Acta Scientiarum
http://periodicos.uem.br/ojs/acta
ISSN on-line: 1807-8664
Doi: 10.4025/actascitechnol.v47i1.73210

COMPUTER SCIENCE

Detection of fake news in social media using CNN with grey wolf optimized BERT

Rajalakshmi Viswanathan*, **Sharon Femi Paul Sunder Nathaniel, Kala Alwarsamy and Khanaghavalle Rajendran**

Sri Venkateswara College of Engineering, Pennalur, Sriperumbudur 602117, Tamil Nadu, India. *Author for correspondence: E-mail: vraji@svce.ac.in

ABSTRACT. Fake news refers to stories that are falsely represented as news, that has the potential to misinform and mislead readers and identifying such fake news is a significant challenge to combat its impact in society. Our proposed work detects fake news using a hybrid method for fake news detection that uses Bidirectional Encoder Representations from Transformers (BERT) framework for feature extraction, Convolutional Neural Networks (CNN) for classification and Grey Wolf optimization for optimizing the parameters of BERT. The raw data is preprocessed and transformed into a format that can be fed as an input to the BERT model for feature extraction. This pre-trained model is fine-tuned using Grey wolf optimization (GWO) and CNN is used to classify each news article as real or fake. The proposed BERT-GWO-CNN model outperforms the existing machine learning models like Random Forest, SVM and deep learning namely RNN and LSTM in terms of prediction accuracy.

Keywords: BERT; N-gram; skip gram; grey wolf optimization; convolutional neural network.

Received on July 21, 2024.
Accepted on January 23, 2025.

Introduction

In today's era, the Internet is driven by news and advertising, with websites leveraging high traffic from hot news and provocative headlines to generate revenue through automated advertising. Also, social media platforms have gained significant popularity due to their ability to facilitate easy access to information and quick sharing of information. But this has also led to the spread of false news which causes harmful effects such as inciting conflicts, interfering with voting processes, and cultivating societal animosity (Lazer et al., 2018). Hence, it is more important to recognize fake news by devising new techniques or models (Zhang et al., 2019).

Fake news can be either based on news content or social context. To detect the news based fake news, style-based techniques (Fazil & Abulaish, 2018; Ruchansky et al., 2017) are used for capturing the writing style of manipulators. Strategies for social context (Ghosh & Shah, 2018; Shu et al., 2019) focus on the information flow between users and news articles. Social interactions can be a crucial factor in identifying fake news. In these methodologies, instance-based techniques analyze user behavior towards social media posts to evaluate the credibility of specific news stories.

Fake news detection is a complex and constantly evolving problem, and there is no single 'silver bullet' solution that will work in all cases. Our contribution is utilizing BERT framework for detecting fake news. BERT is a pre-trained deep learning model that can be used for detecting fake news. It uses transformer-based architecture that allows it to capture complex patterns and relationships in text data (Devlin et al., 2018). Its ability to grasp the contextual meaning within a sentence or text signifies that a word can be shaped by the words in its vicinity. Consequently, considering the context of a word results in a unique numerical value for that word each time. This stands in contrast to static methods, where unique words are encoded without regard for their context. In NLP approaches based on static word embeddings, a given word is consistently encoded in the same manner (Kula et al., 2022).

Convolutional Neural Networks (CNNs) has robustness to variations, parallel processing ability, and performance that makes it to play a pivotal role in advancing machine learning applications across diverse fields (Rajalakshmi & Ganesh Vaidyanathan, 2022b). Hence, in the proposed method CNNs are employed for the classification of fake news. Convolutional layers in CNNs perform convolutions across the input text, extracting features through filters of varying sizes. This enables the model to capture both short and long-range dependencies in the textual data, making it well-suited for tasks like fake news classification. Pooling layers, such as max pooling, are often used to down-sample the spatial dimensions of the extracted features. This reduces the computational load and focuses on the most informative elements, helping the model generalize better to varying inputs. The final layers of the CNN are designed for classification, mapping the extracted features to the output classes (Saleh et al., 2021).

Bio-inspired algorithms play a crucial role across diverse domains, contributing to optimization efforts (Femi et al., 2021; Femi & Vaidyanathan, 2022; Kala & Vaidyanathan, 2022; Rajalakshmi & Ganesh Vaidyanathan, 2022a). Optimization algorithms are used for improving the performance. Smith (Smith, 2018) clearly explains parameter tuning of neural network parameters like learning rate, batch size, momentum. Hence, the proposed method performs parameter tuning to optimize the parameters of BERT using bio-inspired algorithm for achieving the better performance, ensuring efficient use of computational resources. The main contributions of this paper include

- To implement BERT framework for feature extraction and optimize its parameters using Grey Wolf optimization
- To classify the fake news articles using CNN.
- To improve the performance of fake news detection.

The structure of the manuscript is arranged as: Section 2 reviews on the existing literature related to this work. Sections 3, 4 and 5 elaborate on the BERT, Grey Wolf Optimization and CNN algorithm, respectively. Section 6 investigates the proposed hybrid BERT-GWO-CNN model for fake news detection. Experimental results are presented in Section 7, and Chapter 8 serves as the conclusion for this work.

