© MAR 2026 | IRE Journals | Volume 9 Issue 9 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I9-1714856

BiLAT: A BiLSTM Attention Transformer Model with Hyperparameter Optimization for Robust Fake News Detection

T. POORNIMA
Research Scholar (Part time) Department of Computer Science, Sri Ramakrishna Mission Vidyalaya College of Arts and Science, (Affiliated to Bharatiyar University) Periyanaickenpalayam, Coimbatore

Abstract- ***The rapid spread of misinformation on social media, particularly Twitter, poses major problems to information reliability and public trust. This paper includes a complete comparative examination of six deep learning models such as RNN, CNN, BERT, GRU, LSTM, and a suggested BiLSTM Attention Transformer (BiLAT) assessed across three benchmark datasets: Fake and Real News, FakeNewsNet, and ISOT Fabricated News. We systematically test five hyperparameter optimisation methods (Grid Search (GS), Random Search (RS), Bayesian Optimisation (BO), Genetic Algorithm (GA), and BOHB) to see how they affect model performance. Results reveal that transformer-based architectures greatly outperform traditional models, with BiLAT obtaining state-of-the-art performance, including 99% accuracy on FakeNewsNet under BOHB optimization. BOHB consistently gives the best performance gains across all models and datasets, with an accuracy boost of 2–5% over traditional optimisation methods. The findings suggest that integrating advanced transformer topologies with efficient hyperparameter optimization considerably boosts the ability to grasp linguistic intricacies inherent in disinformation. This paper shows the importance of architectural design and optimization technique in constructing effective, scalable fake news detection systems for social media contexts.***

Keywords: Fake News Detection, Deep Learning Models, Hyperparameter Optimization, Transformer Architectures, BOHB

I. INTRODUCTION

The quick spread of false information on social media, especially Twitter, makes it hard for people to talk to each other, trust each other, and believe the information they get. false news is getting more and more complicated, thus detection systems need to be able to pick up on linguistic subtleties, contextual indicators, and complex patterns that are unique to false news. Previous research has investigated multimodal and knowledge-based methodologies; nonetheless, substantial constraints persist in text-based detection, especially concerning model optimisation, scalability, and cross-domain generalisation.

This discovery is significant as proficient automated identification of fake news might alleviate the societal repercussions of disinformation, facilitate informed decision-making, and improve the reliability of digital communication systems. This work immediately enhances online information integrity and safeguards consumers from deceptive content by using comprehensive, efficient, and precise detection mechanisms.

This work rigorously assesses six deep learning models using three benchmark datasets to fill current gaps. It also looks into five ways to optimise hyperparameters and shows how BOHB might improve model performance. [1]

Contributions:

- Comparing six deep learning architectures for finding fake news in a number of datasets.
- Introducing BiLAT, which uses both bidirectional sequence modelling and attention mechanisms to better grasp the context.
- Showing that extensive hyperparameter optimisation, especially BOHB, makes models far more accurate and generalisable.

This paper offers methodological insights and practical direction for the development of scalable,

high-performance disinformation detection algorithms on social media platforms.

II. LITERATURE REVIEW

Recent progress in disinformation research has spurred the creation of more advanced multimodal and cross-lingual detection systems. Kumari and Singh (2024) suggested a deep learning-based architecture that combines advanced NLP pipelines, DeepL translation for linguistic normalisation, and vectorization-driven feature extraction. Their solution uses LSTM networks for semantic modelling and CLIP for contextual visual encoding. It also has a single decision layer that makes it possible to accurately classify several modes. The model got 99.22% accuracy for text-only inputs and 93.12% accuracy for text and image data combined, which is better than current baselines [2]. Nair et al. (2024) developed a knowledge-centric deep learning system for identifying fake information on Twitter. The method builds a knowledge base that includes SPO triplets, sentiment polarity, frequency characteristics, and topics obtained by LDA. The study employs Named Entity Recognition (NER), topic modelling, and various Deep Learning architectures, including RNN, GRU, LSTM, GPT-3, and BERT, concluding that BERT is the best successful classifier. The combination of NLP, information retrieval, and graph-theoretic reasoning makes it possible to automatically check facts and makes detection more reliable [3]. Yan, Fu, and Wu (2024) introduced BTCM, a multimodal detection model augmented by a 1D-CCNet attention mechanism to promote cross-modal interaction. The model uses BERT to encode text and BLIP-2 to encode images and make captions. It then uses a heterogeneous fusion module to combine the two types of stimuli. Tests on Twitter, Weibo, and Gossipcop datasets show that these new multimodal methods work far better than the ones that are already out there.[4].

Zhu et al. (2024) created IFIS, a multimodal architecture that uses entity identification and dual co-attention methods to balance feature aggregation within a modality with semantic fusion between modalities. Tests on Weibo and Twitter showed that coordinated multimodal feature integration improved accuracy by 0.6% and 0.58% above the best benchmarks, proving its worth [5]. M., Ahmad, Pamidimukkala et al. (2025) presented a Modified Transformer integrated with PSODO, a hybrid Particle Swarm–Dandelion Optimisation method aimed at addressing delayed convergence and local optimum issues in multimodal learning. The model was trained through three refinement steps and successfully identifies cross-modal discrepancies, achieving 96.8% accuracy on benchmark datasets, surpassing traditional detectors and LVLMs [6].

At the same time, Sallah et al. (2024) looked at how to find social bots utilising pretrained language models like BERT and GPT-3 along with a feedforward classifier. Their model got an F1-score of 93% and was 3–24% better than Word2Vec and GloVe. They used XAI approaches to make it easier to understand [7]. Huang et al. (2025) presented semi-supervised relational graph attention (SRGAT), a transformer that integrates user metadata, tweet content, and multi-relational network architectures. The model showed improvements of up to 5.4% in accuracy and 4.8% in F1-score across three datasets. This was made possible by consistency loss, which makes the model more robust with minimal labelled data [8]. Koca and Cicekli (2024) investigated cross-domain generalisation issues by assessing classical machine learning and deep learning models across various topic domains. Their results indicate that conventional ML experiences accuracy declines of up to 20% during domain transitions, but DL models sustained over 85% accuracy and over 82% F1-scores, underscoring their enhanced transferability [9].

Almandouh et al. (2024) performed extensive assessments on Arabic datasets employing FastText embeddings, transformer models, and hybrid deep networks. The BiGRU–BiLSTM model exhibited superior performance, achieving accuracies of 0.98 and 0.99 on AFND and ARABICFAKETWEETS, respectively, indicating its robustness for morphologically intricate languages [10]. Lastly, Soga et al. (2024) put forth a system based on a graph transformer that combines user attitude similarity with modelling propagation structure. The method achieved 92.3% accuracy and a 91.7% F1-score across various datasets, surpassing existing GNN algorithms by almost 4%, and accurately capturing stance-driven misinformation transmission [11].

© MAR 2026 | IRE Journals | Volume 9 Issue 9 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I9-1714856

TABLE.1. LITERATURE REVIEW

Author(s) & Year	Dataset(s)	Methodology	Research Gap
Kumari & Singh (2024)	Multilingual & multimedia fake news datasets	Multimodal DL Framework (LSTM + CLIP)	The model lacks robustness under domain shift and its performance degrades when image quality is low.
Nair et al. (2024)	Twitter fake news dataset	Knowledge-Based DL Framework	The approach requires high computational effort to build the knowledge base and demonstrates limited cross-lingual adaptability.
Yan, Fu & Wu (2024)	Twitter, Weibo, Gossipcop	BTCM Model	The model is computationally expensive and its accuracy is affected by low-quality or noisy social media images.
Zhu et al. (2024)	Weibo, Twitter	IFIS Model	The performance improvement is marginal and the model relies heavily on complex attention tuning.
M, Ahmad, Pamidimukkala et al. (2025)	Synthetic & benchmark multimodal datasets	Modified Transformer + PSODO	The use of synthetic training data limits real-world generalization and may not fully capture true misinformation patterns.
Sallah et al. (2024)	Twitter bot dataset	Transformer-based PLM Model	The model requires frequent retraining due to evolving bot behaviour, which reduces long-term effectiveness.
Huang et al. (2025)	Real-world social bot datasets	SRGAT Model	The model suffers from high graph construction overhead and performs poorly on sparse or noisy social networks.
Koca & Cicekli (2024)	Twitter + multi-domain news datasets	ML/DL Cross-Domain Analysis	The models struggle with domain transfer, and traditional ML methods especially show significant accuracy drops across domains.
Almandouh et al. (2024)	AFND, ARABICFAKETWEETS	Hybrid DL Models (Bi-GRU-Bi-LSTM etc.)	The models are limited to Arabic content and do not address cross-dialect or cross-language generalization.
Soga et al. (2024)	Proprietary Twitter dataset, FibVID	Graph Transformer Model	The dataset scope is narrow, and attitude similarity features may not generalize across different social platforms.

