FMDB *Transactions on Sustainable Computer Letters*

Advanced Fake News Detection Using BEOSA-Based and an Attentive Convolutional Transformer

Rakesh Chandrashekar[1,*], Jayasheel Kumar[2], M. Arunadevi Thirumalraj[3], S. Gopikha[4], Prasanna R. Christodoss[5]

[1,2]Department of Mechanical and Engineering, New Horizon College of Engineering, Bengaluru, Karnataka, India.
[3]Department of Computer Science and Engineering, Karunya Institute of Technology and Science, Coimbatore, Tamil Nadu, India.
[3]Department of Computer Science and Business Management, Saranathan College of Engineering, Tiruchirappalli, Tamil Nadu, India.
[4]Department of Information Technology, St. Joseph's College of Engineering, Chennai, Tamil Nadu, India.
[5]Department of Computing, Mathematics and Physics, Messiah University, Mechanicsburg, Pennsylvania, United States of America.
rakesh2687@gmail.com[1], jayasheel.81088@gmail.com[2], aruna.devi96@gmail.com[3], gopikha.in@gmail.com[4], prchristodoss@messiah.edu[5]

Abstract: The rapid dissemination of false information, known as "fake news," has been made possible by the advent of social media platforms. False information not only deceives individuals but also fools the community. The spread of false information and the erosion of trust in information sources have harmed individuals and society, driven by the prevalence of poor-quality content on social media platforms. To improve classification accuracy and identify false information, this work utilises a state-of-the-art algorithm and a strict pipeline for preprocessing and feature extraction. In the preprocessing stage, the first step is to eliminate HTML tags, lowercase words, and stop words. The process of feature extraction using TF-IDF and word embeddings can capture more nuanced patterns in language. An innovative Attentive Convolutional Transformer (ACT) model that combines Transformer and CNN architectures is used to detect false information during classification. The Binary Ebola Optimisation Search Algorithm (BEOSA) is used for ACT hyperparameter tuning. Model discrimination and generalisation are both enhanced by BEOSA.

Keywords: Convolutional Neural Network; Fake News; Social Media; Term Frequency-Inverse Document Frequency; Bogus News; HTML Tags; CNN Architectures; Quick Transmission.

Received on: 25/01/2025, **Revised on:** 04/04/2025, **Accepted on:** 16/05/2025**, Published on:** 22/11/2025

Journal Homepage: https://www.fmdbpub.com/user/journals/details/FTSCL

DOI: https://doi.org/10.69888/FTSCL.2025.000487

Cite as: R. Chandrashekar, J. Kumar, M. A. Thirumalraj, S. Gopikha, and P. R. Christodoss, "Advanced Fake News Detection Using BEOSA-Based and an Attentive Convolutional Transformer," *FMDB Transactions on Sustainable Computer Letters*, vol. 3, no. 4, pp. 213–226, 2025.

1. Introduction

[*]Corresponding author.

The proliferation of handheld devices and high-speed Internet access has led to a massive surge in the usage of digital media. According to the 2020 Digital Global Report, there were 4.75 billion digital media users and 301 million social media users worldwide in 2020 [1]. The world is becoming more interconnected as a result of this digitalisation. People can now access information anywhere in the world with just a single click, thanks to this advancement [2]. This change has brought several benefits and some challenges. One of the issues the digital community is currently facing is the spread of misinformation, also known as fake news [3]. The ubiquitous propaganda known as "fake news" makes use of social media platforms like Facebook, Twitter, and Snapchat to spread misleading information online and sway public opinion. In terms of news consumption, social media can inform the public about breaking news but can also serve as a vehicle for the dissemination of misinformation. On the other hand, social media provides quick, easy, and inexpensive access to news and information, as well as global updates on events [4].

Furthermore, fake news is spread on the Internet because it is so easy to use and has so little regulation. Due to its influence during the 2016 US Presidential election, misinformation has been the subject of extensive media coverage over the past three years [5]. Studies have shown that only 54% of people can detect deception on their own without specialised training [6]. As such, an automated method for accurately classifying fake and real news is required. Although some research has been conducted, further attention and investigation are still necessary [7]. By automatically categorising the news, the proposed study aims to prevent the dissemination of false information and rumours and help readers determine whether to trust the news source [8]. Furthermore, several websites, including PolitiFact, FactCheck, and The Washington Post Fact Checker, among others, aim to verify the veracity of news [9]. On the other hand, the industrial sector cannot simply detect fake news. Research and service providers must collaborate to identify false information in a timely and effective manner. It is challenging to identify fake news accurately for several reasons. The language used in fake news and real news is similar. Identifying accurate news content can be challenging, as most fake news draws on real news.

Because fake news varies across domains, supervised learning requires substantial domain-annotated data [10]. The difficult task of identifying fake news involves several disciplines, including data science, feature engineering, psychology, social science, journalism, statistics, and machine learning. A model train designed to identify bogus political news would not work well for identifying bogus health care news, claims Long [11]. Large datasets were used to train deep learning models gathered from various domains, which are therefore necessary. However, to enhance detection performance, several outstanding issues must be addressed. Because deep learning can analyse vast quantities of data and spot trends and inconsistencies in text, images, and videos, it is a crucial tool in the fight against fake news. Natural language processing (NLP) and image analysis are two methods that deep learning models use to detect semantic anomalies, identify misleading content, and accurately classify the credibility of information. These models help create reliable systems to halt the dissemination of misleading data about digital platforms by continuously learning from and adapting to new deceptive strategies [12]; [13].

1.1. Motivation

Deep learning-based fake news detection utilises cutting-edge algorithms to combat misinformation and safeguard the integrity of information ecosystems. It enables systems to differentiate between real and fake content by utilising deep learning, helping prevent the propagation of misleading and manipulative information. By encouraging informed decision-making, this technology preserves democratic values while also enhancing public confidence in the media. Deep learning models, with their capacity to examine vast volumes of data and identify subtle patterns, are indispensable tools in the ongoing fight against misinformation, promoting a more honest and transparent digital environment for everyone.

