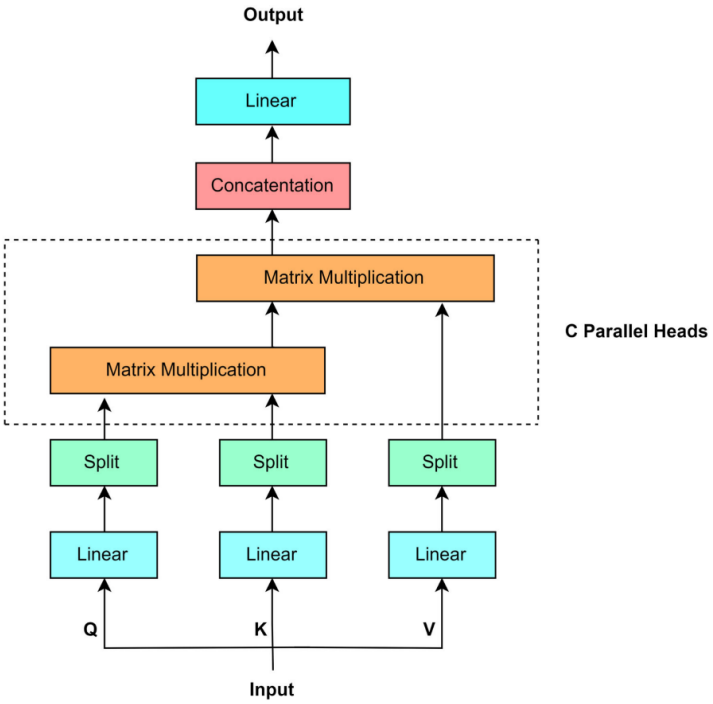


Figure 4. Multi-Head Attention Layer

Attention Layer

Deep neural networks utilize attention layers to selectively emphasize critical components of the input data sequence, preserving both explicit and implicit contextual information (see Figure 4). One widely adopted approach is the multi-head attention mechanism proposed by Vaswani et al. (2017), which employs multiple attention heads to capture diverse contextual representations. This design allows the network to simultaneously attend to information from various representation subspaces and positional encodings. In addition, the concept of self-attention, introduced by Lin et al. (2017), enables the model to capture internal dependencies within a given semantic space. By analyzing the relationships between individual words in a sequence, self-attention facilitates the extraction of contextually rich sentence embeddings, thereby enhancing the model's comprehension and interpretability.

The attention function receives input from a series of queries, represented as $\{Q_1, \ldots, Q_N\}$, & a collection of keys and values, indicated as $\{K, V\} = \{(K_1, V_1), .., (K_R, V_R)\}$,to facilitate

the implementation of the multi-head focus method. Using learnable linear projections, the multi-head focus framework converts the queries(Q), keys (K), and values (V) through C sub-domains. Here, $Q^c, K^C, and\ V^C$, stand for the query's c^{th} head, key, and value, respectively. The corresponding subspace and dimensionality of the model are determined by the learnable weights represented by the parameter matrices $\{W_c^Q, W_c^K, W_c^V\} \in R^{d\times d_k}$. In addition, the output states $O^1,\ldots,O^C$ are obtained by applying C attention functions concurrently, where the output for each head is computed using Equation 11, and attention weights are calculated using Equation 12.

$$\mathrm{O^c} = \mathrm{A^c\ V^c} \tag{11}$$

$$\mathrm{A^c} = \mathrm{softmax}\left(\frac{\mathrm{Q^c K^{c^T}}}{\sqrt{\mathrm{d_k}}}\right) \tag{12}$$

In this context, the focal point allocation was created using the c^{th} interest head is represented by A^c. The final condition is created by concatenating the resultant phases. The computational diagram of the multi-head focus system, illustrating the process, is depicted in Figure 3. The suggested system extracts features by concatenating concealed states fh_k, bh_k from the BiLSTM neural network at every step, as shown in Equation 13. In the subsequent step, the concatenated vector S_{t-1} is inputted into the BiLSTM to obtain the resultant concealed-state column array S_t (Equation 14). The scalar significance threshold $\mathrm{S_t}$is then generated utilizing the traditional Whale Optimization Algorithm (WOA), which relies on a performance measure. Lastly, the focus-based approach computes a weighted average α_t of the present state S_t, as represented in Equation 15.

$$\mathrm{S_t} = \mathrm{Concat}(\mathrm{fh_1}, \mathrm{bh_1}, \mathrm{fh_2}, \mathrm{bh_2}, \ldots\ldots \mathrm{fh_n}, \mathrm{bh_n}) \tag{13}$$

$$\mathrm{S_t} = \mathrm{HBiLSTM}(\mathrm{S_{t-1}}) \tag{14}$$

$$\mathrm{X^l} \leftarrow \mathrm{feature_{fusion}}\left(\mathrm{S_c^l}, \mathrm{X_m^l}, \mathrm{X_d^l}, \mathrm{X_f^l}, \mathrm{X_r^l}, \mathrm{X_v^l}\right) \tag{15}$$

A particular section of a specific comment is essential for spotting fake news. However, there are numerous ways to focus attention on a word, making the use of multiple attentional mechanisms necessary. To communicate the entire meaning of the statement, each word is assigned a suitable weight based on several variables.

$$Y = tanh(W_{k1}\ X^T) \tag{16}$$

$$Z = softmax(W_{k2}\ Y) \tag{17}$$

The whole set of hidden states H is fed into the attention layer. It applies a multiplication operation with $W_{k1} \in \mathbb{R}^{g\times 2p}$, , as described in Equation 16. The resulting output is then fed into a $tanh$ function to produce Y. To extract the attention for each component based on various factors, Y is multiplied by $W_{k1} \in \mathbb{R}^{g\times 2p}$ (Equation 17) and passed through a softmax function. This calculates the scaled significance weights across various attention heads (q),

yielding in a weight vector denoted as Z. Weighted sum over q dimensions is achieved by multiplying the word H hidden states by the weight vector Z. The representation of the sentence embedding is denoted as a matrix M, as shown in Equation 18.

$$M = Z X \quad (18)$$

Hybrid Whale & Multi-Verse (W-MVO) Optimizer

The Traditional Whale Optimization Algorithm (WOA) is an intelligent optimization approach that aims to improve prey location and search space exploration. WOA improves its performance through adaptive control and re-initialization methods. The exploration and exploitation stages must be synchronized to increase search efficiency. On the other hand, WOA faces difficulties, including poor convergence speed and local optima traps. The initial population allocation significantly affects convergence, stability, and search efficiency. Limited prey numbers may lead to early searching and inadequate representation of the solution space.

Table 1. **Proposed W-MVO algorithm**

Input:

1. - Population size (N)
2. - Search space boundaries (LowerBound, UpperBound)
3. - Maximum number of iterations max_iter.