© MAR 2026 | IRE Journals | Volume 9 Issue 9 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I9-1714856

III. MATERIALS AND METHODOLOGY

This section delineates the process for identifying false information on Twitter utilising deep learning models. It first presents three benchmark datasets: Fake and Real News, FakeNewsNet, and the ISOT Fabricated News Dataset, which offer tagged textual and contextual information for model development. The workflow delineates the text pre-processing procedures employed to cleanse and standardise twitter data. Various deep learning classification models are utilised, including RNN, LSTM, GRU, BiLSTM, CNN, BERT, and the hybrid BiLAT, chosen for their efficacy in sequence modelling and contextual representation. The section finishes with a summary of five hyperparameter optimisation techniques, employed to enhance performance and assure robust generalisation. [12]

3.1 Dataset Description

i. Fake and Real News Dataset: The Fake and Real News Dataset functions as a significant benchmark in deep learning research focused on the detection of fake news and the examination of disinformation. The dataset comprises two separate files: fraudulent.csv, including 23,502 entries classified as fraudulent, and True.csv, encompassing 21,417 entries classified as authentic. Each sample contains multiple noteworthy qualities.

Table 2: Description of Dataset Attributes

Attribute	Explanation
Title	The headline or main title of the news article
Text	The complete textual content of the article
Subject	The thematic category or classification of the news
Date	The date when the article was published

This dataset is optimal for training and evaluating deep learning models in text classification tasks, including binary false news detection. Their proportions and balance enable the effective acquisition of complex verbal structures and semantic nuances, hence supporting the development of robust neural network frameworks for misinformation detection. (https://www.kaggle.com/datasets/clmentbisaillon/fake-and-real-news-dataset).

ii. FakeNewsNet: This dataset from Arizona State University (ASU) supports research on the detection of misinformation on Twitter. It augments the FakeNewsNet platform by integrating extensive news material with social context data, enabling deep learning models to leverage diverse information sources.

Labels:

- Fake: Articles containing misinformation.
- Real: Confirmed credible news.

Table.3. Core Dataset Attributes

Category	Attribute	Definition
News Content	Source	Author or publisher of the article
	Headline	Concise summary reflecting the main news topic
	Body_text	Full text detailing the article's claims
	Image_video	Visual media supporting the news story
Social Context	User_profile	Basic information about Twitter users interacting with news
	User_content	Recent tweets posted by these users
	User_followers	Followers of the involved Twitter accounts
	User_followees	Users followed by these accounts

This dataset facilitates the development of advanced deep learning models, including graph neural networks and multimodal fusion networks, which utilise textual, visual, and social signals to improve fake news detection on Twitter. (https://www.kaggle.com/datasets/mdepak/fakenewsnet)

© MAR 2026 | IRE Journals | Volume 9 Issue 9 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I9-1714856

iii. ISOT Fabricated News Dataset: The ISOT Fake News Dataset is a balanced compilation of genuine and false news articles, designed to assist with binary text classification tasks for disinformation detection, particularly relevant to social media platforms like Twitter.

Dataset Summary

- Total Articles: Exceeding 25,000
- Real News: Approximately 12,600 articles from Reuters.com
- Fake News: Approximately 12,600 materials from unreliable sources as identified by Politifact and Wikipedia

The dataset focusses on news articles published between 2016 and 2017, a critical period for the dissemination of misinformation.

Table 4: Attributes and Descriptions

Attribute	Description
Title	News headline; reflects short, tweet-like messaging
Text	Full content of the news article
Subject	Category/topic of the article (e.g., Politics, World)
Date	Publication date of the article
Label	Binary label: Real (1), Fake (0)

This dataset is optimal for deep learning models, enabling accurate detection of fake news through the integration of headlines and article content. It serves as a crucial resource for scholars developing NLP models to combat misinformation on platforms like Twitter.
(https://www.kaggle.com/datasets/emineyetm/fake-news-detection-datasets)

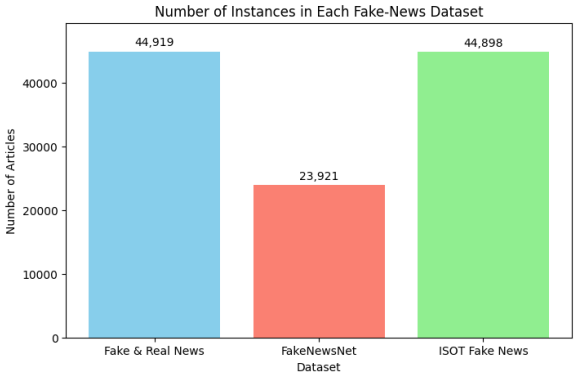

Fig.1. Instance Distribution Across Datasets

This figure displays a bar chart that contrasts the total incidences across three benchmark datasets utilised for fake news detection: the Fake & Real News Dataset, FakeNewsNet, and the ISOT Fake News Dataset. The Fake & Real News and ISOT databases have almost identical and significantly higher quantities of articles (about 45,000 each), but FakeNewsNet contains a relatively smaller collection of about 24,000 instances. This comparison underscores the disparity in dataset scale, which directly affects model training depth and generalisation efficacy.

3.2. Text Pre-processing

This figure displays a bar chart that contrasts the total incidences across three benchmark datasets utilised for fake news detection: the Fake & Real News Dataset, FakeNewsNet, and the ISOT Fake News Dataset. The Fake & Real News and ISOT databases have almost identical and significantly higher quantities of articles (about 45,000 each), but FakeNewsNet contains a relatively smaller collection of about 24,000 instances. This comparison underscores the disparity in dataset scale, which directly affects model training depth and generalisation efficacy. [13]

Raw Tweet: RT @CNN: Trump announces a new economic plan for 2024! US Read more: https://t.co/abcd123

The tweet is analysed sequentially utilizing the subsequent equations:

1. Noise Element Elimination: Exclude URLs, mentions, and retweet indicators from the original tweet content C.

$$\mathcal{N}_1 = \{URLs, @mentions, \text{RT}\}$$

$$\mathcal{C}_1 = \mathcal{C} \setminus \mathcal{N}_1$$

Example: Trump announces a new economic plan for 2024!