1.2. Main Contributions

- Implements a comprehensive preprocessing pipeline for fake news detection, involving the removal of HTML tags, lowercase conversion, and stop word elimination.
- Utilises both TF-IDF and word embeddings for feature extraction to capture nuanced linguistic patterns in fake news articles.
- Introduces the Attentive Convolutional Transformer (ACT) model for classification, integrating the strengths of Transformer and CNN architectures to identify fake news efficiently.
- Enhances the ACT model's performance through hyperparameter tuning using the Binary Ebola Optimisation Search Algorithm (BEOSA), improving its discriminative power and generalisation capabilities.

2. Related Work

The research conducted by Alyoubi et al. [14] provided a prototype for identifying false information on Twitter using deep learning. The model utilised the user's social context and the news content they contributed to spreading. Extensive experiments were conducted using two-word embedding models and different deep learning algorithms to find an efficient fake news

detection model. A self-created dataset was used to assess the experiments. According to the experimental results, the convolutional neural network (CNN) model of MARBERT achieved higher accuracy, with an F1-score of 0.956. This outcome proved that the model successfully identified false information in Arabic Tweets about a range of subjects. The research by Alarfaj and Khan [15] investigated the categorisation of fake news utilising a range of DL and ML methods. A popular "Fake News" dataset was downloaded from Kaggle, which included an annotated news compilation. A range of machine learning models were employed, including logistic regression (LR), Gaussian naïve Bayes (GNB), multinomial naïve Bayes (MNB), Bernoulli naïve Bayes (BNB), and passive-aggressive classifier (PAC). Furthermore, CNN-LSTM, CNN, and long short-term memory (LSTM) were investigated as different DL models. The performances of these models were evaluated using key assessment criteria, including F1 score, recall, accuracy, and precision. To guarantee peak performance, hyperparameter tuning and cross-validation were also performed.

The findings offered insights into the advantages and disadvantages of each model for categorising false news. It was found that DL models performed better than traditional ML models, especially LSTM and CNN-LSTM. In classification tasks, these models showed robustness and improved accuracy. These results demonstrate the effectiveness of DL models in combating the dissemination of false information and highlight the importance of applying cutting-edge methods to this challenging issue. The study by Nadeem et al. [16] enhanced the detection of propaganda and fake news by extending the concept of symmetry into advanced natural language processing methods using deep learning. This paper proposed the hybrid HyproBert model for automatic detection of fake news. First, DistilBERT was used by the suggested HyproBert framework for word embeddings and tokenisation. The convolution layer used the embeddings as input to extract and highlight the spatial features. The result was then passed to BiGRU, enabling the extraction of contextual features. CapsNet and the self-attention layer then simulated the hierarchical relationships among spatial features in the BiGRU output. To incorporate all the necessary characteristics for categorisation, a dense layer was finally implemented. Two fake news datasets (FA-KES and ISOT) were used to evaluate the proposed HyproBert model. The research by Chen et al. [17] employed a variety of deep learning frameworks to compare and identify false information in both Chinese and English, using distinct text feature selections.

For a subsequent true/false prediction, the model was trained on the textual features of all types of information, including both real and fake data. Three models were chosen for fake news detection: the gated recurrent unit (GRU), the long and short-term memory (LSTM), and the bidirectional long and short-term memory (BiLSTM). BiLSTM achieved the best detection results. For Chinese texts, the system achieved a detection accuracy of 94%. For English texts with one or two sentences, the accuracy was 99%. For English texts with longer sentences, the accuracy was 99%. The research by Kishwar and Zafar [18] also employed several cutting-edge artificial intelligence techniques to evaluate the developed dataset. The following five machine learning methods were applied: Naïve Bayes, SVM, Decision Trees, KNN, and Logistic Regression. GloVe and BERT embeddings were utilised with CNN and LSTM, two deep learning techniques. The precision, accuracy, recall, and F1-score of each applied model and embedding were used to compare their respective performances. The outcomes demonstrated the best performance of an LSTM initialised with GloVe embeddings. By contrasting the incorrectly classified samples with human judgements, the study also examined the misclassified samples. In the paper by Choudhury and Acharjee [19], a variety of datasets were used to compare classifiers for SVM, Naïve Bayes, Random Forest, and Logistic Regression to detect fake news. In the datasets containing false job postings, fake news, and liars, the SVM classifier demonstrated the highest accuracy, scoring 61%, 97%, and 96%, respectively.

Once again, the unique GA-based fake news detection algorithm considered SVM, Naïve Bayes, Random Forest, and Logistic Regression as fitness functions. The suggested algorithm's SVM and LR classifiers each achieved 61% accuracy on the LIAR dataset. In comparison, the SVM and RF classifiers achieved 97% accuracy on the fake job posting dataset. Kumar et al. [20] presented the OptNet-Fake model for detecting bogus news. To identify fake news on social media, the suggested model trained a deep neural network using a meta-heuristic algorithm that selects features based on their utility. Using the term frequency inverse document frequency (TF-IDF) weighting technique, the d-D feature vectors for the textual data were first extracted. Then, using the extracted features, a modified grasshopper optimisation (MGO) algorithm was applied to select the most prominent features in the text. After selection, n-gram features were extracted from the data using a variety of convolutional neural networks (CNNs) with different filter sizes. To identify bogus news, these extracted features were ultimately concatenated. Standard evaluation metrics were applied to four datasets of actual fake news to assess the results. A comparative analysis was conducted between various meta-heuristic algorithms and contemporary techniques for detecting false news. The outcomes unequivocally demonstrated the OptNet-Fake model's superior performance compared to existing models across various datasets.

2.1. Research Gaps

Existing research on fake news detection predominantly focuses on Western languages, such as English, leaving a gap in comprehensive studies addressing fake news detection. Furthermore, while deep learning techniques show promise, more studies are needed to investigate their effectiveness across diverse cultural and linguistic contexts. Additionally, research often

lacks standardised evaluation metrics and datasets, hindering comparability between studies and the development of universally applicable detection models. Bridging these gaps requires interdisciplinary collaboration, standardised evaluation protocols, and a focus on diverse linguistic and cultural contexts to develop robust and effective solutions for detecting fake news.

3. Proposed Methodology

Figure 1 illustrates the model's suggested workflow for identifying false news using the BEOSA-ACT model.