Output: Optimized feature set X_i^*

1. Initialize the population of whales $\{X_i\}$ $(i = 1, 2, \ldots, n)$, where n is the population size.
2. Initialize the iteration counter $t = 0$.
3. Initialize the maximum number of iterations max_iter.
4. while $t < max\_iter$:
5. Utilizing the equation, determine the value of a.:
6. $a = 2 - t * (2 / max\_iter)$
7. for each whale X_i:
8. Calculate the fitness value of X_i: $f(X_i)$
9. if $f(X_i) < f_{best}$:
    i. Update the best solution $f_{best} = f(X_i)$
10. Select a random universe X_j from the population where $j \neq i$.
11. if $f(X_j) < f_i$:
    i. Calculate the distance d between X_i and X_j: d = $\|X_i - X_j\|$
    ii. Update X_i's position using the encircling equation:
    iii. $X_i(t+1) = X_j + rand(0,1) * (X_i(t) - X_j) + a * d * (rand(-1,1))$
12. else:
    i. Randomly generate a new position X_{new} within the search space.
    ii. Calculate the distance d between $X_i$and X_{new} : $d = ||X_i - X_{new}||$
    iii. Update X_i'sposition using the encircling equation: $X_i(t+1) = X_{new} + rand(0,1) * (X_i(t) - X_{new}) + a * d * (rand(-1,1))$
13. Increment the iteration counter: $t = t + 1$
14. Return the best solution found: X_i^*

To address the limitations of conventional optimization strategies, the proposed approach integrates the Multi-Verse Optimization (MVO) method with the Whale Optimization

Algorithm (WOA). MVO demonstrates high performance in both unimodal and multimodal test functions by emphasizing exploitation while maintaining robust exploration capabilities, thereby avoiding entrapment in local optima. It addresses local search stagnation by enhancing diversity during the exploration phase and striking a balance between exploration and exploitation. The mean fitness function of MVO continuously guides convergence by improving performance across all "universes" as iterations progress. Building upon these strengths, a hybrid metaheuristic algorithm named MV-WOA is introduced to augment the effectiveness of WOA by incorporating MVO dynamics.

The WOA-MV algorithm is a population-based metaheuristic in which candidate solutions are metaphorically represented by whales. The process iteratively enhances the population by alternating between exploration and exploitation. The balance is regulated by a control parameter “a”, which decreases linearly over time, gradually shifting focus from exploration to exploitation. At each iteration, the fitness value of every whale is calculated using an objective function. If a whale's fitness exceeds that of the current best solution, it replaces the best-known solution. Additionally, the algorithm compares a randomly selected whale's fitness to that of the current whale; if the random whale has better fitness, the encircling mechanism is used to update the whale's position.

In cases where no improvement is found, the algorithm promotes random exploration by generating new positions within the search space, updating whale locations accordingly, and calculating the distance between the current and new positions. This randomness allows the algorithm to navigate unexplored regions of the solution space. The iteration process continues until the predefined maximum number of cycles is reached. Ultimately, the algorithm outputs the optimal solution discovered. WOA-MV effectively fuses the exploratory strength of MVO with the adaptive exploitation strategy of WOA, offering a balanced and efficient optimization mechanism. This synergy enables the algorithm to thoroughly explore the search space while maintaining a strong convergence tendency toward the global optimum (see Table 1).

Prediction Layer

Predicted Outcome $\hat{y}$ implies the chance of labeling the information as deceptive. This prediction is generated by feeding the Ultimate cell output into an LSTM network via a rudimentary neural system with one hidden layer. The last layer of this network uses a simple SoftMax classifier to train its result. Operating within a fully connected layer, the classifier can interpret the input text and make predictions, accordingly, as represented in Equation (19):

$$\hat{y} = \text{softmax}(W_f M + b_f) \quad (19)$$

where W_f and b_f represent the last linear layer's weight distribution and bias, respectively. This model is trained using a binary cross-entropy loss, which is computed as shown in Equation (20):

$$\text{loss} = -\sum_{i=1}^{e} y_i \log(\hat{y_i}) \quad (20)$$

Here, e stands for the quantity of class output labels. y_i denotes the output class labels of the i^{th} class & $\hat{y_i}$ denotes the expected probability for the i^{th} class.

Table 2. **Proposed Multi-Head Attention- Hierarchical BiLSTM (MHA-HBiLSTM) Algorithm**

Input: Kaggle (Ahmed, H. et al.,2017), FakeNewsNet (Shu, K et al., 2020) Dataset.
Output: Fake News/Non-Fake News.

1. Datasets ← get-news-text content from Kaggle/FakeNewsNet Dataset
2. y^l ← Get_Output_Labels
3. X_m^l← get-Tweet-Metric-Representation info from APIs
4. X_d^l ← get-Tweet-User-Demographic Features info from APIs
5. X_f^l←get-Tweet-User-Follower-Network features social networks APIs
6. X_r^l←get-Tweet-User-Role-features social networks APIs
7. X_v^l←get-Twitter-Verification-Feature features social networks APIs
8. X_c^l ← Tweet Pre_Processing(Dataset)
9. X_c^l ← Embedding_layer(X_c^l)
10. $fh_i^l, bh_i^l \leftarrow BiLSTM(X_c^l)$, i=1,2,3…t. & l=1,2,3,…N
11. $S_c^l = Concat(fh_1^l, bh_1^l, fh_2^l, bh_2^l, \dots \dots fh_n^l, bh_n^l)$
12. X^l ← feature_fusion ($S_c^l, X_m^l, X_d^l, X_f^l, X_r^l, X_v^l$)
13. $X_{train}, X_{test}, y_{train}, y_{test} \leftarrow Train\_Test\_Split(X, y)$
14. **For** l=1 to N **do**
15. $W_{k1} = W - MVO(S_t, Binary\_Cross\_Entropy, X_{train}^l, y_{train}^l)$
16. $Y = tanh(W_{k1}\ X^T)$
17. $Z = softmax(W_{k2}\ Y)$
18. $M = Z\ X$
19. $\hat{y} = softmax(W_f\ M + b_f)$
20. **End for**
21. δ ←user_threshold
22. **For** l=1 to N **do**
23. $\hat{y} = softmax(W_f\ M + b_f)$
24. **If** $\hat{y}$ > δ then X_{test} represents fake news
25. **Else then** X_{test} represents non-fake news
26. **End for**

Fake News Detection Method

The MHA-HBiLSTM technique for detecting fake news combines various extracted features to analyze their relationships and hidden information. This is accomplished by combining the Hybrid Whale & Multi Optimization Algorithm with the attention-based Hierarchical BiLSTM technique. The algorithm can ascertain if a given news story is authentic or fraudulent by determining the ideal weights. The output layer categorizes information as bogus if the resulting probability value exceeds a predetermined threshold; otherwise, it is deemed true news. The algorithm starts by representing tweet features and verification features, including tweet metric, user_demographic, user_follower, and User_role features. These features are processed and encoded accordingly. The Bi-HLSTM model is subsequently employed to obtain hidden state information from the Twitter user. Initial features and hidden state features are merged.

The data is separated into two sets: training and evaluation, with training contributing around 80% and testing for 20%. The training process, depicted in lines 14-20, focuses on finding the attention weights using the Whale optimum algorithm. Similarly, the testing process, outlined in lines 21-26, involves using a user_threshold (δ) to perform classification. This is done by employing the optimized attention weights at the attention and output layers. The MHA-HBiLSTM algorithm (Table 2) combines various features, utilizes attention mechanisms, and applies optimization techniques to identify fraud information based on the obtained probability value and a specified threshold.

Results and Discussion

For conducting lab tests, the experimental setup in this study used Python 3, Google Colab, and GPU resources. The proposed framework was implemented using Scikit-Learn and NLTK, among other Python libraries. The tools, dataset, and standard algorithm for the experiments were defined in the first step. The architecture's accuracy was then assessed. Pre-trained word vectors from the Glove set with 200 dimensions were used to start the embedding layer. A comparative analysis was also conducted to compare the classification outcomes of WOA-attention Bi-LSTM-based approaches with those of machine learning methods.

Experimental Datasets

The proposed models were trained, validated, and tested using two benchmark datasets. To ensure comprehensive evaluation, each dataset was split into 80% for training and 20% for testing, following a two-phase analysis protocol. The subsequent section provides an overview of these datasets and highlights their key characteristics. One of the datasets used for fake news detection was sourced from Kaggle, as introduced by Ahmed et al. (2017). This dataset comprises 3,988 news articles, each containing a title, a collection of URLs, and a main body paragraph. A binary label is assigned to each entry, where “0” represents fake news and “1” represents real news.