2. Elimination of Punctuation and Numerals: Exclude punctuation and numerical characters from the sanitized material $\mathcal{C}_1$

$\mathcal{N}_1 = \{punctuation, digits\}$

$$\mathcal{C}_2 = \mathcal{C}_1 \setminus \mathcal{N}_2$$

Example: Trump announces a new economic plan for

3. Tokenization: Divide the text into tokens $\mathcal{T} = Tokenize\ (\mathcal{C}_2)$

Example: $\mathcal{T}$ = ["Trump","announces","a","new","economic","plan", "for"]

4. Lemmatization: Convert each token $t \in \mathcal{T}$ to its base form

Example: $\mathcal{T}_1$ = ["Trump","announce","a","new","economic","plan","for"]

5. Stop word Removal: Remove stop words $\mathcal{S}$ from the lemmatized tokens:

$$\mathcal{T}_2 = \mathcal{T}_1 \setminus \mathcal{S}$$

Example: $\mathcal{T}_2$ = ["Trump","announce","new","economic","plan"]

3.3. DL Classification

We utilised various deep learning models, including RNN, LSTM, GRU, BiLSTM, CNN, BERT, and BiLAT, to classify tweets as either fraudulent or genuine. RNN models are employed for sequential input processing, while LSTM and GRU improve memory retention through gating mechanisms. BiLSTM improved contextual understanding by evaluating sequences in both directions. Convolutional Neural Networks proficiently identified local patterns and textual characteristics. BERT, utilising a transformer-based architecture, delivered strong contextual embeddings. The BiLAT model integrated BiLSTM with an attention mechanism to highlight the most relevant segments of the tweet, hence enhancing classification effectiveness. [14]

i. Recurrent Neural Networks (RNNs)

RNN) are specialised deep learning architectures designed for modelling sequential data, making them especially effective for natural language processing applications, such as identifying false news on Twitter. Tweets, represented as sequences of word embeddings $\{X_1, X_2, \ldots, X_T\}$, where each $X_t \in \mathbb{R}^d$, are examined sequentially to discern temporal dependencies and contextual insights. At each time step t, the hidden state $H_t \in \mathbb{R}^d$, is modified by incorporating the current input and the prior hidden state according to the formula

$$H_t = tanh(W_{xh}X_t + W_{hh}H_{t-1} + b_h)$$

Here, $W_{xh} \in \mathbb{R}^{h\times d}$ modifies the input embedding, $W_{hh} \in \mathbb{R}^{h\times h}$ encapsulates temporal dependencies from prior states, and $b_h \in \mathbb{R}^h$ represents a bias term. The hyperbolic tangent function enables nonlinear transformation and restricts hidden state values to the interval of -1 to 1. Upon concluding the Twitter sequence processing, the final hidden state H_T encapsulates a learnt representation of the tweet's semantic and syntactic characteristics. The embedding is then converted into a scalar logit via the equation: $z = W_{hy}H_T + b_y$, where $W_{hy} \in \mathbb{R}^{1\times h}$ projects the hidden representation into the output space, and $b_y \in \mathbb{R}$ represents the output bias. The estimated chance yˆ that the tweet is fraudulent is derived by utilizing the sigmoid activation function

$$\hat{y} = \sigma(z) = \frac{1}{1 + e^{-z}}$$

Generating an outcome inside the range [0,1]. The model is trained by reducing the binary cross-entropy loss between ŷ and the actual label. $y \in \{0, 1\}$.

$$L = -(y \log \hat{y} + (1 - y) \log (1 - \hat{y}))$$

Backpropagation Through Time (BPTT) is employed to calculate the gradients of the loss concerning the model parameters $\theta = \{W_{xh}, W_{hh}, W_{hy}, b_h, b_y\}$. The gradient concerning the output weight is $\frac{\partial L}{\partial W_{hy}} = \delta^{out} H_T^{\top}$, where $\delta^{out} = \hat{y} - y$, and the gradient pertaining to the final hidden state is $\frac{\partial L}{\partial H_T} = W_{hy}^{\top} \delta^{out}$. The error signal ∂_t at each time step is transmitted backward in time as

$$\delta_t = (W^{\top}{}_{hh}\, \delta_{t+1}) \circ (1 - H_t^2) + \frac{\partial L}{\partial H_T}$$

Where $\circ$ signifies element wise multiplication and $(1 - H_t^2)$ represents the derivative of the tanh activation function. Gradients concerning the input, recurrent weights, and biases aggregate across the sequence as

$$\frac{\partial L}{\partial W_{hy}} = \sum_{t=1}^{T} \delta_t X_t^{\top}, \qquad \frac{\partial L}{\partial W_{hh}} = \sum_{t=1}^{T} \delta_t H_{t-1}^{\top},$$

$$\frac{\partial L}{\partial b_h} = \sum_{t=1}^{T} \delta_t$$

Ultimately, parameters are adjusted by gradient descent or its derivatives as follows: $\Theta := \Theta - \eta \frac{\partial L}{\partial \Theta}$, where η represents the learning rate. This rigorous formulation enables the RNN to recognise complex sequential linkages and semantic indicators inside tweets, crucial for distinguishing false news from genuine information despite the shorthand and informal nature of Twitter language. [15]

ii. Convolutional Neural Networks (CNNs)

Convolutional Neural Networks (CNNs) have demonstrated significant effectiveness in text classification by leveraging their ability to extract local, position-invariant features. Convolutional Neural Networks (CNNs) excel in identifying false news on Twitter due to their proficiency in recognising semantic patterns and linguistic signals within concise text segments such as tweets. A tweet is defined as a series consisting of n tokens. Each token is linked to a d-dimensional embedding vector, resulting in an input matrix: $X \in \mathbb{R}^{n\,X\,d}$. A convolutional layer employs a collection of K filters $\{W_k\}_{k=1}^{K}$, each with dimensions $h \times d$, where h denotes the kernel height (i.e., window size). The convolution operation at position i for a specified filter W_k is defined as

$$c_i^{(k)} = f\langle W_k,\ X_{i:i+h-1}\rangle + b_k)$$

Here, $\langle \cdot, \cdot \rangle$ represents the Frobenius inner product, b_k signifies the bias term for the k-th filter, and $f(\cdot)$ specifies a non-linear activation function, commonly ReLU:

$$f(z) = max(0, z)$$

This generates a feature map $\hat{c}^{(k)} \in \mathbb{R}^{n-h+1}$. To diminish dimensionality and emphasize the most prominent features, a max-pooling procedure is implemented:

$$\hat{c}^{(k)} = \max (c^{(k)})$$

The output characteristics from all K filters are amalgamated into a vector:

$$z = [\, \hat{c}^{(1)}, \hat{c}^{(2)}, \ldots\ldots, \hat{c}^{(K)}]^{\top} \in \mathbb{R}^{K}$$

The vector is transmitted through a fully connected layer with weight matrix $W_{fc} \in \mathbb{R}^{K\,X\,1}$ and bias b_{fc}, subsequently using a sigmoid activation to derive the probability of fake news.

$\hat{y} = \sigma(W_{fc} z + b_{fc})$, where $\sigma(x) = \frac{1}{1+e^{-x}}$

The model is refined utilizing binary cross-entropy loss:

$$L = -(y \log \hat{y} + (1 - y) \log (1 - \hat{y}))$$

CNN-based models excel at identifying fake news on Twitter by recognising n-gram level distinguishing features, such as misleading language, punctuation cues, and repetitive word usage. Employing dropout regularisation before the final dense layer and utilising parameter sharing allows CNNs to reduce overfitting and improve scalability. Their spatial invariance ensures effective feature recognition regardless of word position, while their parallel processing

capabilities offer improved computational efficiency relative to sequential models like RNNs. This makes CNNs ideal for real-time, high-throughput social media analysis, enabling effective and scalable classification of false news. [16]

iii. Bidirectional Encoder Representations from Transformers (BERT)

BERT is a deep bidirectional Transformer encoder that concurrently captures contextual dependencies from both the left and right of a token. Its architecture enhances language understanding, particularly proficient in jobs requiring contextual reasoning. Each input sequence begins with a unique classification token [CLS], followed by the input text and a separator token [SEP]. Each token is represented as the sum of its token embedding, segment embedding, and positional encoding.