Literature survey

Numerous advancements in machine learning and deep learning have been applied to fake news detection, each demonstrating distinct strengths and limitations. A thorough evaluation of these approaches underscores the need for the BERT-GWO-CNN model and its potential to address existing challenges effectively.

Jehad et al. (2020) utilized Random Forest and Decision Tree algorithms (Jehad & Yousif, 2020) alongside TF-IDF for feature extraction. While these models achieved respectable accuracies (89.11% for Decision Tree and 84.97% for Random Forest), they depend heavily on static feature engineering. Although TF-IDF effectively captures term frequencies, it overlooks contextual word relationships, limiting the models' capacity to grasp subtle linguistic nuances characteristic of fake news. Despite their lower computational requirements, these approaches struggle to manage more complex datasets.

Pandey et al. (2022), implemented various classifiers, including K-NN, SVM, Decision Tree, Naïve Bayes, and Logistic Regression, incorporating preprocessing techniques to address class imbalances. Sudhakar et al (M & Kaliyamurthie, 2023) applied Logistic Regression performed particularly well in political fake news detection, achieving 98% accuracy. However, traditional classifiers lack scalability for large datasets or unstructured data, as they cannot capture deep semantic features essential for complex text analysis.

Goldani et al. (2021) introduced a CNN-based architecture (Goldani et al., 2021) incorporating margin loss and various embedding techniques. Their analysis highlighted the adaptability of non-static embedding, which can be incrementally updated during training. However, CNNs, while effective for feature extraction, are limited in capturing long-range dependencies in text, which are essential for identifying nuanced patterns in fake news.

Yang et al. (2018) enhanced CNN-based fake news detection with the TI-CNN model (Yang et al., 2018), which integrates textual and visual features. Although this hybrid approach benefits from using multiple data modalities, it is computationally intensive and may encounter difficulties when image data is unavailable or irrelevant.

Recurrent models, such as those developed by Bahad et al. (2019), Mahara et al. (2022), and Sastrawan et al. (2022), employed LSTM and Bi-LSTM architectures. These models effectively capture sequential dependencies and demonstrated strong performance when paired with GloVe embedding. However, their reliance on sequential processing makes them computationally demanding, and they may face challenges with vanishing gradients in lengthy text sequences.

Kaliyar et al. (2020, 2021) proposed FNDNet and EchoFakeD (Kaliyar et al., 2020), which integrate content and social context using coupled matrix-tensor factorization and deep neural networks. While these approaches effectively utilize diverse data sources, their complexity limits their practicality for real-time applications (Kaliyar et al., 2021).

Sastrawan et al. (2022) applied various deep learning approaches in conjunction with pre-trained word embedding. They trained these models on distinct datasets and used a data augmentation technique called back-translation to address class imbalances in the data. Their findings indicated that the Bidirectional LSTM performed superior to both CNN and ResNet on all tested datasets.

Mahara et al (Mahara & Gangele, 2022) utilized LSTM and Bi-LSTM for identifying fake news. They first employed the NLTK toolkit to tokenize and preprocess the data. Next, they incorporated GLOVE word embedding to leverage the high-level characteristics of the input text extracted by the RNN-LSTM and Bi-LSTM models. Additionally, the proposed model used Dense layers and dropout technology to enhance its effectiveness. Balshetwar et al (Balshetwar et al., 2023) developed an approach for detecting fake news that takes into account sentiment as a significant feature to enhance accuracy. In their investigation, two separate datasets, ISOT and LIAR, were employed, and sentiment analysis was applied to identify pivotal feature words along with their propensity scores for opinions. A lexicon-based scoring algorithm was utilized for this purpose.

Ozbay and Alatas (2021) emphasized the effectiveness of metaheuristic optimization methods, such as Salp Swarm Optimization (SSO) (Ozbay & Alatas, 2021), in enhancing model performance. Their study showcased the ability of optimization techniques to improve parameter selection. However, the reliance on SSO's specific heuristics restricts its flexibility and adaptability to a wide range of datasets and tasks.

This hybrid architecture integrates the advantages of contextual and structural feature extraction with metaheuristic optimization, offering a highly effective solution for fake news detection.

BERT architecture

BERT uses the transformer encoder architecture for processing every token of text. It is a pretrained model which works with the concept of utilizing the bidirectional nature of stacked encoder (Devlin et al., 2018). The stacked encoder scans the text in both directions. The sequence of words (tokens) are expected as input by the BERT. It also expects two special tokens [CLS] and [SEP] as input. The [CLS] is the classification token which is expected to be the first token of the input sequence. The [SEP] is the token appended at the end of the sequence which is mainly used for the prediction task.

The first step in the implementation of the BERT model was to divide input sequences into tokens which is known as tokenization. After tokenization, the [CLS] and [SEP] tokens were appended at both the ends of the input sequence. This is carried out by the BertTokenizer. For a BERT model, the input size of the sequence should be exactly 512. If the length of the sequence is less than 512, padding was performed and at the same time if the length is more than 512, truncation was carried out.