For this study, the analysis focused on the main text content, excluding the headline and the article’s title. Based on the original distribution within the dataset, 1,868 articles are labeled as real news, while 2,120 articles are classified as fake news.

The FakeNewsNet dataset, compiled by Shu et al. (2020), was also utilized in this study. It comprises data from two primary sources: PolitiFact and GossipCop. Each source includes separate files for real and fake news instances. For the PolitiFact dataset, two files were used: "politifact_real.csv" and "politifact_fake.csv". The real news file contains 432 Twitter posts corresponding to verified news articles, while the fake news file comprises 618 Twitter posts linked to fabricated news items. Similarly, the GossipCop data source includes "gossipcop_real.csv" and "gossipcop_fake.csv". The real news file contains 5,328 Twitter

posts associated with authentic news content, whereas the fake news file consists of 5,322 Twitter posts tied to deceptive information.

Each entry across these datasets includes multiple fields, such as a unique identifier, website URL, news headline, and associated social media metadata, facilitating a multifaceted analysis of both textual and user-generated components in fake news detection. All four datasets were combined to create data, resulting in a significant dataset. A column called "label" was added, designating the classification of each document as either true news or false news using the numbers 0 or 1. The dataset currently contains 44,280 tweets in total.

Parameter Setting

This study conducted a detailed analysis to determine the most effective optimization algorithm based on repetition and accuracy. Among the various strategies examined, the Whale Optimization Algorithm (WOA) demonstrated superior performance, achieving an accuracy rate of 96%. This optimizer effectively minimizes objective functions through gradient backpropagation and parameter adjustments. The performance of other widely used optimization algorithms was also evaluated. RMSprop achieved 95.75%, while Adam yielded 94.5%, and Adagrad recorded 93.38%. These comparative results informed the selection of the most suitable optimization approach for the model.

We implemented the dropout strategy in our system to reduce the chance of overfitting. Rather than applying dropout to all nodes in the network, we maintained the usual protocol and limited the effect on the connections across levels. The dropout rate, a learning-optimized hyperparameter, determines the likelihood of node elimination during recurrent training. This dropout normalization strategy dramatically increased the framework's effectiveness by effectively reducing overfitting.

Notably, the accuracy and cost changes between instruction and evaluation were minimal, indicating that the dropout technique effectively promoted correct categorization while preventing overfitting. Table 3 details the factors employed in the research procedure.

Table 3. Configuration Settings

Criterion	Quantities
Input dimensionality	250
Maximum Feature Capacity	100
Batch_Size	32
Iteration Count	40
Training pace	0.001
normalization pace	0.025
Attrition likelihood	0.2
Transfer function	ReLU
Performance Tuning	Adam
Prediction layer Function	Softmax

Evaluation with Other Approaches

K-Nearest Neighbors (K-NN) algorithm, as described by McCallum and Nigam (1998), classifies new textual data based on its similarity to labeled training instances. A given text is categorized by evaluating its proximity to previously labeled examples, where closeness is typically measured using distance metrics such as cosine similarity or Euclidean distance. K-NN is particularly useful in text classification tasks involving multiple content types, including emails, documents, and social media posts. Each text is represented as a feature vector composed of frequently occurring terms or other relevant attributes, enabling effective comparison and classification based on learned patterns.

Naïve Bayes Classifier is a widely used machine learning algorithm for classification tasks, as outlined by Zhang and Lee (2003). It applies Bayes' Theorem to estimate the likelihood that new data belongs to a specific class. Assuming conditional independence among features, the classifier calculates the posterior probability by multiplying the prior probability of a class with the likelihood of its features. The class with the highest posterior probability is selected as the predicted label. This approach is particularly effective in applications such as sentiment analysis, spam filtering, and text classification. In the present study, a parameter value of 0.05 was used; however, this value may vary depending on the dataset and specific problem domain.

Support Vector Machines (SVMs) (Kuang & Xu et al., 2010) are among the often-utilized supervised learning approaches for text categorization. They want to optimize the gap between classes in the data by defining a hyperplane that separates them. The information points nearest the hyperplane, or support vectors, are crucial in identifying the decision boundary. By optimizing the margins, SVMs can handle new data effectively and increase generalization. When a linear hyperplane is insufficient, SVMs use Kernel Mapping to move the input vector into a higher-dimensional space. They are successful at segregating the data and identifying nonlinear decision boundaries. SVMs can manage complicated data connections and capture nonlinear patterns because the kernel function is versatile, including linear, polynomial, and radial basis function (RBF) implementations. Because of their adaptability, SVMs are helpful for text categorization and other classification tasks.

Convolutional Neural Networks (CNNs) (Kim 2014) are powerful deep models for text classification. They are competent in identifying patterns and extracting crucial information from various text types, including emails, papers, and social media posts. By training a CNN on a wide range of text data, we get essential insights into the structures, trends, and patterns present in textual content. CNNs process large volumes of text data efficiently, exposing hidden information and allowing for detailed analysis. Their use of text categorization allows us to gain essential insights and draw well-informed conclusions from a comprehensive textual study.

Evaluation Settings

We used standard categorization criteria to assess the suggested technique's performance. In this study, the acronyms TP, TN, FP, and FN denote true positive, true negative, false positive, and false negative, respectively. These indicators are benchmarks for determining the strategy's efficacy and precision in distinguishing between positive and unfavorable scenarios.

We calculated accuracy by calculating the ratio of correct predictions to total predictions made. In essence, accuracy evaluates a model's ability to produce exact forecasts, as shown in Equation (21):

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}} \tag{21}$$

We also evaluate the proposed model's ability to generate True positives (TP) among all positive predictions, including real positives and erroneous positive results. It is calculated as the ratio of TP to the total of TP and FP, as shown in Equation (22):

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{22}$$

Recall is calculated by dividing total TP by the sum of TP and FN. A high recall rate indicates the system's ability to recognize affirmative cases. However, considering the possibility of increasing misclassification of negative cases as positives should be evaluated. It is calculated as in Equation (23):

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{23}$$

The F1 score extensively assesses categorization outcomes, considering both recall and precision. It is calculated as the harmonic average of recall and accuracy, with each measurement weighted equally. A higher F1 score (0–1) indicates more excellent model performance, as shown in Equation (24):

$$\text{F1} - \text{score} = \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \tag{24}$$

Experimental Outcomes

In this study, we developed a neural network model named MHA-HBiLSTM (Multi-Head Attention with Bidirectional Long Short-Term Memory), specifically designed for the detection of fraudulent news. The model evaluates the significance of each word within a tweet by considering its contextual relevance. Words that are less pertinent to the classification objective receive lower attention weights, while more significant terms are given higher weights, enhancing the model's interpretability and focus.

To evaluate the effectiveness of the proposed system, comparative experiments were conducted using various established classification algorithms, including those described by McCallum and Nigam (1998), Zhang and Lee (2003), Kuang and Xu (2010), and Kim (2014). These comparisons were carried out using the Kaggle fake news dataset (Ahmed et al., 2017). The results of the experiments are illustrated in Figures 5 and 7(a). Additionally, Figures 6 and 7(b) present a comparative analysis of the model's performance against the benchmark classifiers.