$$x_i = E_{token}(t_i) + E_{segment}(s_i) + E_{position}(p_i)$$

Where, $E_{token}(t_i)$ represents the embedding of the i^{th} token, $E_{segment}(s_i)$ denotes the token's affiliation with phrase A or B, and $E_{position}(p_i)$ encodes the token's position within the sequence. BERT comprises multiple layers of Transformer encoders, each featuring multi-head self-attention processes and feedforward neural networks with residual connections. The self-attention mechanism calculates attention scores to determine the contextual relevance among tokens by employing:

$$Attention(Q, K, V) = softmax\left(\frac{QK^{\top}}{\sqrt{d_k}}\right)V$$

Where, $Q = XW^Q, K = XW^K$, and $V = XW^V$ represent linear projections of the input X utilizing learnable matrices W^Q, W^K, and W^V, and d_k denoting the dimension of the key vectors. BERT is initially pre-trained utilising two unsupervised objectives. Masked Language Modelling (MLM) involves the arbitrary masking of 15% of input tokens and forecasting them based on the contextual information surrounding them. The corresponding loss function is:

$$L_{MLM} = -\sum_{i \in M} log\, P(\, t_i \mid X_{masked}\,)$$

Here, M denotes the indices of the masked tokens, whereas X_{masked} indicates the input in which the masked tokens are replaced by a special token. The secondary pre-training objective is Next Sentence Prediction (NSP), which assists the model in understanding relationships between sentence pairs. The model assesses whether sentence B logically follows sentence A, utilising the binary cross-entropy loss function.

$$L_{NSP} = -[\, y\, log(\hat{y}) + (1 - y)\backslash log(1 - \hat{y})]$$

Where $y \in \{0, 1\}$ is The ground truth label indicating the continuation of the sentence pair. BERT is fine-tuned for tweet veracity classification by utilising a task-specific classification layer on the final hidden state of the [CLS] token. The representation is analysed by a sigmoid-activated dense layer to predict the likelihood that a tweet is genuine or fraudulent:

$$\hat{y} = \sigma left(\, W_{cls} \cdot h_{[CLS]} + b\,)$$

Where $h_{[CLS]}$ represents the contextual embedding of the [CLS] token, W_{cls} denotes the classifier weight matrix, b signifies the bias term, and σ indicates the sigmoid function. The associated loss function for binary classification is

$$L_{tweet} = -[\, y\, log(\hat{y}) + (1 - y) log(1 - \hat{y})]$$

This fine-tuning enables BERT to adapt to the tweet classification task by learning from labelled data, hence enhancing its sensitivity to linguistic patterns typical of deception. BERT's capacity to effectively articulate both semantic and syntactic features makes it particularly adept at handling the informal, abbreviated, and context-sensitive nature of social media content, including elements such as sarcasm, coded language, and rhetorical devices.[17]

iv. Gated Recurrent Units (GRUs)

GRUs offer a computationally efficient alternative to LSTM networks for representing sequential dependencies in text. In the domain of fake and

authentic tweet classification, GRUs excel in recognising the temporal and syntactic patterns present in tweet sequences, enabling the model to differentiate subtle language variations between genuine and misleading content. GRUs employ gating mechanisms to dynamically manage the flow of information, allowing the network to retain or discard context as required, without the necessity for complex memory cells. The GRU architecture consists of two primary gates: the update gate z_t and the reset gate r_t. These gates govern the retention or exclusion of previous information, hence influencing the construction of the current hidden state. The mathematical expression is as follows:

- Update gate: $z_t = \sigma(W_z x_t + U_z h_{t-1} + b_z)$
- Reset gate: $r_t = \sigma(W_r x_t + U_r h_{t-1} + b_r)$
- Candidate hidden state: $\tilde{h}_t = \tanh(W_h x_t + U_h(r_t \odot h_{t-1}) + b_h$
- Final hidden state update: $h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t$

Here, σ signifies the sigmoid function, ⊙ indicates element-wise multiplication, and h_t encapsulates the enhanced hidden representation at time step t. This architecture allows GRUs to effectively encode contextual information across both short and long-term dependencies in twitter data, resulting in enhanced classification performance in differentiating between bogus and genuine tweets. [18]

v. Long Short-Term Memory (LSTM)

LSTM networks excel in analysing Twitter data due to its ability to model long-range dependencies and contextual nuances in sequential text. LSTM is an effective method for distinguishing between genuine and false tweets, as it can selectively retain pertinent information over time while discarding irrelevant or extraneous stuff typically seen in social media communications. The LSTM architecture utilises gating mechanisms to regulate the information flow within its memory cell. The candidate cell state $\tilde{c}_t$ is computed as

$$\tilde{c}_t = \tanh(W_c x_t + U_c h_{t-1} + b_c)$$

The input, forget, and output gates regulate the modification and visibility of the cell state.

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i), \quad f_t = \sigma(W_f x_t + U_f h_{t-1} + b_i), \; o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o)$$

The cell state c_t is updated by integrating the prior state $c_{t-1}$with the current candidate, regulated by the forget and input gates

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t$$

Ultimately, the concealed state h_t, which encapsulates the processed information for classification, is obtained as $h_t = o_t \odot tanh.$ This architecture allows the LSTM to proficiently discern temporal patterns and dependencies in tweets, enhancing the accuracy of distinguishing between bogus and genuine material in intricate, noisy Twitter datasets. [19]

vi. Bi-LSTM Attention Transformer (BiLAT)

Given an input sequence $x \in \mathbb{R}^{L\,x\,d}$ of length L with d dimensional embeddings, the model aims to predict $y_t \in \mathbb{R}^{L\,x\,2}$, representing the token-wise probability of authenticity or falsity. The attention span s is originally set at 32, while the sample pool size P is established at 256. In each epoch, the forward and backward LSTM hidden and cell states (h_f, c_f, h_b, c_b) are initialised and incrementally updated across the sequence length. The outputs of the bidirectional LSTM are merged into h_{bi}, which is employed to compute the initial attention weights. A hierarchical attention refinement is performed in multiple stages: when s ≥ 1, the span is halved, the relative positions of embeddings are adjusted, attention weights are recalibrated, and a cumulative distribution function (CDF) is generated from the attention scores. For K iterations (where K = 1.5'), tokens are sampled from the cumulative distribution function using uniform sampling throughout the pool size P, their local context is established, and indices are aggregated. Active embeddings are selected based on these indices and processed by multi-head attention to provide refined attention representations (neuAttr), which are employed to adjust the original attention weights. After optimising the attention mechanism, final token predictions are produced by a softmax

activation applied to the adjusted embedding at each position: $y_t = softmax(W . h_{bi}^t + b)$, resulting in class probabilities $[P(real), P(fake)]$.

Algorithm: Bi-directional Long Short-Term Memory Attention Transformer

Input: $x\,(L, d)$ ← Sequence of length L with d-dimensional embedding
Output: $y_t(L, 2)$ → Predictions for each token
Steps:
s ← Initial Attention Span;
P ← Sampling Pool Size
for each epoch,
Initialize the forward hidden state h_f and the backward hidden state h_b.
for t=1 to L do:
$h_f^t, c_f^t \leftarrow LSTM_{forward}\left(x_t, h_f^{t-1}, c_f^{t-1}\right)$
$h_b^t, c_b^t \rightarrow LSTM_{backward}\ (x_t, h_b^{t-1}, c_b^{t-1})$
end for
h_{bi} ← Concatenate $\left(h_f, h_b\right)$
$attr \leftarrow computeAttention\ Weights\ (h_{bi})$
t ← 0
While $s \geq 1$ do:
$s \rightarrow s\ /\ 2$
Adjust Relative Positions $(h_{bi}, s)\ h$
$attr \leftarrow rescaleAttention(attr)$
$cdf \leftarrow buildCumulativeDistribution(attr)$
$K \leftarrow [1.5']$
$for\ k \leftarrow 0\ to\ K\ do$:
$u \leftarrow uniform(0, P)$
$indices \leftarrow sample\ Tokens(cdf, u)$
$neighbours \leftarrow findLocalContext(indices, s)$
$indices \leftarrow merge(indices, neighbours)$
$active\_embeddings \leftarrow select(h\_bi, indices)$
$neuAttr \leftarrow multiHeadAttention(active\_embeddings)$
$attr \leftarrow updateWeights(attr, neuAttr, indices)$
end for
end While
for $t = 1\ to\ L$ do:
$y_t \leftarrow softmax(W . h_{bi}^t + b) \leftarrow [P(real), P(fake)]$
End for
End for