During BERT model training, to interpret the context, BERT refers to the text that comes right before it. In order to determine the true meaning of words, it also examines the contextual connections between words within the sentence. The given sentence will subsequently be transformed into an embedding vector by BERT. The embedding vector contains the distinctive words in a given document. BERT guarantees that words with the same meaning will be represented in a similar fashion. Usually machine learning algorithms work better with numbers rather than with text. This is the major reason why the input text was converted to embedding vectors which is required for the efficient working of the machine learning models. In this work, GloVe embedding was used to capture the local and global connections between the words. Using GloVe, the co-occurance matrix between every pair of words was built. N-gram and Skip-gram techniques are used for efficient feature extraction.

The integration of GloVe and BERT embeddings combines the advantages of static and contextual representations. GloVe embedding capture word-level co-occurrence patterns across the entire corpus, offering a global semantic perspective. Meanwhile, BERT produces context-sensitive representations that reflect the relationships between words within a sentence. This hybrid approach enriches the model's ability to detect fake news by blending global semantic insights with local contextual nuances.

The BERT-base-uncased model was employed alongside a feed-forward network featuring hidden sizes of 768. The classifier was constructed from the ground up, incorporating a convolutional layer with a size of 128. Batch normalization was applied to standardize the inputs, followed by the addition of a dropout layer with a 0.6 rate to prevent overfitting. The model's output was a vector with a size equivalent to the number of classes in the classification task. To classify input news as either fake or real, two flattening layers with an output size of 2 were added.

Grey Wolf Optimization (GWO) for BERT parameter tuning

Grey Wolf Optimization crafted by Seyedali Mirjalili, draws inspiration from the societal structure and hunting tactics of grey wolves in nature. This optimization algorithm (Mirjalili et al., 2014) mirrors the leadership patterns and hunting strategies observed in grey wolves, using these insights to tackle optimization challenges (Mirjalili et al, 2014).

This section outlines the mechanics and application of GWO to derive an optimized solution. Beginning with a random initialization of solution populations, likened to a pack of wolves in the search space, the algorithm identifies the top-tier solutions: alpha, beta, delta, and omega. Among these, alpha stands as the foremost solution, succeeded by beta, gamma, and delta. Through a hunting process mirroring the encircling behavior of wolves, the algorithm iteratively updates the positions of these key solutions, aiming to track down the optimal solution. Subsequently, the fitness function is utilized to assess and refine the new positions, perpetuating the hunting and updating cycle until a specified number of iterations is accomplished. Figure 1 shows the implementation of GWO.

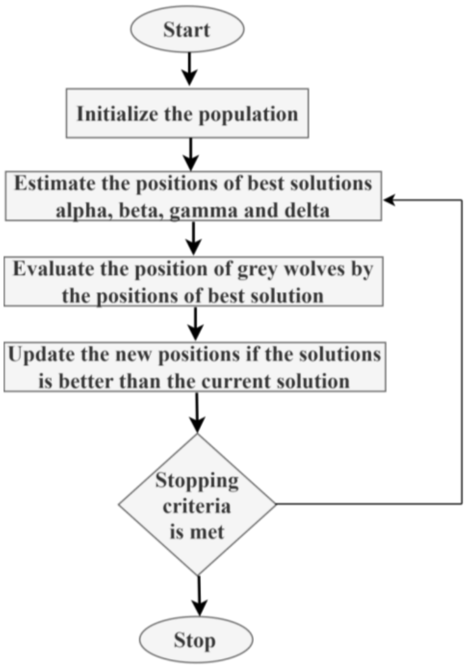

Figure 1. Grey Wolf Optimization.

Basic Mathematical formulation of BERT-GWO

The objective of the work is to enhance the prediction and to minimize the error in predictions. Hence our problem is a minimization problem with the fitness function f(x), where x is the solution or position of the wolf in the entire search space.

- Initialisation:
  - Population of N solutions are initialized in the search space
  - The position of each solution is represented by xi
- Formulation of solution hierarchy:
  - The objective function f(xi) is evaluated to determine the fitness of each solution
  - Select four best solutions based on their fitness values: alpha, beta, gramma and delta.
- Update the position of selected solutions:
  - The following formulae are used to calculate the new positions of the solutions
  - For alpha solution x_{alpha}:

$$x_{Nalpha} = x_{alpha} - \alpha \times distance(x_i, x_{alpha})$$

  - For alpha solution x_{beta}:

$x_{Nbeta} = x_{beta} - \beta \times distance(x_i, x_{beta})$

- For alpha solution x_{gamma}:

$x_{Ngamma} = x_{gamma} - \gamma \times distance(x_i, x_{gamma})$

- For alpha solution x_{delta}:

$x_{Ndelta} = x_{delta} - \omega \times distance(x_i, x_{delta})$

Where,
$\alpha, \beta, \gamma, \omega$ are values that are predefined.
$distance(x_i, x_j)$ is the distance between the two solutions x_i, x_j.

- Fitness computation and update:
  - Evaluate the fitness with the new positions of the solutions.
  - Update the positions if the new solutions is better than the previous solution

$x_{alpha} = x_{Nalpha}$

$x_{beta} = x_{Nbeta}$

$x_{gamma} = x_{Ngamma}$

$x_{delta} = x_{Ndelta}$

- Termination Condition:
  - The process is repeated for the predefined number of iterations
- BERT Parameter tuning:
  - The best solution from the GWO is assigned as weights and bias values for the BERT architecture. The features are extracted using BERT and the extracted features are given to CNN model for classification.