An in-depth performance evaluation of various machine learning algorithms, including the proposed Multi-Head Attention Bidirectional Long Short-Term Memory (MHA-BiLSTM) model, was conducted using the Kaggle dataset (Ahmed et al., 2017). The results are visually presented in Figures 5 and 7(a). A comparative analysis of each model's classification performance is further outlined through the evaluation metrics depicted in Figure 6. The performance of the K-Nearest Neighbors (KNN) algorithm was tested with different values for the "neighbors" parameter. For k=5, the model achieved an accuracy (Acc) of 75%, precision (Pr) of 78%, recall (Rc) of 80%, and F1-score of 79%. When the number of neighbors increased to 10, accuracy improved to 78%; however, the other metrics—precision, recall, and F1-score—showed a decline. A further increase to 15 neighbors led to a marginal drop in all performance metrics. The Naive Bayes classifier yielded an F1-score of 0.77, recall of 0.79, accuracy of 0.80, and precision of 0.75. The Support Vector Machine (SVM) model was evaluated using multiple kernel functions. With a linear kernel, the model achieved an accuracy of 68%, precision of 55%, recall of 67%, and an F1-score of 60.5%. When using a polynomial kernel (degree = 3), the model's performance improved: recall reached 79%, accuracy was 77%, precision was 67%, and F1-score rose to 72.6%. In contrast, the sigmoid kernel underperformed, achieving 75% accuracy, 43% precision, 65% recall, and an F1-score of 51.8%.

The Convolutional Neural Network (CNN) achieves an Acc of 0.79, Pr of 0.74, Rc of 0.8, and an F1-Score of 0.769. Long Short-Term Memory (LSTM) system performs more effectively, with an Acc of 84%, Pr of 89%, Rc of 86%, and F1-Score of 87.5%. Outcomes are further improved by the Attention-Based LSTM (ABiLSTM) model, which has an F1-Score of 91.5%, Rc of 90%, Acc of 91%, and Pr of 93%. The WOA-ABiLSTM model then attains an F1-Score of 92.5%, Rc of 94%, Acc of 93%, and Pr of 90%. With an F1-Score of 93.7%, Acc of 94%, Pr of 91%, and Rc of 96%, the suggested MHA-BiLSTM model performs even better. Ultimately, the model with the most outstanding performance is the MHA-HABiLSTM, which has an F1-Score of 95.1%, accuracy of 95.4%, precision of 93%, and recall of 95%. Overall, Figure. 5 compares the various machine learning models, with the MHA-HABiLSTM model outperforming all other evaluated models.

Figures 6 and 7(b) compare many predictive algorithms with the recommended MHA-BiLSTM model for the FakeNewsNet (Shu et al., 2020) dataset. The Model's effectiveness is evaluated beginning with K Nearest Neighbors (KNN) and progressing through various

"neighbours" parameter values. With 5 neighbors, the KNN model achieves 72% Acc, 75% Pr, 76% Rc, and 75.5% F1-score. The accuracy rises to 77% when the number of neighbors is increased to 10, but Pr, Rc, and F1-Score all slightly decline. Furthermore, Acc, Pr, Rc, and F1-Score dramatically decrease when the number of neighbors is increased to 15.

Figure 5. Heat Map-Performance Comparison of Proposed MHA-BiLSTM Model on Kaggle (Ahmed et al., 2017) Dataset

Moving on to Naive Bayes, the model achieves an Acc of 67%, Pr of 56%, Rc of 62%, and an F1-Score of 58.9%. The SVM model is evaluated using different kernel functions. The linear kernel's Acc is 52%, Pr is 40%, Rc is 55%, and the F1-Score is 46.4%. When using the polynomial kernel with a degree of 3, the accuracy improves significantly to 80%, Pr to 71%, Rc to 79%, and the F1-Score to 74.8%. However, the sigmoid kernel shows moderate performance with an Acc of 63%, Pr of 58%, Rc of 61%, and an F1-Score of 59.5%. The Convolutional Neural Network (CNN) achieves an Acc of 81%, Pr of 0.78, Rc of 0.83, and an F1-Score of 0.805. The Long Short-Term Memory (LSTM) model demonstrates better performance with an accuracy of 86%, precision of 82%, recall of 84%, and an F1-Score of 83%. The Attention-Based LSTM (ABiLSTM) model further improves the results with an Acc of 88%, precision of 86%, recall of 90%, and an F1-Score of 88%.

Next, the WOA-ABiLSTM model has an Acc of 92%, Pr of 89%, Rc of 94%, and an F1-Score of 91.5%. The recommended MHA-BiLSTM model produces even better results, with an F1-Score of 93.4%, Acc of 93.5%, Pr of 92%, and Rc of 96%. Finally, the model with the best performance is the MHA-HABiLSTM, with an F1-Score of 95.1%, Acc of 94.6%, Pr of 93%, and Rc of 95%. This suggests that, compared to other models, the MHA-HABiLSTM model performs better and is wildly successful in classifying the FakeNewsNet (Shu et al., 2020) dataset by identifying significant trends.

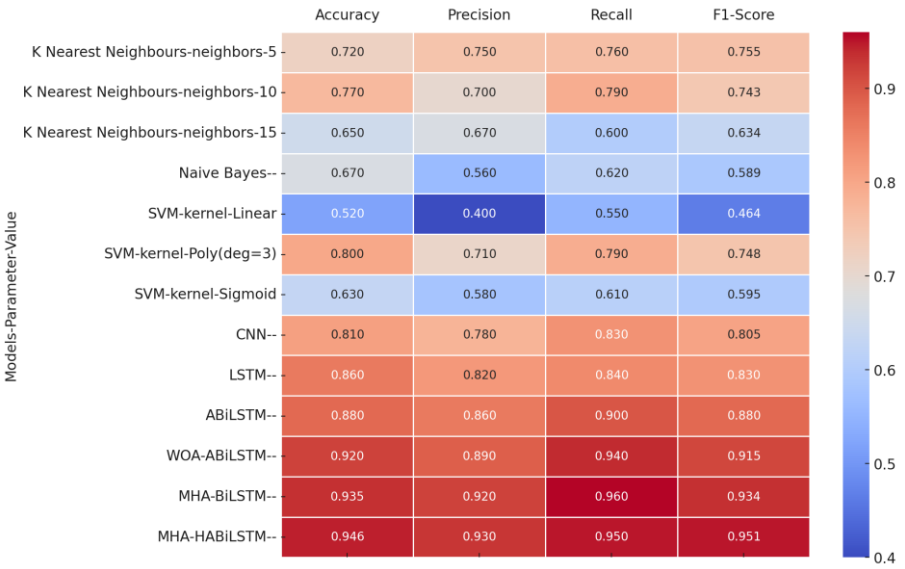
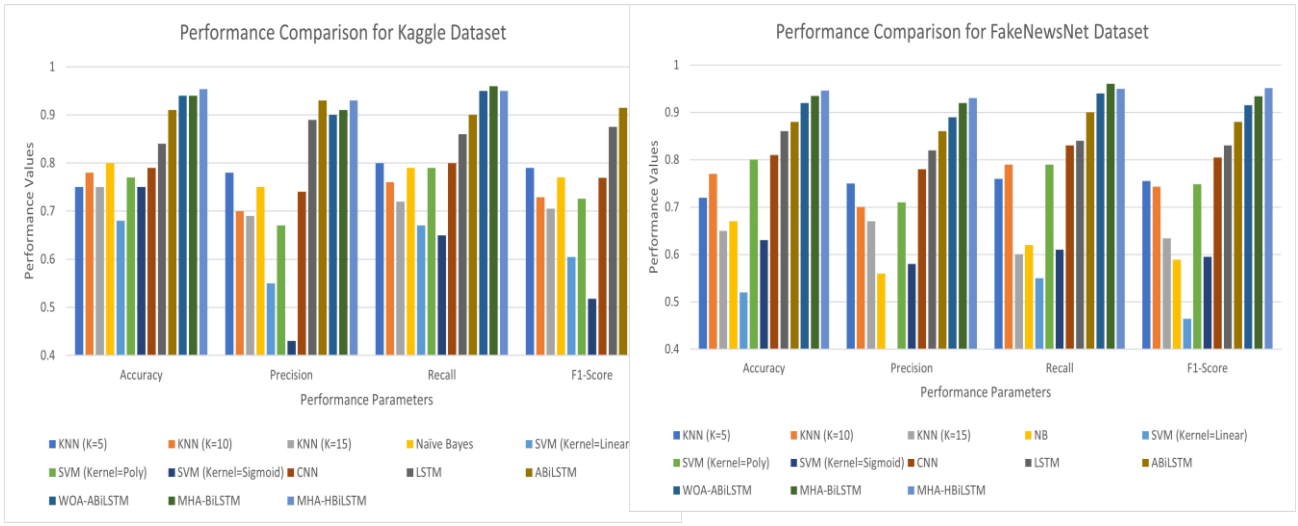
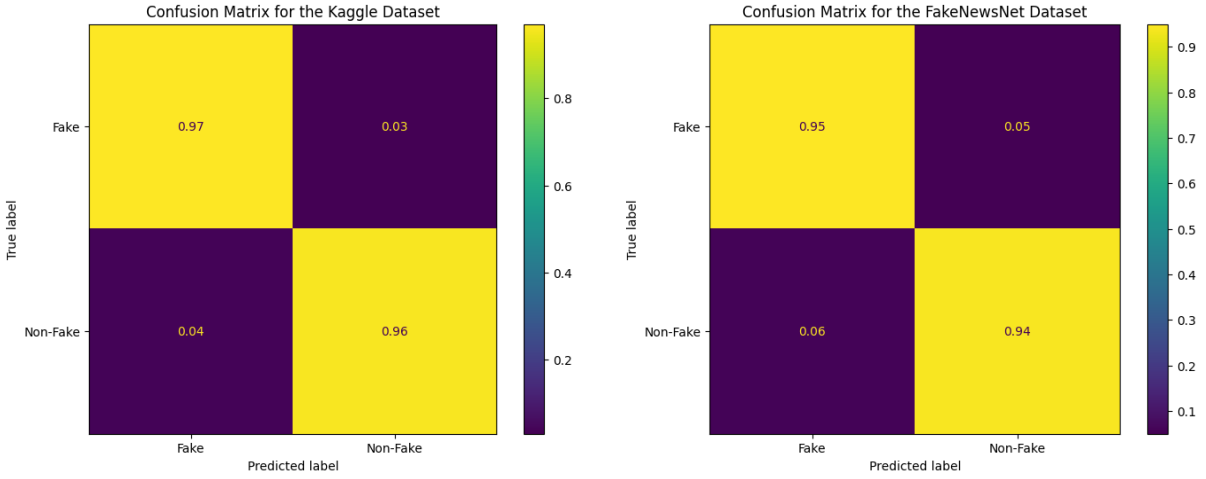