3.4 Hyper parameter optimization technique

Effective hyperparameter optimisation is crucial for improving the performance and generalisation capacities of deep learning models. This study employed four common methodologies: GS, RS, BO, and GA each with distinct characteristics for exploring the hyperparameter space.

i. Grid Search (GS) is a conventional and thorough hyperparameter optimisation technique that methodically explores a predefined grid of hyperparameter values to identify the optimal configuration that improves deep learning model performance. In deep neural networks (DNN), prevalent hyperparameters include learning rate, batch size, number of layers, number of neurones per layer, activation functions, and optimiser types. The total number of alternative configurations, given k hyperparameters each with n_i discrete values, is calculated by the Cartesian product. $|G| = \prod_{i=1}^{k} n_i$. Each configuration $\theta^{(j)} \epsilon\ G$ is assessed utilizing either a predetermined validation set or k-fold cross-validation. The mean performance throughout the k folds is calculated as

$$CV_{avg}^{j} = \left(\frac{1}{k}\right)\sum_{i=1}^{k} Score_i^j$$

The optimal hyperparameter configuration is thereafter established by maximising this average score.

$$\theta^* = \arg\max\left(\theta^{(j)} \in\ G\right) CV_{avg}^{(j)}$$

Despite its comprehensiveness, Grid Search has significant computational expense, particularly in deep learning, where model training can be protracted. The overall time complexity is expressed as $\mathcal{O}\left(\prod_{i=1}^{k} n_i\,.\,T_{train}\right)$, where T_{train} signifies the duration required to train and evaluate the model for a single configuration. As the number of hyperparameters and their value ranges increase, Grid Search becomes impractical for large-scale deep learning models. Its deterministic nature ensures consistency, making it an appropriate standard for evaluating more advanced optimisation tactics in experimental research. [20]

ii. Random Search (RS)

RS is a hyperparameter optimisation technique that randomly selects n configurations from the search space instead of exhaustively assessing all possibilities. In a deep learning model with k hyperparameters, each hyperparameter θ_i is sampled from either a continuous uniform distribution $U(a_i, b_i)$ or from a discrete set of possible values. Each sampled configuration $\theta^{(j)} = \left(\theta_1^{(j)}, \theta_2^{(j)}, \dots \dots, \theta_k^{(j)}\right)$ is utilised to train the model on Twitter data, producing predictions $\hat{y}_t\left(\theta^{(j)}\right)$ for tokens $t = 1, \dots, m$ in the validation set. The effectiveness of each configuration is evaluated using an average loss or measure, commonly the cross-entropy loss defined as

$$L\,(y_t, \quad \hat{y}_t) = -\sum_{c=1}^{C} \left(y_{t,c} \log(\hat{y}_t, c)\right)$$

Where, $C = 2$ for both genuine and counterfeit classes. The validation score for each configuration is expressed as

$$Score\left(\theta^{(j)}\right) = \frac{1}{m}\sum_{t=1}^{m} L\left(y_t, \hat{y}_{t(\theta^j)}\right)$$

The optimal hyper parameter set θ^* is identified by maximising this score among the sampled configurations:

$$\theta^* = \arg\max(\theta^j \in G)\, Score\ (\theta^j)$$

Random Search reduces computational complexity to $O(n\,.T_{train})$. Here, T_{train} represents the average training duration for each configuration. RS is notably effective and scalable for optimising deep learning models within vast hyperparameter spaces, exemplified by Twitter's real/fake categorisation tasks, where thorough Grid Search is computationally unfeasible. [21]

iii. Bayesian Optimization (BO)

BO is an efficient global optimisation technique that is particularly effective for hyperparameter tweaking in deep learning applications, such as classifying Twitter data as authentic or fraudulent. It denotes the unspecified objective function f(θ), which correlates a hyperparameter configuration $\theta \in R^k$ (e.g., learning rate, batch size, dropout rate) with a performance metric (e.g., accuracy or F1), and seeks to identify the optimal configuration.

$$\theta^* = \arg\max(\theta^j \in G)\, f(\theta)$$

Given the high cost of directly assessing $f(\theta)$ over complete training iterations, BO utilizes a probabilistic surrogate model, commonly a Gaussian Process (GP), to estimate f. The Gaussian Process delineates a distribution over functions and yields a posterior mean μ(θ) and uncertainty $\sigma(\theta)$ for each candidate θ, informed by prior observations $D_{1:t} = \left\{\left(\theta^i, f(\theta^i)\right)\right\}_{i=1}^{t}$. To determine the next evaluation point, Bayesian Optimization maximizes an acquisition function $\alpha(\theta)$, which reconciles exploration and exploitation. Common acquisition functions encompass Expected Improvement (EI), Upper Confidence Bound (UCB), and Probability of Improvement (PI). For instance, EI is articulated as

$$\begin{aligned}\alpha_{EI}(\theta) = \ & E[\max(f(\theta) - \ f, 0)]. \\ & = (\mu(\theta) - \ f(\theta^+) - \ \xi)\ \Phi(z) \\ & + \ \sigma(\theta)\emptyset(z)\end{aligned}$$

Where, $z = \frac{(\mu(\theta) - f(\theta^+) - \xi)}{\sigma(\theta)}$, with $f(\theta^+)$ representing the current optimal performance, and Φ and $\emptyset$ denoting the cumulative distribution function and probability density function of the standard normal distribution, respectively. The subsequent arrangement for assessment is

$$\theta_{next} = \arg\max(\theta \ \in \odot \ \alpha(\theta))$$

The model is updated incrementally as new data is obtained, making BO particularly suitable for enhancing high-cost deep learning models in natural language processing tasks, such as identifying bogus news on Twitter. Notwithstanding its cubic time complexity $O(n^3)$ and quadratic space complexity $O(n^2)$, BO is notably data-efficient, often requiring fewer evaluations than Random or Grid Search to achieve near-optimal configurations. [22]

iv. Grid Algorithm (GA)

The GA is a deterministic optimisation technique used to determine the ideal hyperparameter configuration for deep learning models. It generates a multidimensional grid from the Cartesian product of discrete candidate sets for each hyperparameter. Formally, for a model with k hyperparameters $\theta_1, \theta_2, \dots, \theta_k$, where each $\theta_i \in \Theta_i$and $|\Theta_i| = N_i$, the complete search space is defined as

$\mathcal{G} = \Theta_1 \, X \, \Theta_2 \, X \dots \dots X \Theta_k$ With size $|\mathcal{G}| \prod_{i=1}^{k} N_i$

Each $\theta^{(j)} \epsilon \, \mathcal{G}$ indicates a location within this search space. Each configuration entails the training and validation of the deep learning model. The objective of optimisation is to determine

$$\theta^* = arg \max_{\theta^j \in G} S(\theta^j)$$

S denotes a scoring metric, which may encompass validation accuracy, F1-score, or the reciprocal of validation loss. In classification tasks, a standard evaluation involves minimising categorical cross-entropy loss.

$$L(\theta^j) = \left(\frac{1}{m}\right) \sum_{t=1}^{m} \sum_{c=1}^{C} y_{t,c} \log\left(\hat{y}_{t,c}^{j}\right)$$

And picking θ^* that minimises this loss or enhances performance metrics. The Grid Algorithm, while comprehensive, is particularly efficient for low-dimensional hyperparameter spaces where full coverage is computationally feasible. However, its scalability is limited by the exponential rise in complexity $\mathcal{O}\left(\prod_{i=1}^{k} N_i\right)$, making it impractical for high-dimensional deep learning architectures that include variables such as learning rate, batch size, number of layers, dropout rates, and types of optimisers. [23]

v. BOHB (Bayesian Optimization with Hyper band)

The algorithm begins by initializing the best accuracy $a* \leftarrow -\infty$ and an empty observation set $D \leftarrow \emptyset$, then constructs a Gaussian Process model $GP(\mu_0, k)$ to predict accuracy from hyperparameters, for each bracket s from $s_{\max down}$ to 1, it calculates $n = \lceil (s+1) \cdot \eta^s \rceil$ configurations and allocates budget $B = \left\lfloor \frac{E}{s+1} \cdot \eta^s \right\rfloor$ epochs, samples hyper parameters $\{\theta_1, \dots, \theta_k\}$ using Expected Improvement, and for each θ, starts training with r = p epochs, progressively increasing to $r \leftarrow 3r$ while tracking validation accuracy a terminating early if GP predicts P(improvement) < δ then updates GP with new (θ, r, a) observations and maintains the best configuration θ^* with accuracy a^*, repeating this process until exhausting the total budget before returning the optimal θ* and a*, achieving efficient neural network hyperparameter optimisation through alternating Bayesian sampling guided by GP, Hyper band's geometric resource allocation, and continuous model refinement that optimises all parameters at once while automatically removing unpromising trials and scaling well with network complexity.