CNN for text classification

The input sample is a sequence of words represented as $x = x_1, x_2, x_3 \ldots\ldots, x_N$ where N is the length of the sequence. Each word x_i in the sequence is represented as a high-dimensional vector through word embedding. Let E be the embedding matrix, then the input sequence is represented as a matrix X of dimensions $n \times d$, where d is the embedding dimension. The convolution filters are applied to the input matrix X to extract the features. The number of words to be considered at a time is set as the filter size. There are k filters of size $h \times d$, where h is the filter size. Consider an input matrix X and a filter W_i with bias b_i, the convolution operation z_i for the i^th filter is computed as follows:

$z_i = ReLU(W_i * X + b_i)$

In the above equation, * is the convolution operator, W_i is the kernel/filter and b_i is the bias term. After convolution, max- pooling operation is applied for dimensionality reduction and capture the most important features from the convolved features. Followed by the pooling layer, the dense layer is formed with ReLU activation functions is computed as follows:

$h = ReLU(W_{dense} . flatten + b_{dense})$

The output from the dense layer is flattened into a vector and fed into one or more fully connected layers, followed by an output layer with softmax activation for classification. The output is computed as follows:

$$\hat{y} = Softmax(W_{out}.h + b_{out})$$

The CNN is trained using a binary cross-entropy loss function, and optimization techniques like stochastic gradient descent (SGD) or Adam are employed to minimize the loss. The binary cross entropy (BCE) is computed as follows:

$$BCE = -\frac{1}{m}\sum_{i}^{m}\sum_{j}^{c} y_{ij} log(\sigma(\widehat{y_{ij}})) - (1 - y_{ij}) log(1 - \sigma(\widehat{y_{ij}})))$$

Where i is the training instance index, j is the class label index (0 for Fake and 1 for Real), $\widehat{y_{ij}}$ is the output of the fully connected layer and y_{ij} is the actual output.

Proposed BERT-GWO-CNN model for fake news detection

In this section, a novel approach has been proposed in order to combine the BERT, a powerful language representation model with an evolutionary algorithm, Grey Wolf Optimization (GWO) in order to detect fake news. As BERT has the capability to comprehend the context of languages, it can be used to extract the appropriate features for fake news detection. It can analyze the semantics and relationships between words in a sentence, which is crucial in understanding the nuances of fake news. On the other hand, Grey Wolf Optimization is used for fine-tuning the parameters of the BERT model or optimize some aspects of the fake news detection process. GWO is known for its capability to optimize complex problems efficiently by mimicking the leadership hierarchy and hunting behavior of grey wolves. Finally, the CNN architecture analyzes the textual information and makes predictions about the authenticity of news articles or text passages. Figure 2 shows the proposed BERT-GWO-CNN architecture.