Figure 6. Heat Map-Performance Comparison of Proposed MHA-BiLSTM Model on FakeNewsNet (Shu et al., 2020)



(a) (b)

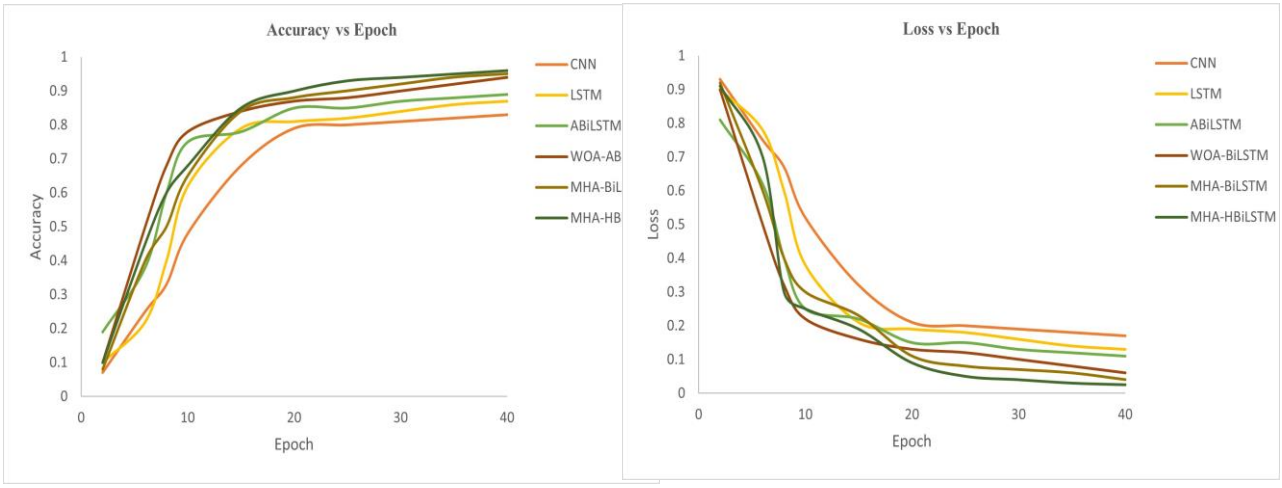
Figure. 7. Bar Diagram-Performance comparison of the proposed MHA-BiLSTM



(a) (b)

Figure 8. Confusion Matrix MHA-BiLSTM on Kaggle (Ahmed et al., 2017) and FakeNewsNet (Shu et al., 2020) Dataset

The confusion matrices presented in Figure 8 illustrate the performance of the proposed MHA-HBiLSTM algorithm in classifying tweets as either fake or non-fake. Each matrix provides a visual representation in which the columns correspond to actual class labels and the rows represent predicted labels. To ensure a fair evaluation, the entries in the confusion matrix are normalized based on the class distribution within the dataset. Specifically, Figure 8(a) highlights the evaluation metrics for the Kaggle fake news dataset (Ahmed et al., 2017). The diagonal elements represent correctly classified instances, including true positives (97%) and true negatives (96%). In contrast, the off-diagonal elements indicate misclassified cases, including false positives (3%) and false negatives (4%).



(a) (b)

Figure 9. Performance comparison of the MHA-BiLSTM in terms of Accuracy/Loss vs Epoch curve.

Similarly, the FakeNewsNet dataset (Shu et al., 2020) was analyzed, and the corresponding results are presented in Figure 8(b). The matrix shows true positives (95%) and true negatives (94%) as diagonal entries, indicating high classification accuracy. The off-diagonal elements include false positives (5%) and false negatives (6%), reflecting incorrectly labeled instances.

Upon concluding the assessment of the confusion matrices, it's deducible that the proposed MHA-BiLSTM model yields strong performance on both the Kaggle and FakeNewsNet datasets. The minimal rates of false positives and negatives reflect the model's proficiency in precise categorization and differentiation between legitimate and illegitimate news items in the datasets. An Accuracy versus Epoch graph was employed to gauge the model's performance during its training phase on the Kaggle dataset. This plot illustrates Training set performance as a function of the number of training epochs, as shown in Figure. 9(a). The accuracy curve illustrates the performance of different models, including CNN, LSTM, ABiLSTM, WOA-ABiLSTM, MHA-BiLSTM, and MHA-HBiLSTM, as the number of training epochs increases. Initially, in Epoch 2, all models have relatively low accuracy values ranging from 0.07 to 0.19. However, as training progresses, the accuracy steadily improves for all models. By Epoch 40, the MHA-BiLSTM and MHA-HBiLSTM models exhibit the highest accuracy, reaching 0.95 and 0.96, respectively. The LSTM and ABiLSTM models also significantly improve, achieving accuracy values of 0.87 and 0.89, respectively. Overall, the accuracy curve shows that the MHA-BiLSTM and MHA-HBiLSTM models consistently outperform the other models in terms of accuracy, indicating their superior performance in the task.