Algorithm: BOHB (Bayesian Optimization with Hyper band)

Input

- Neural network framework N,
- Hyper parameter space Λ (learning rate, batch size, layers, etc.).
- Minimum epochs p, maximum epochs E
- Threshold for early termination δ
- Training dataset (X_{train}, y_{train}), validation dataset (X_{val}, y_{val})

Output

- Optimal hyper parameters θ*
- Highest validation accuracy a*

Procedure

Set best $a* \leftarrow -\infty$ and initialize observed data $\mathcal{D} \leftarrow \emptyset$
Construct the Gaussian Process model $GP\ (\mu_0, k)$
With s ranging from $s_{maxdown} to\ 1$:
$n \leftarrow \lceil (s+1) \cdot \eta^s \rceil \triangleright$ Number of configurations
$B \leftarrow \lfloor E/(s+1) \cdot \eta^s \rfloor \triangleright$ Total budget per bracket $(\eta = 3)$
Sample n configurations $\{\theta_1, \dots, \theta_k\}$ utilizing the Expected Improvement of the GP
For each $\theta \in \{\theta_1, \dots, \theta_k\}$

$r \leftarrow p$ ▷ Preliminary epoch allocation
While $r \leq to\ E$
Train model $\mathcal{N}$ with parameter θ for r epochs.
Evaluate validation accuracy a
$\mathcal{D} \leftarrow \mathcal{D} \cup (\theta, r, a)$
if $(a > a^* | GP) < \delta$: break
$r \leftarrow \eta \cdot r$ ▷ Geometric budget augmentation
Revise GP in accordance with revised observations $\mathcal{D}$
If $max(a) > a^*$
$a^* \leftarrow max(a)$
$\theta* \rightarrow argmax(\mathcal{D})$
Return (θ, a)

IV. RESULTS AND DISCUSSION

This section evaluates the efficacy of six deep learning models on three benchmark datasets employing five hyperparameter optimisation methods. The assessment of Accuracy, Precision, Recall, and F1-Score reveals significant disparities among designs and tuning methodologies, with BERT and the suggested BiLAT regularly surpassing competitors, particularly under BOHB optimisation. The results underscore the most efficient model-optimization pairings for reliable and precise false news identification across various datasets. In the categorisation of legitimate versus fraudulent tweets on Twitter, performance metrics are essential for assessing a model's effectiveness in distinguishing between genuine and misleading material.

Table.5.Performance of Evaluation Metrics

Metric	Formula
Accuracy	$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$
Precision	$Precision = \frac{TP}{TP + FN}$
Recall	Recall= $\frac{\text{TP}}{\text{TP+FN}}$
F1 Score	F-score=2 X $\frac{\text{Precision X Recall}}{\text{Precision Recall}}$

These metrics provide a comprehensive evaluation of the categorisation models, assessing their effectiveness and reliability across many performance attributes.

Table.6.Hyperparameter Sets Across DL Models

Model	Hyperparameters	Set 1	Set 2	Set 3
CNN	- Number of Conv Layers	2	3	4
	- Filter Size	(3×3)	(5×5)	(3×3)
	- Activation Function	'ReLU'	'ReLU'	'LeakyReLU'
	- Optimizer	'Adam'	'SGD'	'RMSprop'
	- Dropout Rate	0.3	0.2	0.4
BERT	- Learning Rate	2e-5	3e-5	5e-5
	- Batch Size	16	32	64
	- Number of Epochs	3	4	5
	- Warmup Steps	100	200	500
	- Max Sequence Length	128	256	512
GRU	- Hidden Units	64	128	256
	- Number of Layers	1	2	3
	- Dropout Rate	0.2	0.3	0.5
	- Optimizer	'Adam'	'RMSprop'	'Nadam'
	- Learning Rate	0.001	0.0005	0.0001
LSTM	- Hidden Units	64	128	256
	- Number of Layers	1	2	3
	- Dropout Rate	0.2	0.4	0.5
	- Optimizer	'Adam'	'SGD'	'RMSprop'
	- Learning Rate	0.001	0.0005	0.0001
BiLAT	- Population Size	20	30	40
	- Loudness (A)	0.5	0.7	0.9

	- Pulse Rate (r)	0.5	0.8	0.9
	- Frequency Range	[0, 2]	[0, 1.5]	[0, 3]
	- Learning Rate	0.001	0.0005	0.0001

The table above delineates the essential hyperparameter configurations for the CNN, BERT, GRU, LSTM, and BiLAT models across three experimental setups aimed at enhancing performance in fake news detection.

Table.7. DL Model Performance with BOHB (Fake and Real News)

	Accuracy					Precision				
Model	GS	BO	RS	GA	BOHB	GS	BO	RS	GA	BOHB
RNN	88	89	87	88	92	87	88	86	87	91
CNN	90	91	89	90	94	89	90	88	89	93
BERT	94	95	93	94	97	93	94	92	93	96
GRU	89	90	88	89	92	88	89	87	88	91
LSTM	91	92	90	91	95	90	91	89	90	94
BiLAT	92	93	91	92	98	87	88	86	87	91
	Recall					F1-Score				
RNN	87	88	86	87	91	87	88	86	87	91
CNN	90	91	89	90	93	90	91	89	90	93
BERT	96	97	95	96	97	95	96	94	95	96
GRU	88	89	87	88	91	88	89	87	88	91
LSTM	90	91	89	90	94	90	91	89	90	94
BiLAT	92	93	91	92	96	92	93	91	92	95

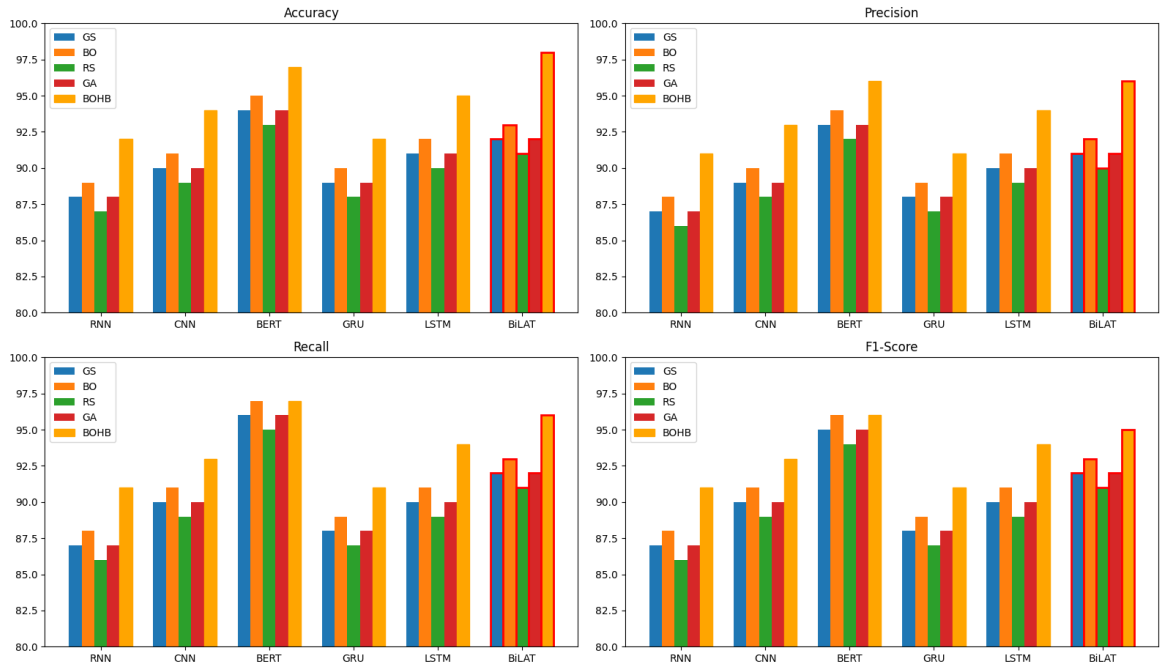

Fig.2. Comparative Metrics of DL Models Under BOHB (Fake and Real News)