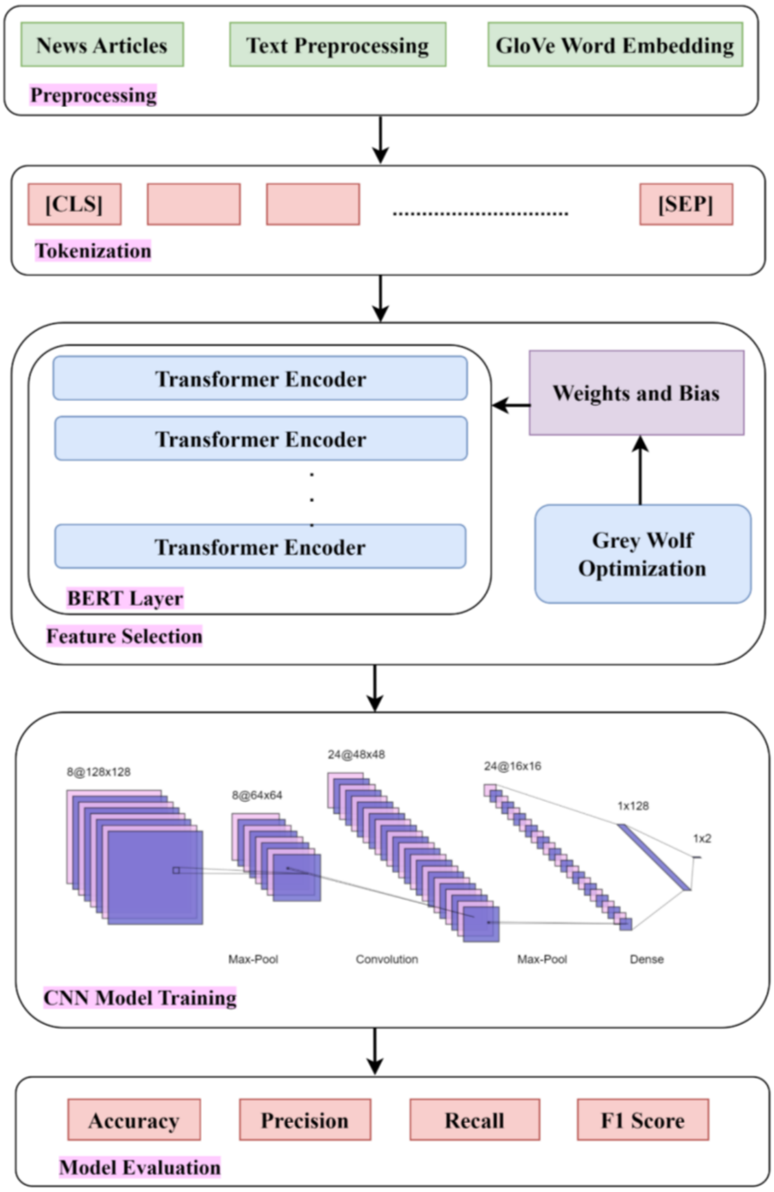

Figure 2. Proposed BERT-GWO-CNN Architecture for fake news detection.

The flow of the proposed BERT-GWO-CNN is detailed in this section. Initially the data set is cleaned, tokenized and encoded the news articles. The pre-trained BERT is utilized to extract contextual embeddings for the news articles. These embeddings capture the semantic meaning of words in the context of the entire article. Later, using the BERT embeddings, relevant features are extracted that can help classify fake news from real news. The feature extraction involves sentiment analysis, word frequency, or contextual information captured by BERT.

The weights and bias for the BERT model is optimized by applying Grey Wolf Optimization. This process involves tuning parameters of the BERT model, optimizing the feature selection process and thereby improving the classification accuracy. The feature extraction is followed by model building using Convolutional Neural Network. The CNN model building involves the following steps:

1. Embedding Layer: This layer is modeled to map words to dense vectors.
2. Convolutional Layers: In this layer, multiple convolutional layers are applied with different filter sizes to capture various text features.
3. Pooling Layers: In the pooling layers, the max-pooling operation is utilized to down sample the output of convolutional layers.
4. Flatten and Fully Connected Layers: In this layer, the output is flattened and connected to one or more fully connected layers for classification.
5. Output Layer: A softmax layer for binary classification to produce the output for fake news detection.

Finally, the model's performance is evaluated using metrics like accuracy, precision, recall, and F1-score on a test dataset. The process is repeated until good accuracy is obtained. The synergy between BERT's language understanding capabilities and GWO's optimization techniques can potentially enhance the accuracy and efficiency of fake news detection systems. However, implementing this combination would require careful tuning of parameters, rigorous testing, and validation to ensure the effectiveness of the approach.

In overall, this novel approach integrates Grey Wolf Optimization (GWO) to fine-tune BERT’s parameters, enhancing its ability to extract rich and meaningful embeddings for fake news detection. GWO’s bio-inspired mechanism balances exploration and exploitation, enabling an efficient search for optimal parameters. By combining BERT’s contextual embeddings, which capture nuanced word relationships, with CNN’s capability to identify critical textual patterns, the model leverages the strengths of both architectures. This hybrid design results in a robust classification system that outperforms existing models, achieving superior accuracy, precision, recall, and F1-scores through optimized embeddings and powerful classification layers.

Results and discussions

Dataset description

The proposed model is trained using the fake news dataset from the Information Security and Object Technology (ISOT) Research Lab and Kaggle Fake News dataset. Both the ISOT and Kaggle dataset contain 2 classes namely Fake and Real. The dataset is split in the ratio of 75:25 for training and testing of the model. Table 1. describes the volume of data in all the three datasets used for model training and testing.

Table 1. Volume of Data.

Data	ISOT (No. of Articles)		Kaggle (No. of Articles)	
	Fake News	Real News	Fake News	Real News
Total	23481	21417	290	290
Training	17610	16063	218	218
Testing	5871	5354	72	72

Model Parameters and Evaluation Metrics

The training data is fed into the BERT-GWO-CNN model for training. The GPU was enabled during model training as the BERT base model had millions of parameters. The number of epochs was fixed to be 50 with a learning rate of 1e-5 for ISOT dataset and the parameters for model training using Kaggle dataset was fixed to be 40 and learning rate as 1e-5. The cross entropy function is used as loss function for categorical outputs. The softmax activation function is used to classify the binary output as real or fake news. Figure 3 shows the cross entropy loss obtained from the BERT-GWO-CNN model training using ISOT dataset. From the graph, the loss is reduced during 50 epochs.

GridSearchCV is used to tune the learning rate for a CNN by testing a range of predefined learning rate values. CNN is trained for each learning rate and evaluates the model's performance using cross-validation. Finally, the optimal learning rate is obtained. The learning rate values are specified in a parameter grid, and GridSearchCV automatically tries each value. The best learning rate is selected based on the highest cross-validation accuracy. This process helps optimize the CNN’s training efficiency and model performance.