Similarly, we plot Loss versus Epoch, which provides valuable insights into how the model learns and improves over time, as shown in Figure 9(b). The y-axis charts the loss magnitude, while the x-axis is the number of training epochs. We can see that the MHA-HBiLSTM model's loss values decrease quickly compared with other models, including CNN, LSTM, ABiLSTM, WOA-BiLSTM, and MHA-BiLSTM, as the number of epochs increases. Initially, in Epoch 2, all models show high loss values ranging from 0.81 to 0.93. As training advances, all models' losses eventually reduce. By Epoch 40, the MHA-HBiLSTM model has the smallest loss value of 0.025, followed by the MHA-BiLSTM model's loss of 0.04. The LSTM and ABiLSTM models also show considerable improvements, with loss values of 0.13 and 0.11, respectively. The loss curve shows that the MHA-HBiLSTM and MHA-BiLSTM models consistently have the lowest loss, demonstrating a greater capacity to decrease mistakes throughout training.

Table 4 compares the accuracy of different optimization techniques applied to the MHA-HBiLSTM model. The experiments were conducted using a batch size 32 and training for 40 epochs. The Hybrid Whale-Multi-Verse Optimization (HW-MVO) approach obtained the maximum accuracy at 95%, according to the results. It fared better than the other methods, which were Grid Search at 77%, Random Search at 72%, Whale Optimization Algorithm (WOA) at 92%, and Hybrid Whale-Grey Wolf Optimization (HW-GWO) (Rathore et al.,

2020) at 93%. HW-MVO combines the Moth Flame Optimization and Whale Optimization techniques to enhance optimization performance. Overall, Table 4 shows how optimization strategies affect the accuracy of the MHA-HBiLSTM model, with HW-MVO being the most successful. Out of all the approaches that were taken into consideration, HW-MVO showed the best accuracy. In that order, HW-GWO, WOA, Grid Search, and Random Search came next.

Table 4. Analyzing Optimization Techniques' performance using MHA-HBiLSTM

Batch size	Epoch	Optimizer	Accuracy [%]
32	40	Stochastic Search (Bergstra & Bengio 2012)	73%
32	40	Parameter Grid Testing (Liashchynsky & Liashchynskyi 2019)	78%
32	40	WOA	91%
32	40	HW-GWO (Rathore et al., 2020)	93%
32	40	HW-MVO [Proposed]	95%

Figure 10 illustrates the accuracy progression of various optimization strategies applied to the Multi-Head Attention Hybrid Bi-Directional Long Short-Term Memory (MHA-HBiLSTM) model across 40 training epochs. The optimization techniques are visually distinguished using different colors: blue for Random Search, green for Grid Search, red for the Hybrid Whale–Gray Wolf Optimizer (HW-GWO), and orange for the proposed Hybrid Whale–Multi-Verse Optimizer (HW-MVO).

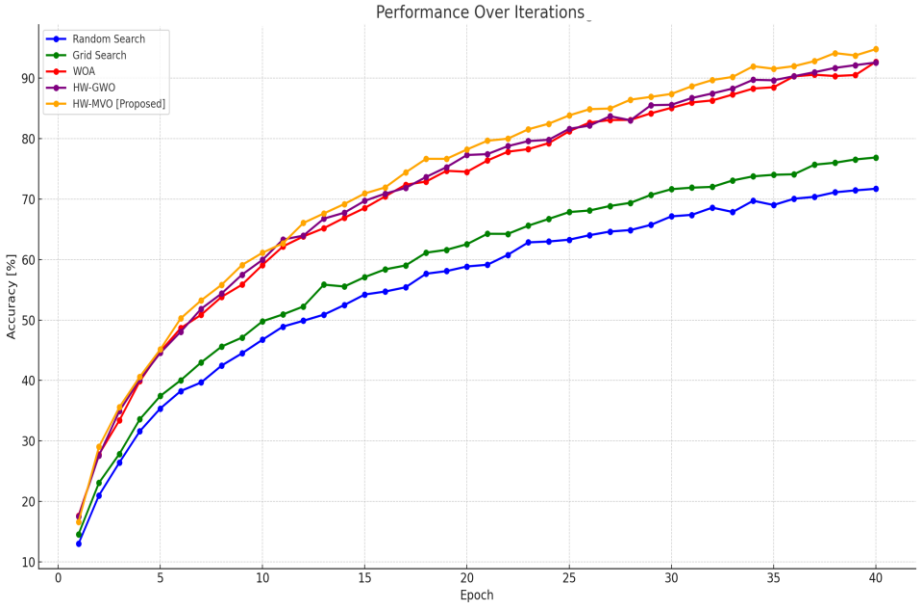


Figure 10. Benchmarking the proposed HW-MVO against Random Search, Grid Search, WOA, and HW-GWO algorithms

The accuracy starts at 0% for each optimizer and increases over time, reflecting the learnability from data. The curves exhibit some noise, indicating the variability that can occur due to different factors, such as the stochastic nature of the learning algorithm, the randomness in the selection of batches during training, or fluctuations in the data itself. There is a noticeable overlap among the curves, especially in the mid-epochs, where the performance of the optimizers converges. This overlap could suggest that during these epochs, the differences in optimizer strategies impact model performance. As the epochs progress, the increases in accuracy begin to taper off, reflecting a common phenomenon in training machine learning models where initial gains are significant and improvements become incremental as the model approaches its potential based on the given data and architecture. The final accuracy values indicated by the end points of the curves align with the expected outcomes, with the proposed HW-MVO achieving the highest accuracy of 95%, followed by HW-GWO at 93%, WOA at 92%, Grid Search at 77%, and Random Search at 72%.

Figure.11 shows the Computational Time Bar Chart to compare optimization techniques applied in training a Multi-Head Attention Hybrid Bi-Directional Long Short-Term Memory (MHA-HBiLSTM) model over 40 epochs, leveraging GPU acceleration. The chart categorizes each optimization technique—Random Search (RS), Grid Search (GS), Whale Optimization Algorithm (WOA), Hybrid Whale-Gray Wolf Optimizer (HW-GWO), and the proposed Hybrid Whale-Multi-Verse Optimizer (HW-MVO). The Y-axis measures the computational time required by each technique in minutes, serving as an efficiency indicator.

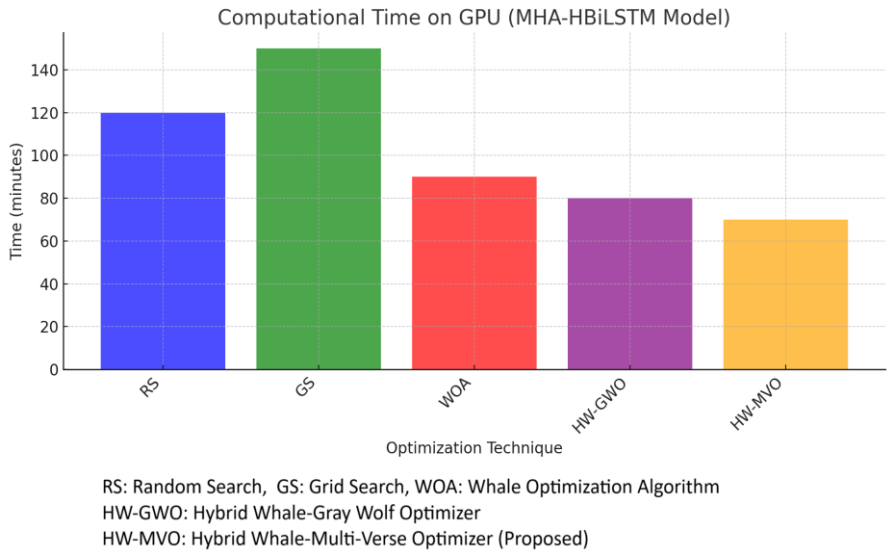


Figure 11. Computational Time comparison of the proposed HW-MVO with another optimization algorithm

The Random Search (RS) algorithm has the highest computational time among the techniques shown in Figure.11, which indicates its simple yet exhaustive approach. Similarly, Grid Search (GS) also has a significant computation time, indicative of a systematic and comprehensive search strategy. Whale Optimization Algorithm (WOA) requires less computing time than RS and GS, demonstrating the efficiency advantages of metaheuristic algorithms replicating intelligent natural behaviors. Hybrid Whale-Gray Wolf Optimizer (HW-GWO) exhibits a computational time that suggests a balance between search thoroughness and computational efficiency, likely due to the collaborative effects of combining strategies from several nature-inspired algorithms. The proposed Hybrid Whale-Multi-Verse Optimizer (HW-MVO) has less computation time than other RS, WOA, and HW-GWO methods, indicating it is the most computationally efficient optimizer in this comparison. This may be attributable to its effective use of GPU acceleration and possibly more advanced search and parallel processing optimizations. Figure. 11 accurately depicts the relative computing needs of different optimizers, revealing the trade-offs between time cost and technique sophistication.