© MAR 2026 | IRE Journals | Volume 9 Issue 9 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I9-1714856

The model performance table compares six deep learning models—RNN, CNN, BERT, GRU, LSTM, and BiLAT—across five optimisation techniques (GS, BO, RS, GA, BOHB) for four metrics: Accuracy, Precision, Recall, and F1-Score. BERT and BiLAT regularly attain the greatest rankings across all metrics. Under BOHB, BERT achieves 97% accuracy, 96% precision, 97% recall, and 96% F1-score, whereas BiLAT attains the maximum accuracy at 98%, along with robust precision (91%), recall (96%), and F1-score (95%). CNN and LSTM provide consistent mid-high performance, with BOHB accuracy rates of 94% and 95%, respectively. RNN and GRU exhibit slightly lesser performance, with BOHB accuracy ranging from 91% to 92%. BOHB consistently yields superior outcomes compared to GS, BO, RS, and GA across all models.

Table.8.DL Model Performance with BOHB (FakeNewsNet)

	Accuracy					Precision				
Model	GS	BO	RS	GA	BOHB	GS	BO	RS	GA	BOHB
RNN	89	90	88	89	93	88	89	87	88	92
CNN	91	92	90	91	95	90	91	89	90	94
BERT	95	96	94	95	98	94	95	93	94	97
GRU	90	91	89	90	94	89	90	88	89	93
LSTM	92	93	91	92	96	91	92	90	91	95
BiLAT	93	94	92	93	99	92	93	91	92	97
	Recall					F1-Score				
RNN	88	89	87	88	92	88	89	87	88	92
CNN	91	92	90	91	94	91	92	90	91	94
BERT	97	98	96	97	99	96	97	95	96	98
GRU	89	90	88	89	93	89	90	88	89	93
LSTM	91	92	90	91	95	91	92	90	91	94
BiLAT	93	94	92	93	99	93	94	92	93	97

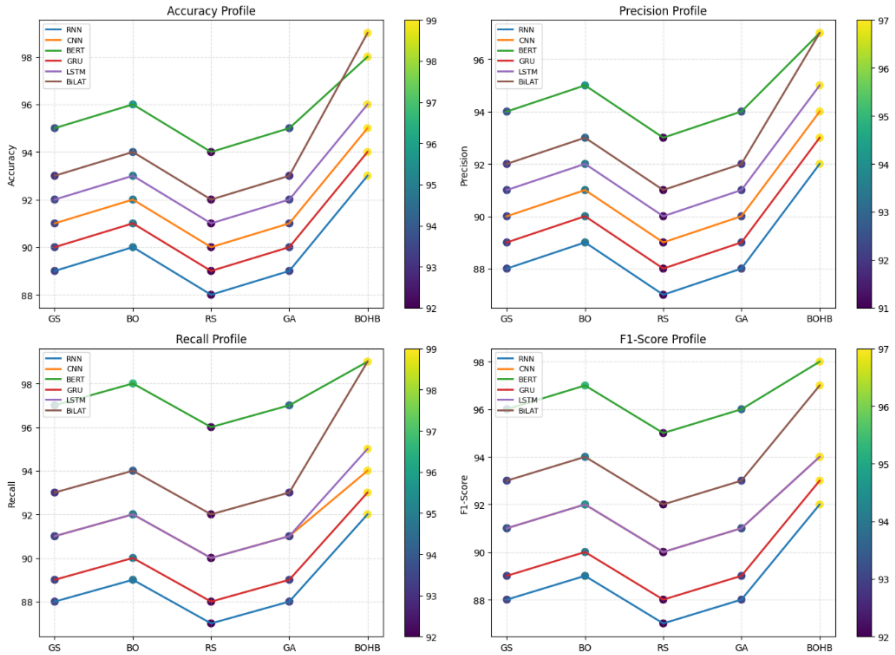

Fig.3. Comparative Metrics of DL Models Under BOHB (FakeNewsNet)

© MAR 2026 | IRE Journals | Volume 9 Issue 9 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I9-1714856

According to the table and figure, BiLAT (99), BERT (98), and LSTM (96) attain the highest BOHB accuracy values, succeeded by CNN (95), GRU (94), and RNN (93), indicating that BiLAT excels in BOHB optimisation. In terms of precision, BiLAT (97), BERT (97), and LSTM (95) excel under BOHB, whereas CNN (94), GRU (93), and RNN (92) exhibit marginally inferior performance. A comparable pattern is observed in recall, with BERT and BiLAT attaining the highest score of 99, succeeded by LSTM (95), CNN (94), GRU (93), and RNN (92). The F1-scores reflect this trend, with BiLAT (97), BERT (98), and LSTM (94) exhibiting robust performance under BOHB. In summary, BOHB optimisation markedly enhances all deep learning models, with BERT and BiLAT distinguished as the leading performers across all measures.

Table.9.ISOT Fabricated News Dataset (ISOT Fabricated News)

	Accuracy					Precision				
Model	GS	BO	RS	GA	BOHB	GS	BO	RS	GA	BOHB
RNN	87	88	86	87	92	86	87	85	86	90
CNN	89	90	88	89	94	88	89	87	88	93
BERT	93	94	92	93	97	92	93	91	92	96
GRU	88	89	87	88	93	87	88	86	87	92
LSTM	90	91	89	90	95	89	90	88	89	94
BiLAT	91	92	90	91	98	90	91	89	90	97
	Recall					F1-Score				
RNN	86	87	85	86	91	86	87	85	86	90
CNN	89	90	88	89	93	89	90	88	89	92
BERT	95	96	94	95	98	94	95	93	94	97
GRU	87	88	86	87	92	87	88	86	87	91
LSTM	89	90	88	89	94	89	90	88	89	93
BiLAT	90	91	89	90	97	90	91	89	90	96