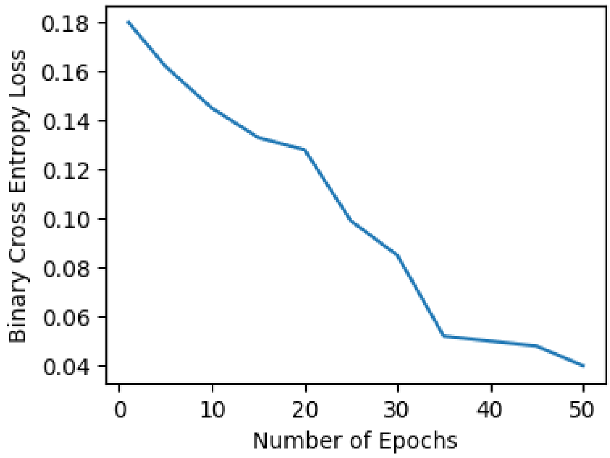

Figure 3. Cross Entropy loss for BERT-GWO-CNN model training using ISOT dataset.

To show the efficacy of the proposed BERT-GWO-CNN model, both the CNN and CNN integrated with BERT frameworks are trained and tested using identical test data. Figure 3 and Figure 4 presents the confusion matrices for these three models for the ISOT and Kaggle datasets respectively.

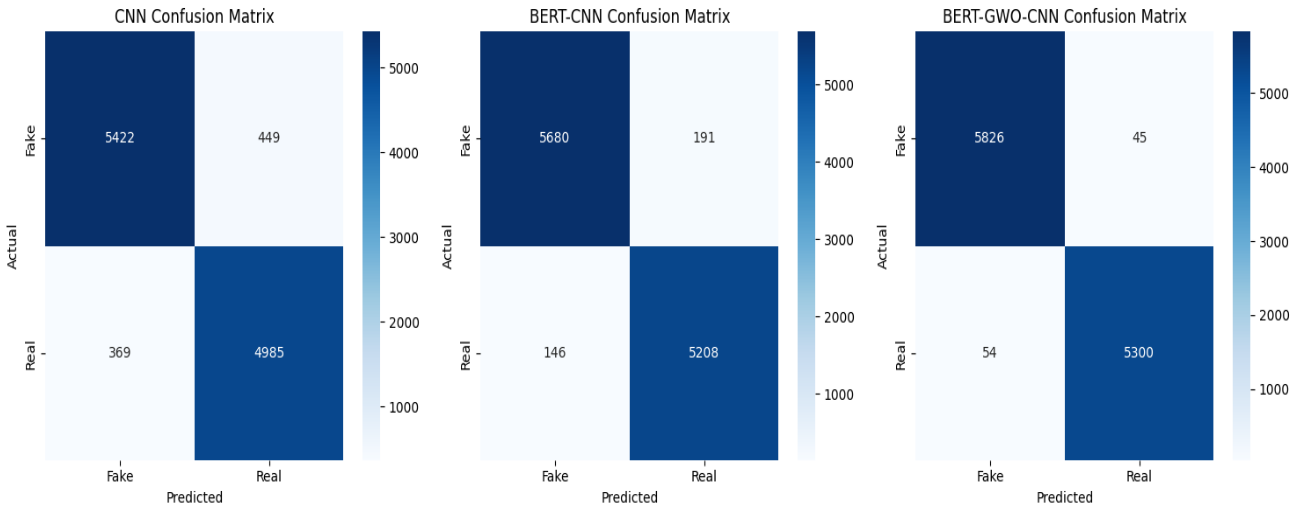

Figure 3. Confusion matrix for various models using ISOT Dataset.

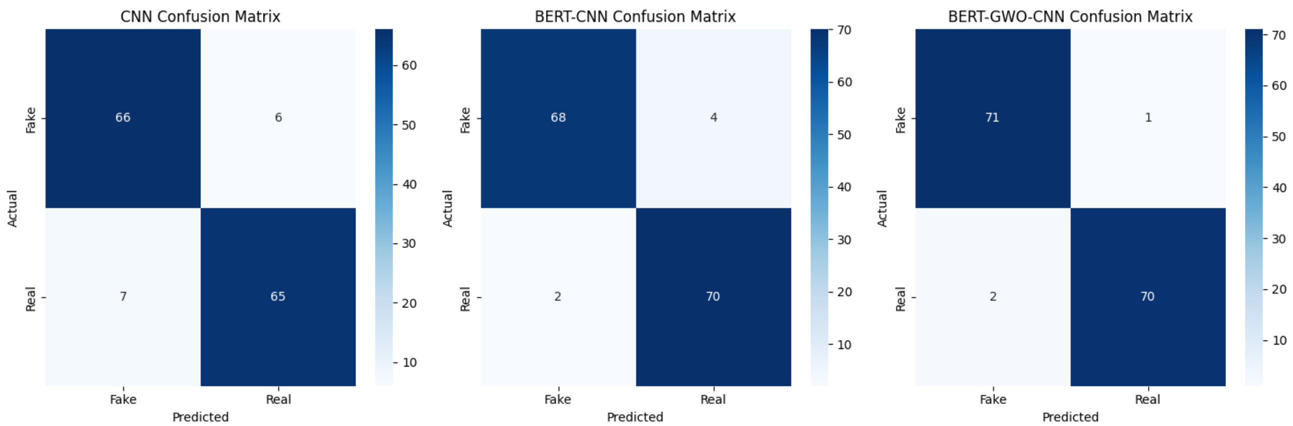

Figure 4. Confusion matrix for various models using Kaggle Dataset.