Conclusion

This study proposes a novel Multi-Head Attention–Hierarchical Bidirectional Long Short-Term Memory (MHA-HBiLSTM) model for detecting fake news on Twitter. The proposed framework integrates user interaction metrics, demographic information, verification status, and tweet content to enhance detection performance. Additionally, a Hybrid Whale–Multi-Verse Optimization (HW-MVO) algorithm was introduced to optimize attention weights within the neural network. The model incorporates Hierarchical BiLSTM (HBiLSTM), GloVe word embeddings, and an advanced tweet preprocessing pipeline to capture latent semantic features. A late fusion strategy was employed to combine bidirectional hidden state features derived from tweet content, enabling more accurate classification. Experimental evaluations were conducted using two publicly available datasets—FakeNewsNet and Kaggle. The MHA-HBiLSTM model achieved classification accuracies of 96% and 94% on the Kaggle and FakeNewsNet datasets, respectively. These results highlight the model's effectiveness and robustness in distinguishing between fake and authentic news, outperforming both traditional and deep learning baselines.

Availability of Data and Materials

The datasets used and/or analyzed during the current study are available from the corresponding author upon reasonable request.

Author Contributions

Varalakshmi K. contributed to conceptualizing, designing, and implementing the HBLSTM model, including text preprocessing and feature representation. Ashok Kumar P. M. handled

the theoretical formulation, model optimization, and performance evaluation. Both authors collaborated on writing, reviewing, and refining the manuscript, ensuring the accuracy and effectiveness of the proposed methodology for fake news detection.

Conflict of interest

The authors declare no potential conflict of interest regarding the publication of this work. In addition, the ethical issues including plagiarism, informed consent, misconduct, data fabrication and, or falsification, double publication and, or submission, and redundancy have been completely witnessed by the authors.

Funding

The author(s) received no financial support for the research, authorship, and/or publication of this article.

References

Adedoyin, Z., & Mariyappan, B. (2022). Fake news detection using machine learning algorithms and recurrent neural networks. *Advance Preprint*, *1*(1), 1–31. https://doi.org/10.31124/advance.20751379.v1

Ahmed, H., Traore, I., & Saad, S. (2017). Detection of online fake news using N-gram analysis and machine learning techniques. In *Lecture Notes in Computer Science* (Vol. 10618, pp. 127–138). Springer. https://doi.org/10.1007/978-3-319-69155-8_9

Bahad, P., Saxena, P., & Kamal, R. (2019). Fake news detection using bidirectional LSTM recurrent neural network. *Procedia Computer Science*, *165*, 74–82. https://doi.org/10.1016/j.procs.2020.01.072

Bergstra, J., & Bengio, Y. (2012). Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, *13*, 281–305. https://dl.acm.org/doi/10.5555/2188385.2188395

Castillo, C., Mendoza, M., & Poblete, B. (2011). Information credibility on Twitter. In *Proceedings of the 20th International Conference on World Wide Web* (pp. 675–684). https://doi.org/10.1145/1963405.1963500

Chen, T., Li, X., Yin, H., & Zhang, J. (2018). Call attention to rumors: Deep attention-based recurrent neural networks for early rumor detection. In *Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining* (pp. 40–52). Springer. https://doi.org/10.48550/arXiv.1704.05973

Chen, X., Lian, C., Wang, L., Deng, H., Fung, S. H., Nie, D., Thung, K. H., Yap, P. T., Gateno, J., & Xia, J. J. (2019). One-shot generative adversarial learning for MRI segmentation of cranio-maxillofacial bony structures. *IEEE Transactions on Medical Imaging*, *39*(3), 787–796. https://doi.org/10.1109/TMI.2019.2935409

Chen, Y., Sui, J., Hu, L., & Gong, W. (2019). Attention-residual network with CNN for rumor detection. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management* (pp. 1121–1130). https://doi.org/10.1145/3357384.3357950

Ciampaglia, G. L., Shiralkar, P., Rocha, L. M., Bollen, J., Menczer, F., & Flammini, A. (2015). Computational fact-checking from knowledge networks. *PLoS ONE, 10*(6), Article e0128193. https://doi.org/10.1371/journal.pone.0128193

Conroy, N. J., Rubin, V. L., & Chen, Y. (2015). Automatic deception detection: Methods for finding fake news. In *Proceedings of the 78th ASIS&T Annual Meeting: Information Science with Impact: Research in and for the Community* (pp. 1–4). https://doi.org/10.1002/pra2.2015.145052010082

Dhiman, P., Kaur, A., & Bonkra, A. (2023). Fake information detection using deep learning methods: A survey. In *Proceedings of the 2023 International Conference on Artificial Intelligence and Smart Communication (AISC)* (pp. 858–863). https://doi.org/10.1109/aisc56616.2023.10085519

Dong, M., Yao, L., Wang, X., Benatallah, B., Sheng, Q. Z., & Huang, H. (2018). Dual: A deep unified attention model with latent relation representations for fake news detection. In *Proceedings of the International Conference on Web Information Systems Engineering* (pp. 199–209). https://doi.org/10.1007/978-3-030-02922-7_14

Guacho, G. B., Abdali, S., Shah, N., & Papalexakis, E. E. (2018). Semi-supervised content-based detection of misinformation via tensor embeddings. In *Proceedings of the IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining* (pp. 322–325). https://doi.org/10.48550/arXiv.1804.09088

Hu, G., Ding, Y., Qi, S., Wang, X., & Liao, Q. (2019). Multi-depth graph convolutional networks for fake news detection. In *Proceedings of the Natural Language Processing and Chinese Computing* (pp. 698–710). https://doi.org/10.1007/978-3-030-32233-5_54

Kim, Y. (2014). Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (pp. 1746–1751). https://doi.org/10.48550/arXiv.1408.5882

Konkobo, P. M., Zhang, R., Huang, S., Minoungou, T. T., Ouedraogo, J. A., & Li, L. (2020). A deep learning model for early detection of fake news on social media. In *Proceedings of the 7th International Conference on Behavioural and Social Computing (BESC)* (pp. 1–6). https://doi.org/10.1109/BESC51023.2020.9348311

Kuang, Q., & Xu, X. (2010). Improvement and application of TF-IDF method based on text classification. In *Proceedings of the IEEE International Conference on Internet Technology and Applications* (pp. 1–4). https://doi.org/10.1109/ITAPP.2010.5566113

Li, X., Lu, P., Hu, L., Wang, X., & Lu, L. (2021). A novel self-learning semi-supervised deep learning network to detect fake news on social media. *Multimedia Tools and Applications*, 1–9. https://doi.org/10.1007/s11042-021-11065-x