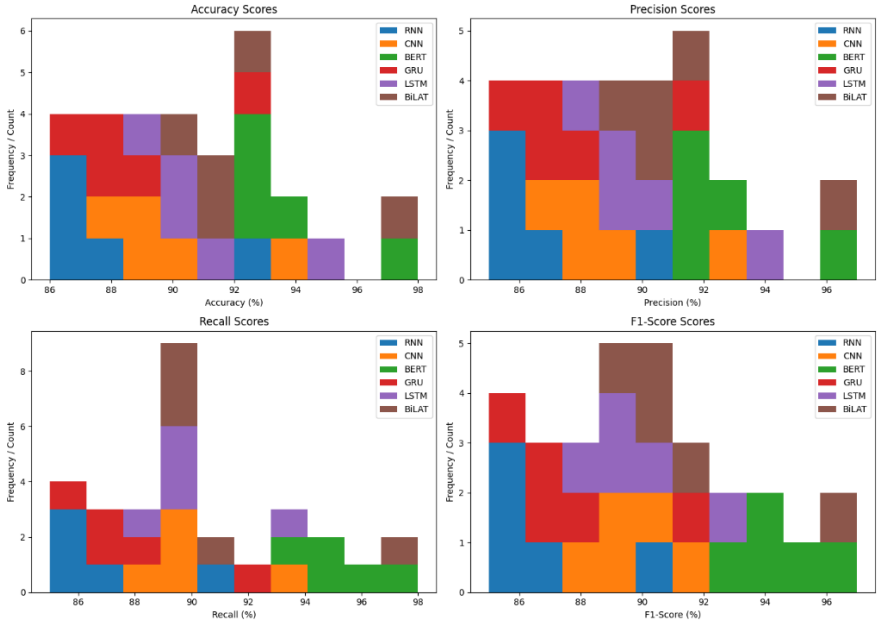

Fig.4. Model Performance on ISOT Fabricated News Dataset

In the ISOT Fabricated News Dataset, BiLAT and BERT consistently surpass other models across all metrics, with BiLAT attaining the highest BOHB accuracy (98%) and precision (97%), followed by BERT (97% accuracy, 96% precision) and LSTM (95% accuracy, 94% precision). CNN, GRU, and RNN are listed in descending order. A comparable pattern is noted in recollection and F1-score, with BiLAT (97% recall, 96% F1) and BERT (98% recall, 97% F1) in the forefront, indicating that BOHB optimisation markedly enhances model efficacy, with BiLAT distinguished as the overall superior performer.

V. CONCLUSION

This study provides a thorough assessment of deep learning models for detecting fake news on Twitter, demonstrating that transformer-based architectures, especially BERT and the proposed BiLAT, regularly surpass traditional RNN variations. BiLAT attained an accuracy of up to 99% on FakeNewsNet, illustrating the efficacy of integrating bidirectional sequence modelling with attention mechanisms. The findings underscore the essential importance of hyperparameter optimisation, with BOHB yielding consistent accuracy enhancements of 2–5% across all models and datasets. These findings highlight the significance of model architecture and optimisation method in attaining effective misinformation detection. The findings offer explicit direction for the design of high-performance detection systems; nonetheless, computing requirements and cross-platform generalisation continue to pose obstacles. Subsequent investigations ought to examine efficient model variants, ensemble methodologies, transfer learning, and the incorporation of contextual or temporal factors to improve flexibility and scalability in practical applications.

REFERENCES

[1] M. Q. Alnabhan and P. Branco, "Fake News Detection Using Deep Learning: A Systematic Literature Review," in *IEEE Access*, vol. 12, pp. 114435-114459, 2024

[2] Kumari, Shweta & Singh, Maheshwari. (2024). A Deep Learning Multimodal Framework for Fake News Detection. Engineering, Technology & Applied Science Research. 14. 16527-16533. 10.48084/etasr.8170.

[3] Vinita Nair, Dr. Jyoti Pareek, Sanskruti Bhatt, A Knowledge Based Deep Learning Approach for Automatic Fake News Detection using BERT on Twitter, Procedia Computer Science, Volume 235, 2024, Pages 1870 1882, ISSN 1877-0509.

[4] Yan, Y.; Fu, H.; Wu, F. Multimodal Social Media Fake News Detection Based on 1D-CCNet Attention Mechanism. Electronics 2024, 13, 3700.

[5] Zhu, P., Hua, J., Tang, K. et al. Multimodal fake news detection through intra-modality feature aggregation and inter-modality semantic fusion. Complex Intell. Syst. 10, 5851–5863 (2024).

[6] M, G.K., Faizz Ahmad, K., Pamidimukkala, S.G. et al. Hybrid optimization driven fake news detection using reinforced transformer models. Sci Rep 15, 14782 (2025).

[7] Sallah, E. A. Abdellaoui Alaoui, S. Agoujil, M. A. Wani, M. Hammad, Y. Maleh, and A. A. Abd El-Latif, "Fine Tuned Understanding: Enhancing Social Bot Detection With Transformer Based Classification," IEEE Access, vol. 12, pp. 118250–118269, 2024.

[8] Di Huang, Jinbao Song, Xingyu Zhang, Semi-Supervised Social Bot Detection with Relational Graph Attention Transformers and Characteristics of the social environment, Information Fusion, Volume 118, 2025, 102956, ISSN 1566-2535.

[9] S. Koca and I. Cicekli, "Fake Detection for Tweets and News with Different Domain Datasets," 2024 Innovations in Intelligent Systems and Applications Conference (ASYU), Ankara, Turkiye, 2024, pp. 1-6.

[10] E.Almandouh, M., Alrahmawy, M.F., Eisa, M. et al. Ensemble based high performance deep learning models for fake news detection. Sci Rep 14, 26591 (2024).

[11] Kayato Soga, Soh Yoshida, Mitsuji Muneyasu, Exploiting stance similarity and graph neural networks for fake news detection, Pattern Recognition Letters, Volume 177, 2024, Pages 26-32, ISSN 0167-8655.

[12] M. F. Mridha, A. J. Keya, M. A. Hamid, M. M. Monowar, and M. S. Rahman, "A comprehensive review on fake news detection with deep learning," *IEEE Access*, vol. 9, pp. 1, Nov. 2021.

[13] R. Budiarto, H. Setiawan, and R. N. Yasa, "Text preprocessing for optimal accuracy in Indonesian sentiment analysis using a deep learning model with word embedding," *AIP Conference Proceedings,* vol. 2601, Aug. 2021..

[14] Ahmed, S.F., Alam, M.S.B., Hassan, M. et al. Deep learning modelling techniques: current progress, applications, advantages, and challenges. Artif Intell Rev 56, 13521–13617 (2023).

[15] M. Massaoudi, H. Abu-Rub, S. S. Refaat, I. Chihi and F. S. Oueslati, "Deep Learning in Smart Grid Technology: A Review of Recent Advancements and Future Prospects," in *IEEE Access*, vol. 9, pp. 54558-54578, 2021.

[16] Nasir, J. A., Khan, O. S., & Varlamis, I., "Fake news detection: A hybrid CNN-RNN based deep learning approach," International Journal of Information Management Data Insights, vol. 1, no. 1, Article 100007, Apr. 2021.

[17] Al-alshaqi, M.; Rawat, D.B.; Liu, C. A BERT-Based Multimodal Framework for Enhanced Fake News Detection Using Text and Image Data Fusion. Computers 2025, 14,237.

[18] T. Bhatia, B. Manaskasemsak, and A. Rungsawang, "Detecting fake news sources on Twitter using deep neural network," in Proc. 2023 11th International Conference on Information and Education Technology (ICIET), Mar. 2023.

[19] S. Khan and E. Guzmán, “A hybrid LSTM-Transformer framework for accurate fake news detection on Twitter,” available at SSRN, SSRN: 5674494, 2023.

[20] .M. A. K. Raiaan, S. Sakib, N. M. Fahad, A. A. Mamun, M. A. Rahman, S. Shatabda and M. S. H. Mukta, “A systematic review of hyperparameter optimization techniques in Convolutional Neural Networks,” Decision Analytics Journal, vol. 11, Art. no. 100470, 2024.

[21] Jaber, Y.; Dharmasena, P.; Nassif, A.; Nassif, N. Hyperparameter Optimization of Neural Networks Using Grid Search for Predicting HVAC Heating Coil Performance. Buildings 2025, 15, 2753.

[22] Zulfiqar, M.; Gamage, K.A.A.; Kamran, M.; Rasheed, M.B. Hyperparameter Optimization of Bayesian Neural Network Using Bayesian Optimization and Intelligent Feature Engineering for Load Forecasting. Sensors 2022, 22, 4446.

[23] M. Massaoudi, H. Abu-Rub, S. S. Refaat, I. Chihi and F. S. Oueslati, "Deep Learning in Smart Grid Technology: A Review of Recent Advancements and Future Prospects," in IEEE Access, vol. 9, pp. 54558-54578, 2021.