Evaluations metrics like accuracy, precision, recall and F1 Score are determined for all the three models using ISOT dataset and Kaggle dataset and their results are presented in Table 2 and Table 3 respectively. The inference is that the BERT-GWO-CNN model outperforms other models with an accuracy of 99.11%, precision of 99.08%, recall of 99.23% and with the best F1 Score of 99.15% using ISOT dataset and with an accuracy of 97.92%, precision of 97.26%, recall of 98.61% and with the F1 Score of 97.93% using ISOT dataset.

Table 2. Performance metrics of proposed BERT-GWO-CNN model and other deep learning models using ISOT dataset.

Models/ Metrics	NN	ResNet50	CNN	BERT-CNN	Proposed BERT-GWO-CNN
Accuracy	85.98	91.28	92.71	97	99.11
Precision	82.45	92.88	93.63	97.49	99.08
Recall	83.33	90.57	92.35	96.75	99.23
F1 Score	85.22	90.79	92.99	97.12	99.15

Table 3. Performance metrics of proposed BERT-GWO-CNN model and other deep learning models using Kaggle dataset.

Models/ Metrics	NN	ResNet50	CNN	BERT-CNN	Proposed BERT-GWO-CNN
Accuracy	83.98	88.28	90.97	95.83	97.92
Precision	82.64	87.88	90.41096	97.14286	97.26
Recall	83.72	88.57	91.66667	94.44444	98.61
F1 Score	83.57	88.79	91.03448	95.77465	97.93

Table 4 shows the comparison between the proposed models with the various state-of-art models. From the results presented in Table 4, BERT-GWO-CNN performs better than the NN, ResNet50, CNN and BERT-CNN models using ISOT dataset. Results clearly prove the efficiency of the proposed BERT-GWO-CNN model in fake news detection. The accuracy was around 99% for the proposed architecture which is higher than the traditional models in detecting the fake news.

Table 4. Comparison of proposed BERT-GWO-CNN with state-of-art models.

Models	Accuracy
Decision Tree	89.11%
Random Forest	84.97%
Support Vector Machine	89.33%
RNN	94.38%
Long Short Term Memory (LSTM)	96.35%
Bidirectional LSTM	94%
CNN	92.71%
BERT-CNN	96.99%
Proposed BERT-GWO-CNN	99.11%

Conclusion

Optimizing BERT's weights using metaheuristic algorithms like GWO involves considerable experimentation, careful tuning of parameters, and validation to ensure effectiveness and efficiency in improving BERT's performance in detecting fake news. BERT is a powerful pre-trained model that effectively captures the contextual meaning of text and recognize patterns in language. In this work, by fine-tuning BERT's weight and bias parameters with GWO on a dataset of labeled fake and real news articles, the features are extracted effectively from data. Then the trained CNN model accurately classifies the news articles as fake or real. The proposed BERT-GWO-CNN model outperforms other models in fake news detection tasks, achieving high accuracy scores of 99.11% for ISOT dataset and 97.93% for Kaggle dataset.

References

Balshetwar, S. V., Rs, A., & R, D. J. (2023). Fake news detection in social media based on sentiment analysis using classifier techniques. *Multimedia Tools and Applications*, *82*(23), 35781–35811. https://doi.org/10.1007/s11042-023-14883-3

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding* (Version 2). arXiv. https://doi.org/10.48550/ARXIV.1810.04805

Fazil, M., & Abulaish, M. (2018). A hybrid approach for detecting automated spammers in Twitter. *IEEE Transactions on Information Forensics and Security*, *13*(11), 2707–2719. https://doi.org/10.1109/TIFS.2018.2825958

Femi, P. S., & Vaidyanathan, S. G. (2022). An efficient ensemble framework for outlier detection using bio-inspired algorithm. *International Journal of Bio-Inspired Computation*, *19*(2), 67. https://doi.org/10.1504/IJBIC.2022.121239

Femi, P. S., Vaidyanathan, S. G., & Kala, A. (2021). Integrating fuzzy constraint with feature correlation for local outlier mining. *Sādhanā*, *46*(3), 172. https://doi.org/10.1007/s12046-021-01688-z

Ghosh, S., & Shah, C. (2018). Towards automatic fake news classification. *Proceedings of the Association for Information Science and Technology*, *55*(1), 805–807. https://doi.org/10.1002/pra2.2018.14505501125

Goldani, M. H., Safabakhsh, R., & Momtazi, S. (2021). Convolutional neural network with margin loss for fake news detection. *Information Processing & Management*, *58*(1), 102418. https://doi.org/10.1016/j.ipm.2020.102418

Jehad, R., & A.Yousif, S. (2020). Fake News Classification Using Random Forest and Decision Tree (J48). *Al-Nahrain Journal of Science*, *23*(4), 49–55. https://doi.org/10.22401/ANJS.23.4.09

Kala, A., & Vaidyanathan, S. G. (2022). Forecasting monthly rainfall using bio-inspired artificial algae deep learning network. *Fluctuation and Noise Letters*, *21*(02), 2250018. https://doi.org/10.1142/S0219477522500183