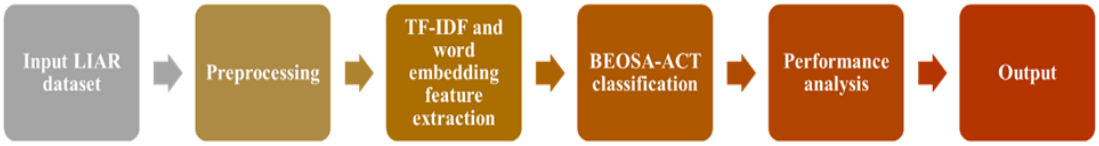

Figure 1: Block diagram

3.1. Dataset Description

The "LIAR" dataset originates from POLITIFACT.COM and comprises 12,800 human-labelled brief statements used in the study. A POLITIFACT.COM editor verifies the veracity of each statement. The label can rate accuracy in six categories: trousers fire, false, mostly true, half true, mostly true, and true [21]—the main dates in the statement range from 2007 to 2016. Republicans and Democrats coexist among the speakers, and each has a wealth of metadata, including past totals of false claims made by them. These statements are drawn from a variety of settings and speakers and cover a wide range of topics. Table 1 presents the dataset description. A statistical breakdown of the number of false statements each speaker has ever made is also included in Table 1. Standard deviation (σ), range, and mean (μ) are numerical variables that are used. However, there has been a categorical variation in the number of categories. Additionally, Table 1 displays the number of missing values for each attribute. There are only three missing values in the dataset: the context, the state data, and the speaker's job title. The study utilised 4,557 records, split into true and false class labels. Two thousand and fifty-five news records, respectively, possess both a genuine and a fake class label.

Table 1: An explanation of the LIAR dataset

No.	Feature Name	Datatype	Missing Values	Mean (μ) + Std (σ)	Range	No. of Categories
1	ID of the statement	Object	-	-	-	-
2	Label	Object	-	-	-	2
3	Statement	Object	-	-	-	4007
4	Subject(s)	Object	-	-	-	1823
5	Speaker	Object	-	-	-	9
6	Speaker's job title	Object	1184	-	-	656
7	State info	Object	926	-	-	-
8	Party affiliation	Object	-	-	-	4
9	Barely true counts	Int (64)	-	11.59 + 18.98	0–70	-
10	False counts	Int (64)	-	13.36 + 24.14	0–114	-
11	Half true counts	Int (64)	-	17.19 + 35.85	0–160	-
12	Mostly true counts	Int (64)	-	16.50 + 36.17	0–163	-
13	Pants on fire counts	Int (64)	-	6.25 + 16.18	0–70	-
14	The context (venue/location of the speech or statement)	Object	52	-	-	-

3.2. Preprocessing

Not every character in the fake news text has any significance [22]. For instance, most news articles contain words, punctuation, and other elements that don't relate to the text's topic. In addition to increasing the time required for classification learning, retaining every character will create high-dimensional features, increasing the amount of noisy data and reducing classification accuracy. For this reason, data preprocessing is required. The four preprocessing steps listed below are what this article used:

- Remove HTML tags

- Remove non-letters
- Convert sentences to lowercase and divide them
- Remove stop words

Using Python's Beautiful Soup library, step 1 involved removing HTML tags like '
 ' from the text of the comment. Regular expressions and the Natural Language Toolkit (NLTK) were used to implement steps 2-4. Here, the sentence is broken up into words, and the third step—using the NLTK's stop word list—converts each word to lower case, which is used in the fourth step and removes the words from both the stop list and the comment text. In the second step, the comment text is cleaned up of punctuation, numbers, and other non-English characters. Some noise words ("the", "is", "are", "a", "an", etc.) that don't describe the text subject are included in the stop word list. Furthermore, it added a few vocabulary words specific to this article when combined with the dataset's features. These specific high-frequency terms will influence the results of the subsequent keyword extraction and sentiment analysis. But since these terms typically depict picturesque locations objectively, they are not useful for sentiment analysis.

Feature Extraction: This method of dimensionality reduction partitions a large set of raw data into more manageable categories, enabling faster processing [23].

3.3. TF-IDF

The TF-IDF vector is created by combining term frequency with inverse document frequency. It lists the terms that appear most frequently in the news, as well as those that are used infrequently. Selecting a distinct term vector to serve as a training feature set is beneficial. The TF-IDF Vectorizer described in this article transforms text into vector features that an estimator can use as input. A dictionary's vocabulary uses the frequency of each word in the matrix to map each word token to a feature index, assigning a distinct index to each token.

$$W(d,t) = TF(d,t) * \log\left(\frac{N}{df(t)}\right) \quad (1)$$

Where d symbolises documents, t signifies terms found, where N is the total number of documents in the document.

3.3.1. Word Embedding

There are four methods for word embedding: Word2Vec, Embedding from Language Models (ELMo), Global Vectors for Word Representation (GloVe), and FastText. In the model, we have employed Word2Vec as one of these techniques. By utilising a neural network model, the Word2Vec algorithm extracts word associations from a vast corpus of text. After preprocessing the dataset, we convert it to a vector. The model cannot handle the data because news sentences vary in length, so we must add padding to each news item. The news is typically 50 pages long. Therefore, small news sentences are made into 50-word sentences with the aid of padding. A pre-trained vector has allowed us to create a matrix of (18210, 300).

Classification using Attentive Convolutional Transformer: This section provides a detailed presentation of the proposed ACT. Section 3.4.1 introduces the attentive convolution mechanism of ACT; Section 3.4.2 describes ACT's multi-head, multilayer architecture; and Section 3.4.3 presents the global attention mechanism for the final news representation.

3.4. Attentive Convolution Mechanism

The core function of ACT is the attentive convolution mechanism [24]. Through careful filter combination, news is transformed into a convolutional filter space after performing n-gram convolution over the news. Both local and global features of news can be captured by the attentive convolution mechanism, which utilises feature maps as attention weights in various ways. Representation of local features, considering a news input $t = [t_1, t_2, \dots, t_l]$First, we depict every word token t_i as the embedding of words $q_i \in \mathbb{R}^{d_w}$and acquire the embedded inputs $Q = [q_1, q_2, \dots, q_l]$ by searching for "embedding matrix" $W^{wrd} \in \mathbb{R}^{d_w \times V}$ where d_w is the word embedding dimension and V is the size of the vocabulary. Following the input embeddings, n-gram convolution is applied. Q employs convolutional filters in its execution, $F = [f_1, f_2, \dots, f_m]$ where $f_i \in \mathbb{R}^{nd_w}$ and n is the size of the convolution kernel. A matrix of feature maps $M \in \mathbb{R}^{m \times l}$ is produced in this manner:

$$M = Q \circledast F \quad (2)$$

Where $\circledast$ demonstrates the convolution process of f_i over Q. Equation 2 illustrates the specific calculation of the feature map's value:

$$m_{ij} = f\left(f_i^T \cdot Cat\left(q_j, q_{j+1}, \ldots, q_{j+n-1}\right) + b\right) \quad (3)$$

Where Cat means concatenation, f is a nonlinear activation function, and b is a bias term.