Liashchynskyi, P., & Liashchynskyi, P. (2019). Grid search, random search, genetic algorithm: A big comparison for NAS. https://doi.org/10.48550/arXiv.1912.06059

Lin, Z., Feng, M., dos Santos, C. N., Yu, M., Xiang, B., Zhou, B., & Bengio, Y. (2017). A structured self-attentive sentence embedding. In *Proceedings of the 5th International Conference on Learning Representations (ICLR)*. https://doi.org/10.48550/arXiv.1703.03130

Long, Y., Lu, Q., Xiang, R., Li, M., & Huang, C. R. (2017). Fake news detection through multi-perspective speaker profiles. In *Proceedings of the 8th International Joint Conference on Natural Language Processing* (pp. 252–256). https://aclanthology.org/I17-2043.pdf

Ma, J., Gao, W., & Wong, K. F. (2019). Detect rumors on Twitter by promoting information campaigns with generative adversarial learning. In *Proceedings of the International World Wide Web Conferences* (pp. 3049–3055). https://doi.org/10.1145/3308558.3313741

Ma, J., Gao, W., Wei, Z., Lu, Y., & Wong, K. F. (2015). Detect rumors using time series of social context information on microblogging websites. In *Proceedings of the 24th ACM International*

Conference on Information and Knowledge Management (CIKM) (pp. 1751–1754). https://doi.org/10.1145/2806416.2806607

Ma, Z., Yu, C., & Hu, B. (2018). Detecting rumors from microblogs with recurrent neural networks. In *Proceedings of the 26th International Conference on World Wide Web* (pp. 1531–1539). https://ink.library.smu.edu.sg/sis_research/4630/

McCallum, A., & Nigam, K. (1998). A comparison of event models for naive Bayes text classification. In *Proceedings of the Workshop on Learning for Text Categorization* (pp. 91–98). http://www.kamalnigam.com/papers/multinomial-aaaiws98.pdf

Naithani, K., Raiwani, Y. P., Alam, I., & Aknan, M. (2023). Analyzing hybrid C4.5 algorithm for sentiment extraction over lexical and semantic interpretation. *Journal of Information Technology Management*, *15*(Special Issue), 57–79. https://doi.org/10.22059/jitm.2023.95246

Potthast, M., Köpsel, S., Stein, B., & Hagen, M. (2017). A stylometric inquiry into hyperpartisan and fake news. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management* (pp. 167–176). https://doi.org/10.18653/v1/P18-1022

Qiu, S., Zhao, Y., Jiao, J., Wei, Y., & Wei, S. (2019). Referring image segmentation by generative adversarial learning. *IEEE Transactions on Multimedia*, *22*(5), 1333–1344. https://doi.org/10.1109/TMM.2019.2942480

Rathore, R. S., Sangwan, S., & Prakash, S. (2020). Hybrid WGWO: Whale grey wolf optimization-based novel energy-efficient clustering for EH-WSNs. *Journal of Wireless Communications and Networking*, *2020*(101). https://doi.org/10.1186/s13638-020-01721-5

Ruchansky, N., Seo, S., & Liu, Y. (2017). CSI: A hybrid deep model for fake news detection. In *Proceedings of the 26th International Conference on World Wide Web* (pp. 797–806). https://doi.org/10.1145/3132847.3132877

Samadi, M., & Momtazi, S. (2023). Multichannel convolutional neural networks for detecting COVID-19 fake news. *Digital Scholarship in the Humanities*, *38*(1), 379–389. https://doi.org/10.1093/llc/fqac023

Shikalgar, M. B., & Arage, C. S. (2023). Fake news detection using hybrid BiLSTM-TCN model with attention mechanism. In *Proceedings of the 2023 2nd International Conference on Applied Artificial Intelligence and Computing (ICAAIC)* (pp. 1130–1136). https://doi.org/10.1109/ICAAIC56838.2023.10140491

Shu, K., Mahudeswaran, D., Wang, S., & Liu, H. (2017). Fake news detection on social media: A data mining perspective. *ACM SIGKDD Explorations Newsletter*, *19*(1), 22–36. https://doi.org/10.1145/3137597.3137600

Shu, K., Mahudeswaran, D., Wang, S., Lee, D., & Liu, H. (2020). FakeNewsNet: A data repository with news content, social context, and spatiotemporal information for studying fake news on social media. *Big Data*, *8*(3), 171–188. https://doi.org/10.48550/arXiv.1809.01286

Shu, K., Wang, S., & Liu, H. (2018). Understanding user profiles on social media for fake news detection. In *Proceedings of the 2018 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)* (pp. 430–435). https://doi.org/10.1109/MIPR.2018.00092

Singh, M. K., & Kumar, A. (2023). Cucumber leaf disease detection and classification using a deep convolutional neural network. *Journal of Information Technology Management*, *15*(Special Issue: Intelligent and Security for Communication, Computing Application (ISCCA-2022)), 94–110. https://doi.org/10.22059/jitm.2023.95248

Tanuku, S. R. (2022). Novel approach to capture fake news classification using LSTM and GRU networks. In *Proceedings of the 2022 International Conference on Futuristic Technologies (INCOFT)* (pp. 1–4). https://doi.org/10.1109/incoft55651.2022.10094467

Trueman, T. E., Kumar, A. J., Narayanasamy, P., & Vidya, J. (2021). Attention-based C-BiLSTM for fake news detection. *Applied Soft Computing*, *110*, 107600. https://doi.org/10.1016/j.asoc.2021.107600

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. In *Proceedings of the Neural Information Processing Systems (NIPS)*. https://doi.org/10.48550/arXiv.1706.03762

Vosoughi, S., Roy, D., & Aral, S. (2018). The spread of true and false news online. *Science*, *359*, 1146–1151. https://doi.org/10.1126/science.aap9559

Wang, W. Y. (2017). "Liar, liar pants on fire": A new benchmark dataset for fake news detection. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL)* (Vol. 2, pp. 422–426). https://doi.org/10.18653/v1/p17-2067

Wu, L., Rao, Y., Jin, H., Nazir, A., & Sun, L. (2019). Different absorption from the same sharing: Sifted multi-task learning for fake news detection. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)* (pp. 4644–4653). https://arxiv.org/pdf/1909.01720.pdf

Yu, F., Liu, Q., Wu, S., Wang, L., & Tan, T. (2019). Attention-based convolutional approach for misinformation identification from massive and noisy microblog posts. *Computers & Security*, *83*, 106–121. https://doi.org/10.1016/j.cose.2019.02.003

Yuan, C., Ma, Q., Zhou, W., Han, J., & Hu, S. (2020). Early detection of fake news by utilizing the credibility of news, publishers, and users based on weakly supervised learning. In *Proceedings of the 28th International Conference on Computational Linguistics (COLING)* (pp. 5444–5454). https://doi.org/10.48550/arXiv.2012.04233

Zhang, D., & Lee, W. S. (2003). Question classification using support vector machines. In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 26–32). https://doi.org/10.1145/860435.860443

Zhang, Y., Chen, W., Yeo, C. K., Lau, C. T., & Lee, B. S. (2017). Detecting rumors on online social networks using multi-layer autoencoder. In *Proceedings of the 2017 IEEE Technology & Engineering Management Conference (TEMSCON)* (pp. 437–441). https://doi.org/10.1109/TEMSCON.2017.7998415

Bibliographic information of this paper for citing:

K., Varalakshmi, & P. M., Ashok Kumar (2025). Enhancing Fake News Detection by Attention-Based BiLSTM and Hybrid Whale-Multi-Verse Optimization. *Journal of Information Technology Management*, 17 (Special Issue), 168-197. https://doi.org/10.22059/jitm.2025.102975