Kaliyar, R. K., Goswami, A., & Narang, P. (2021). EchoFakeD: Improving fake news detection in social media with an efficient deep neural network. *Neural Computing and Applications*, *33*(14), 8597–8613. https://doi.org/10.1007/s00521-020-05611-1

Kaliyar, R. K., Goswami, A., Narang, P., & Sinha, S. (2020). FNDNet – A deep convolutional neural network for fake news detection. *Cognitive Systems Research*, *61*, 32–44. https://doi.org/10.1016/j.cogsys.2019.12.005

Kula, S., Kozik, R., & Choraś, M. (2022). Implementation of the BERT-derived architectures to tackle disinformation challenges. *Neural Computing and Applications*, *34*(23), 20449–20461. https://doi.org/10.1007/s00521-021-06276-0

Lazer, D. M. J., Baum, M. A., Benkler, Y., Berinsky, A. J., Greenhill, K. M., Menczer, F., Metzger, M. J., Nyhan, B., Pennycook, G., Rothschild, D., Schudson, M., Sloman, S. A., Sunstein, C. R., Thorson, E. A., Watts, D. J., & Zittrain, J. L. (2018). The science of fake news. *Science*, *359*(6380), 1094–1096. https://doi.org/10.1126/science.aao2998

Mahara, G. S., & Gangele, S. (2022). Fake news detection: A RNN-LSTM, Bi-LSTM based deep learning approach. In *2022 IEEE 1st International Conference on Data, Decision and Systems (ICDDS)* (pp. 01–06). https://doi.org/10.1109/ICDDS56399.2022.10037403

Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Grey Wolf Optimizer. *Advances in Engineering Software*, *69*, 46–61. https://doi.org/10.1016/j.advengsoft.2013.12.007

Ozbay, F. A., & Alatas, B. (2021). Adaptive Salp swarm optimization algorithms with inertia weights for novel fake news detection model in online social media. *Multimedia Tools and Applications*, *80*(26–27), 34333–34357. https://doi.org/10.1007/s11042-021-11006-8

Pandey, S., Prabhakaran, S., Subba Reddy, N. V., & Acharya, D. (2022). Fake news detection from Online media using Machine learning Classifiers. *Journal of Physics: Conference Series*, *2161*(1), 012027. https://doi.org/10.1088/1742-6596/2161/1/012027

Rajalakshmi, V., & Ganesh Vaidyanathan, S. (2022a). Hybrid CNN-LSTM for Traffic Flow Forecasting. In G. Mathur, M. Bundele, M. Lalwani, & M. Paprzycki (Eds.), *Proceedings of 2nd International Conference on Artificial Intelligence: Advances and Applications* (pp. 407–414). Springer Nature Singapore. https://doi.org/10.1007/978-981-16-6332-1_35

Rajalakshmi, V., & Ganesh Vaidyanathan, S. (2022b). MLP-PSO framework with dynamic network tuning for traffic flow forecasting. *Intelligent Automation & Soft Computing*, *33*(3), 1335–1348. https://doi.org/10.32604/iasc.2022.024310

Ruchansky, N., Seo, S., & Liu, Y. (2017). *CSI: A Hybrid Deep Model for Fake News Detection*. https://doi.org/10.48550/ARXIV.1703.06959

Saleh, H., Alharbi, A., & Alsamhi, S. H. (2021). OPCNN-FAKE: Optimized Convolutional Neural Network for Fake News Detection. *IEEE Access*, *9*, 129471–129489. https://doi.org/10.1109/ACCESS.2021.3112806

Sastrawan, I. K., Bayupati, I. P. A., & Arsa, D. M. S. (2022). Detection of fake news using deep learning CNN–RNN based methods. *ICT Express*, *8*(3), 396–408. https://doi.org/10.1016/j.icte.2021.10.003

Shu, K., Cui, L., Wang, S., Lee, D., & Liu, H. (2019). dEFEND: Explainable Fake News Detection. *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 395–405. https://doi.org/10.1145/3292500.3330935

Smith, L. N. (2018). *A disciplined approach to neural network hyper-parameters: Part 1 -- learning rate, batch size, momentum, and weight decay* (Version 2). arXiv. https://doi.org/10.48550/ARXIV.1803.09820

Sudhakar, M., & Kaliyamurthie K. P. (2023). Effective prediction of fake news using a learning vector quantization with hamming distance measure. *Measurement: Sensors*, *25*, 100601. https://doi.org/10.1016/j.measen.2022.100601

Yang, Y., Zheng, L., Zhang, J., Cui, Q., Li, Z., & Yu, P. S. (2018). *TI-CNN: Convolutional Neural Networks for Fake News Detection* (Version 3). arXiv. https://doi.org/10.48550/ARXIV.1806.00749

Zhang, C., Gupta, A., Kauten, C., Deokar, A. V., & Qin, X. (2019). Detecting fake news for reducing misinformation risks using analytics approaches. *European Journal of Operational Research*, *279*(3), 1036–1052. https://doi.org/10.1016/j.ejor.2019.06.022