The final feature map's values indicate the relevance of n-grams and convolutional filters in a semantic sense. We preserve the sequential information in news by converting each n-gram from a convoluted word space into a more informative filter space used by convolutional neural networks. This is achieved by processing the feature map values as attention weights and carefully aggregating the semantic convolutional filters. Equation 4 represents the attentive convolution in formal terms for local feature representation.

$$O = F \cdot M = F \cdot (Q \circledast F) \quad (4)$$

whereas O= [o_1, o_2,…,o_l]∈R^(nd_w×l) is what's produced after careful convolution.

In contrast to self-attention, which maintains an intricate word space with dynamically updated elements based on the input, the attentive convolution mechanism forms an output space composed of globally and independently learned n-gram convolutional filters. Relevant n-grams will have small values in this space, while significant n-grams will be near the matching filters. Therefore, it is possible to effectively capture the significant local features (n-grams) that appear in the news. By applying careful convolution and the max-pooling method, a commonly used technique in traditional CNNs, the representation of global features can capture both local and global features of news. Each row of the feature map M's max-pooling process determines the news's overall relevance to each convolutional filter. The general semantics of news in the filter space can be inferred by carefully aggregating the convolutional filters using max pooling. Equation 5 represents the formal attentive convolution for the global feature representation.

$$g = F \cdot \max(M) \quad (5)$$

Where g∈R^ (nd_w), where max denotes row-wise max-pooling. In contrast to the methods used today, unlike traditional CNNs, the proposed attentive convolution utilises the semantic meaning of convolutional filters in addition to feature maps for news representation, whose outputs are derived from both feature maps and convolutional filters. This enables the model to learn convolutional filters efficiently, as each filter directly contributes to the final representation. Additionally, the pooling process in traditional CNNs ignores the news's sequential information, whereas the method's local feature representation preserves it while capturing significant n-gram features.

In contrast to the traditional attention mechanism, which determines attention weights by vector producting queries (Q) and keys (K), the proposed approach computes attention weights by convolving queries (Q) with keys (F). In the attention mechanism, the values and keys are convolutional filters learned end-to-end. In contrast to the vector product of individual words, the convolution operation operates over a broader context (n-grams), thereby improving the model's ability to capture significant n-gram features. The essential terms and phrases for text classification are precisely these n-gram features. Additionally, as noted in Section 3.4.1, because the output space is composed of convolutional filters that are independent of the inputs, it is more straightforward and informative.

3.4.1. Multi-head Multilayer Attentive Convolution

The proposed ACT features multiple heads and layers, inspired by the Transformer. Before performing *h*-attentive convolution, it linearly transforms the input embeddings Q *h* times for the *h*-head attentional convolution. Equation 6 illustrates how, following concatenation, the outputs from different attention heads are linearly transformed to the original input dimension.

$$\text{MultiHead}\ (Q) = W^O Cat(O_1, O_2, \ldots, O_h) \quad (6)$$

In this case, AttenConv denotes the suggested attentive convolution method. W_i^Q∈R^((d_w/h)×d_w) and W^O∈R^(d_w×nd_w) are the weight matrices for the linear transformations. It also employs layer normalisation and residual connections. To obtain higher-level local representations for various ACTs, all that is needed is to send the upper-layer input the local representations from the lower layer. First, the global representation comes from the uppermost ACT layer. Due to the multi-head structure of ACT, the model can jointly capture important n-gram features across different subword spaces, where the n-grams in these spaces contribute differently to the final representation. The framework can efficiently capture higher-level semantics because of its multilayer structure. The upper layer can produce more abstract and discriminative representations because it involves a larger context for convolution.

3.4.2. Global Attention and Classification

To classify fake news, it proposes a global attention mechanism that summarises the successive outputs of ACT. Local and global representations, along with each token's position data, are used to compute the attention weights. The local presence O ∈ R^ (d_w × l) and global representation g ∈ R^ (d_w) are acquired from ACT's upper layer. The embedding of position P∈R^(d_p×l) is acquired by converting the absolute position of each token to d_p- dimensional embeddings using a position embedding matrix that can be trained W^p∈R^(d_p×P), where P is the total number of jobs available. Attention weights α_i are determined using local representation o_i, global representation g, and position embedding p_i of each token. Equation 7 provides the completed written rendition:

$$r = O \cdot \text{Softmax}\left(f\left(W_o O + W_p P\right)^T c + \frac{O^T g}{\sqrt{d_w}}\right) \quad (7)$$

Where f is a non-linear activation function, $W_o \in \mathbb{R}^{d_a \times d_w}$ $W_p \in \mathbb{R}^{d_a \times d_p}$ are weight matrices for linear transformations, d_a is the attention dimension, $c \in \mathbb{R}^{d_a}$is a context vector that the neural network has learned, $\sqrt{d_w}$is a scaling factor that is dependent on the dimension of the input. To predict class probabilities, it feeds the completed depiction r to a classifier comprising a softmax layer and a fully connected layer. With momentum and learning rate decay, BEOSA is used to minimise categorical cross-entropy loss and centre loss, thereby training the model.

BEOSA Hyper Parameter Tuning: In this paper, BEOSA is utilised for hyperparameter tuning of the ACT model. This section outlines the recommended binarisation strategy for the EOSA algorithm. An overview of the immunity-based version of the EOSA algorithm and its design is presented [25]. An explanation of the process for creating and binarising the search space comes next. Next, the binary version of EOSA is developed and added to the binary search space. The continuous space can be mapped to a discrete space using the suggested transformation functions. There is also a discussion of the classification models that underpin the feature selection procedure.

3.4.3. Overview of EOSA and IEOSA

The Ebola virus propagation model and the traditional SIR model served as inspiration for the development of the EOSA metaheuristic. Immunity-based variant (IEOSA) is a new concept proposed because people naturally develop immunity to specific viral strains, and the potential defence that an immune individual can provide to a vulnerable person. Using continuous benchmark functions, both the immunity-based variant and the base algorithm underwent extensive testing [26]. The obtained results confirmed their viability. We provide a summary of the mathematical models used in the procedures, enabling discussion of the methods for the proposed BEOSA and BIEOSA. Equations (8) and (9) describe how to initialise the EOSA and IEOSA populations.

$$\text{ind}_i = L + \text{rand} * (U - L) \quad (8)$$

$$\text{ind}_{i+1} = g * \text{ind}_i * (1 - \text{ind}_i) \quad (9)$$

Where g is a constant (10), a real number generated at random is called a random variable, L is the minimum value, and U is the upper bound of the optimization problem. Equation (10) describes how infected individuals mutate in continuous space, where Δ is an individual's change factor and gbest is the globally best solution.

$$\text{ind}_i^{new} = \Delta * e^{rand} \cos(2\pi \text{ rand}) * (\text{ind}_i - \text{gbest}) \quad (10)$$

In light of the growing need to solve binary optimisation problems and the exceptional results obtained from the EOSA approach, this study proposes the binary EOSA (BEOSA). We provide in-depth details about the BEOSA and BIEOSA algorithm designs in the subsections that follow.

3.4.4. Binarisation of Search Space

There are individuals in the BEOSA search space whose representations are in the binary search space [27]. The entire population is comprised of individuals whose bodies are composed of binary numbers [28]. To facilitate the identification and differentiation of specific features from non-selected ones, this representation is necessary. First, two parameters, the dataset dimension D and the population size p, are used to determine the total number of people in the search area [29]. How many features in dataset X are used to calculate D, and when is the population's initialisation process performed, given that p size is specified? Using an iterative process, for every single ind i in the entire D dimension, the population is initialised to a value of

1 in ind _i. It is anticipated that the BEOSA application ⊕ will optimise solutions with a modified internal representation spanning the entire D dimension of the search space, with values for ind i ranging from 0 to 1. It is anticipated that the entire optimisation process will take several iterations to generate results for every person in _i. Presumably, cells with values of 1 "s are considered to translate into the chosen features [30]. Remember that the arbitrary solution's D dimension ind _i is comparable to the quantity of features |F| in the dataset of X. Consequently, we tally the quantity of 1s in the D dimension for each ind _i, which stands for the examples found in dataset X. To facilitate the EOSA's binarization, which is suitable for resolving the feature selection problem, the search space must be formalized. We outline the elements of the suggested BEOSA method in the subsection that follows [31].

Binarisation of EOSA (BEOSA): To maximise answers within a different solution space, a new version of EOSA is designed by incorporating additional operators into the existing algorithm. Determining the transformation functions to be used is the first step in converting a continuous solution representation and optimisation process into a discrete one. This is required to enable the new approach to handle feature selection-specific problems. The fitness function is changed to create the new variant BEOSA in the second operation modelled. Calculating the solutions' fitness is necessary to evaluate them and determine which individual is the global best. To fit the problem domain, a fitness function definition is provided.

3.4.5. Transformation of Method

It suggested using transformation functions to place the afflicted people in a discrete space. Two functions are described for the former group and two for the latter, which correspond to the well-known S-functions and V-functions categories. Equations (11) and (12) contain the S1 and S2 functions from the S-transform function, whereas the V1 and V2 functions from the V-function family are found in equations (13) and (14).

$$S1 = \frac{1}{1+e^{(-x/2)}} \tag{11}$$

$$S2 = 1 - \frac{1}{1+e^{x}} \tag{12}$$

$$V1 = \left|\frac{x}{\sqrt{2+x^2}}\right| \tag{13}$$

$$V2 = |\tan x| \tag{14}$$

Plotting the behaviour of the transform functions demonstrates their ability to produce patterns that closely resemble the functions they transform. For example, applying the function to values [-6,6] results in an S-shaped pattern from the two S-functions, but applying the V-functions to the same values results in a V-shaped pattern. The y-axis values should only ever appear as values between [0,1] when utilising the transform functions. Utilising these transformation functions is intended to ensure that an individual's feature position composition is either 0 or 1. Furthermore, these positions could raise the likelihood of a change in that person's innate makeup, making them a viable option for resolving feature selection issues. Equations (15) and (16) are used to illustrate this. The first of the two equations determines which function to use when applying the S-function (S1 or S2) and which function to use (T1 or T2) utilising the V-function. A determining factor guides this choice: if rand (0|1), the S2 or T2 function is called if the function generates 1, and the S1 or T1 function is called otherwise. The value found in equation two's second the k^"th "place in the individual's representation ind _i is changed to 1 when r>S(ind_i^k) for S-functions and r>T(ind_i^k) T-functions; in the absence of such, zero is allocated to the k^"th " the point at which k is between 0≤k<D, and r is produced at random between [0,1].

$$S\left(ind_i^k\right), T\left(ind_i^k\right) = \begin{cases} S2\left(ind_i^k\right), T2\left(ind_i^k\right) & \text{rand } (0 \mid 1) == 1 \\ S1\left(ind_i^k\right), T1\left(ind_i^k\right) & \text{rand } (0 \mid 1) == 0 \end{cases} \tag{15}$$

$$ind_i^k = \begin{cases} 1 & r > S\left(ind_i^k\right) \mid r > T\left(ind_i^k\right) \\ 0 & \text{otherwise} \end{cases} \tag{16}$$

An illustration showing how the transform functions are applied to translate the BEOSA, transitioning from discrete to continuous space. The susceptible group is the population with which the optimisation process starts. According to the EOSA method's natural phenomenon, some people are exposed to the virus, which causes some of them to be classified as members of the afflicted subset. These infected individuals are those who have been optimised over several iterations. Nearly every member of the susceptible subgroup is expected to transition to the infected subgroup during the iteration. For each ind _i in

the I subgroup, the k^”th “ position is altered through the use of either the V-function or the S-function, contingent upon the pos (i)< THRESHOLD criteria's satisfaction. Please note that the function pos(i) determines each person's current location and displacement, denoted as ind d_i. Throughout the experiment, the THRESHOLD parameter was set to 0.5. Which of the S-functions and V-functions is applied depends on how well this condition can be satisfied. The optimisation process's ultimate result is a vector of 0s and 1s. At the end of the iterative condition and the I subgroup, each individual in the population will have its fitness evaluated, thereby identifying the current worldwide best solution to the feature selection problem. The study's fitness function is covered in the subsection that follows.

3.4.6. Fitness and Cost Functions

The best-performing solution to the feature selection problem was identified by combining the evaluations of both the cost and fitness functions. Equation (17)'s fitness function assesses the solution based on its performance on a subset of the dataset's classifier, CLF, as X[:1^(ind_i)]. Additionally, when the control parameter, ω, is applied. The notation 1^ (〚” ind “〛 _i), as employed in the formula, yields the number of 1s in the array that represents each ind _i. Note that the notation |F| gives the number of the individual's chosen features, and D stands for the dimension of the feature in dataset X. ω was given a value of 0.99 for research purposes.

$$\text{fit} = \omega * \left(1 - \text{clf}\left(\text{X}[:1^{\text{ind i}}]\right)\right) + \left((1-\omega)\frac{|F|}{\text{D}}\right) \quad (17)$$

Equation (18) evaluates the cost function by deducting the value that fit returned from 1, thereby evaluating it from the fitness function's result. The quality and relevance of each best solution for each dataset are analysed and graphically interpreted, utilising the cost and fitness function values.

$$\text{cost} = 1 - \text{fit} \quad (18)$$

It illustrates the use of these functions in the following subsection's explanation of the recommended BEOSA method. The algorithm is first formalised in Algorithm 1.

Algorithm 1: BEOS Algorithm Pseudocode

```
Input: epoch, psize, srate, lrate
Output: gbest, costs, fcount
begin
Initialize the populations (psize) as S
Binarize the solution space S
Assign first item in population to first infected case (I)
  Make newly infected case global best
  while e < epoch and size (I) > 0 do:
Compute individuals to be quarantinea
I = difference of current infected cases (I) from quarantine cases
for i in 1 to size(I) do:
generate new infected (nI) case from S
for i in 1 to size(nI) do:
randomly generate d between 1 I 0
if displacement (nI[i]) > 0.5 do:
update size of nI using srate
s = use S2(nI[i]) to transform all dimensions if d is 1 , otherwise use S1(nI[i])
if s >= rand do:
nI[i] = 1
else:
nI[i] = 0
else:
  update size of nI using lrate
t = use T2(nI[i]) to transform all dimensions if d is 1 , otherwise use T1(nI[i])
if t >= rand do:
nI[i] = 1
else:
 nI[i] = 0
```

Evaluate new fitness of nI[i]
add (nI) cases to (I) cases
Update all compartment
Update best solution so far
Increment e by 1
End while
Compute feature count (fcount)
Return best solution, cost of best solution, fcount

4. Results and Discussion

Experimental Setup: Python 3.9.0 was used to implement the model, and several libraries, including matplotlib, Keras, and Shiny, were used.

4.1. Performance Metrics

The performance measures were determined using Equations (19–22):

$$\text{Accuracy (ACC)} = \frac{TP+TN}{TP+TN+FP+FN} \tag{19}$$

$$\text{Precision (PR)} = \frac{TP}{TP+FP} \tag{20}$$

$$\text{Recall(RC)} = \frac{TP}{TP+FN} \tag{21}$$

$$\text{F1Score (F1)} = \frac{2TP}{2TP+FP+FN} = \frac{2\times Precision\times Recall}{Precision+Recall} \tag{22}$$

4.2. Classification Validation

Table 2 and Figure 2 present the training validation results of various models, including AlexNet, ShuffleNet, ResNet50, SqueezeNet, and the proposed BEOSA-ACT model.

Table 2: Training validation of the proposed model

Models	ACC	PR	RC	F1
Alex Net	91.50	90.22	91.12	91.56
Shuffle Net	92.42	91.23	92.33	92.32
ResNet50	93.35	93.36	94.35	93.54
Squeeze Net	94.72	94. 73	94.87	94.43
Proposed BEOSA-ACT model	95.95	95.80	95.92	95.81

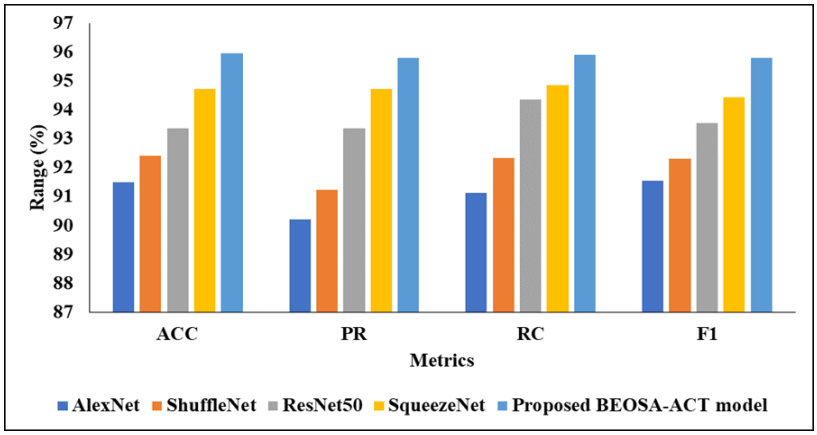

Figure 2: Training validation of the proposed BEOSA-ACT model

The performance of each model is assessed using accuracy (ACC), precision (PR), recall (RC), and F1 score. Among the models, the proposed BEOSA-ACT model outperforms the others with an impressive accuracy of 95.95%. Additionally, it demonstrates superior precision, recall, and F1 scores of 95.80%, 95.92%, and 95.81%, respectively. Notably, SqueezeNet also shows strong performance across all metrics, achieving 94.72% accuracy and 94.43% F1 score. These findings demonstrate the effectiveness of the proposed BEOSA-ACT model for data identification and classification, highlighting its potential for practical application in real-world scenarios. Table 3 and Figure 3 present the testing validation outcomes for several models, including AlexNet, ShuffleNet, ResNet-50, SqueezeNet, and the proposed BEOSA-ACT model. Evaluation metrics, including accuracy (ACC), precision (PR), recall (RC), and F1 score, are used to assess each model's performance. Remarkably, the proposed BEOSA-ACT model achieves exceptional results, with an accuracy of 99.77% and high precision, recall, and F1 scores of 99.31%, 99.24%, and 99.43%, respectively.

Table 3: Testing validation

Models	ACC	PR	RC	F1
Alex Net	97.91	97.94	97.67	97.53
Shuffle Net	98.47	97.69	98.67	98.22
ResNet50	98.70	98.40	98.68	98.66
Squeeze Net	97.92	98.68	98.21	98.79
Proposed BEOSA-ACT model	99.77	99.31	99.24	99.43

ResNet50 also demonstrates strong performance, closely trailing the BEOSA-ACT model with 98.70% accuracy and 98.66% F1 score. These findings underscore the BEOSA-ACT model's superior ability to accurately classify data during testing, suggesting its potential for robust deployment in practical scenarios.

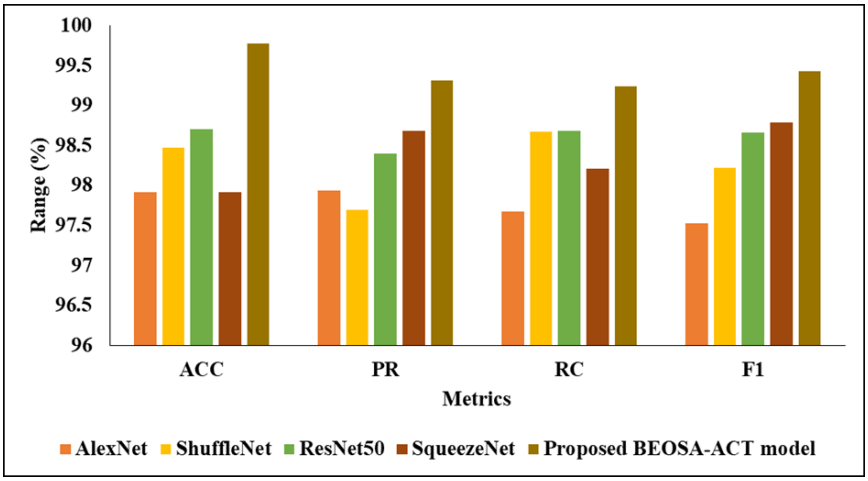

Figure 3: Testing analysis of the proposed BEOSA-ACT model

Table 4 and Figures 4 and 5 present the comparative performance of the ACT model with and without the BEOSA hyperparameter tuning technique. In the absence of BEOSA, Alex Net achieves an accuracy of 81.42%, precision of 81.17%, recall of 70.4%, and F1-score of 67.82%. Upon integrating BEOSA into the model, substantial improvements are observed, elevating accuracy to 97.91%, precision to 97.94%, recall to 97.67%, and F1-score to 97.53%.

Table 4: Comparison with and without optimisation

Models	Without BEOSA				With BEOSA			
	ACC (%)	PR (%)	RC (%)	F1 (%)	ACC (%)	PR (%)	RC (%)	F1 (%)
Alex Net	81.42	81.17	70.4	67.82	97.91	97.94	97.67	97.53
Shuffle Net	82.54	82.21	82.62	79.53	98.47	97.69	98.67	98.22
ResNet50	85.38	85.23	87.61	86.22	98.70	98.40	98.68	98.66
Squeeze Net	92.10	92.07	91.26	90.34	97.92	98.68	98.21	98.79
Proposed BEOSA-ACT model	93.21	93.13	93.02	93.09	99.77	99.31	99.24	99.43

ShuffleNet similarly experiences significant improvements when BEOSA is employed, with accuracy increasing from 82.54% to 98.47%, precision rising from 82.21% to 97.69%, recall improving from 82.62% to 98.67%, and F1-score increasing from 79.53% to 98.22%. ResNet50 also demonstrates notable improvements across all metrics with BEOSA, achieving 98.70% accuracy, 98.40% precision, 98.68% recall, and 98.66% F1-score, which significantly surpasses its performance without BEOSA. Squeeze Net exhibits marked improvements in accuracy (97.92%), precision (98.68%), recall (98.21%), and F1-score (98.79%) with BEOSA.

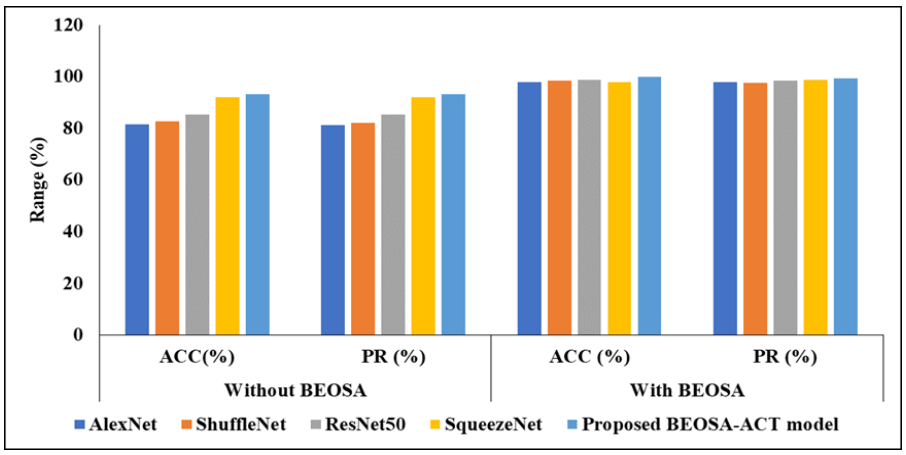

Figure 4: ACC and PR comparison

The proposed BEOSA-ACT model consistently outperforms other models, achieving 99.77% accuracy, 99.31% precision, 99.24% recall, and 99.43% F1-score, demonstrating significant enhancements enabled by BEOSA (Figure 5).

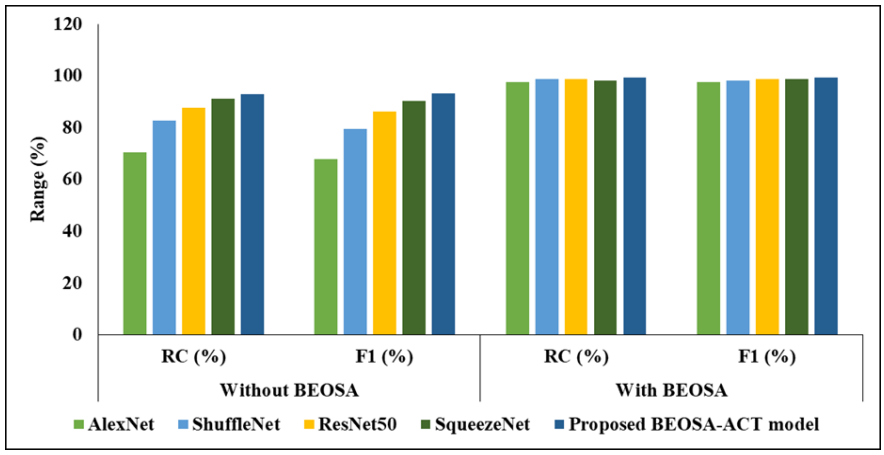

Figure 5: RC and F1 validation

These findings underscore the efficacy of BEOSA in fine-tuning the ACT model's hyperparameters, resulting in superior classification accuracy and robustness across diverse datasets.

5. Conclusion

Ultimately, the study provides a comprehensive framework for detecting fake news, integrating sophisticated preprocessing techniques, advanced feature extraction methods, and a novel classification model. By meticulously preprocessing text data, extracting informative features using TF-IDF and word embeddings, and combining Transformer and CNN architectures in the Attentive Convolutional Transformer (ACT) model, it achieves remarkable accuracy in distinguishing genuine from fake news articles. The incorporation of the Binary Ebola Optimisation Search Algorithm (BEOSA) for hyperparameter tuning further enhances the model's robustness and performance. The BEOSA-ACT model's suggested outcomes demonstrate exceptional performance metrics, including 99.77% accuracy, 99.31% precision, 99.24% recall, and 99.43% specificity. Future work will explore ensemble techniques, multi-modal data integration, and real-time deployment strategies to enhance fake news detection further.

Acknowledgment: The authors collectively extend their gratitude to the faculty and research support teams of New Horizon College of Engineering, Karunya Institute of Technology and Science, Saranathan College of Engineering, St. Joseph's College of Engineering, and Messiah University for their invaluable guidance. Their combined encouragement and academic resources significantly contributed to the successful completion of this work.

Data Availability Statement: The datasets generated and analyzed during this study are securely maintained by the authors' team and can be accessed upon reasonable request. Any shared data will follow institutional policies and protect participant confidentiality.

Funding Statement: This work was conducted entirely by the authors, without external funding support from any agencies or grants.

Conflicts of Interest Statement: The authors jointly declare that they have no known financial or personal conflicts of interest that could have influenced the outcomes of this research.

Ethics and Consent Statement: All research procedures were conducted in accordance with ethical standards. Necessary approvals were obtained, and informed consent was received from all participants involved in the study.

References

1. S. H. Kong, L. M. Tan, K. H. Gan, and N. H. Samsudin, "Fake News Detection using Deep Learning," *2020 IEEE 10th Symposium on Computer Applications and Industrial Electronics (ISCAIE),* Malaysia, 2020.
2. Z. Khanam, B. N. Alwasel, H. Sirafi, and M. Rashid, "Fake news detection using machine learning approaches," *in Proc. Int. Conf. Electr., Commun. Comput. Eng. (ICECCE),* Kuala Lumpur, Malaysia, 2021.
3. 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," *in IEEE Access*, vol. 9, no. 11, pp. 156151-156170, 2021.
4. T. Chauhan and H. Palivela, "Optimization and improvement of fake news detection using deep learning approaches for societal benefit," *Int. J. Inf. Manage. Data Insights*, vol. 1, no. 2, p. 100051, 2021.
5. S. R. Sahoo and B. B. Gupta, "Multiple features based approach for automatic fake news detection on social networks using deep learning," *Appl. Soft Comput.*, vol. 100, no. 3, p. 106983, 2021.
6. J. A. Nasir, O. S. Khan, and I. Varlamis, "Fake news detection: A hybrid CNN-RNN based deep learning approach," *Int. J. Inf. Manage. Data Insights*, vol. 1, no. 1, p. 100007, 2021.
7. I. Ahmad, M. Yousaf, S. Yousaf, and M. O. Ahmad, "Fake news detection using machine learning ensemble methods," *Complexity*, vol. 2020, no. 1, pp. 1–11, 2020.
8. M. Umer, Z. Imtiaz, S. Ullah, A. Mehmood, G. S. Choi, and O. Byung-Won, "Fake News Stance Detection Using Deep Learning Architecture (CNN-LSTM)," *in IEEE Access*, vol. 8, no. 8, pp. 156695-156706, 2020.
9. R. K. Kaliyar, A. Goswami, P. Narang, and S. Sinha, "FNDNet–a deep convolutional neural network for fake news detection," *Cogn. Syst. Res.*, vol. 61, no. 6, pp. 32–44, 2020.
10. D. Mouratidis, M. N. Nikiforos, and K. L. Kermanidis, "Deep learning for fake news detection in a pairwise textual input schema," *Computation*, vol. 9, no. 2, p. 20, 2021.
11. R. K. Kaliyar, A. Goswami, and P. Narang, "EchoFakeD: improving fake news detection in social media with an efficient deep neural network," *Neural Comput. Appl.*, vol. 33, no. 1, pp. 8597–8613, 2021.
12. T. Jiang, J. P. Li, A. U. Haq, A. Saboor, and A. Ali, "A Novel Stacking Approach for Accurate Detection of Fake News," *in IEEE Access*, vol. 9, no. 2, pp. 22626-22639, 2021.
13. A. Choudhary and A. Arora, "Linguistic feature based learning model for fake news detection and classification," *Expert Syst. Appl.*, vol. 169, no. 5, p. 114171, 2021.
14. S. Alyoubi, M. Kalkatawi, and F. Abukhodair, "The detection of fake news in Arabic tweets using deep learning," *Appl. Sci.*, vol. 13, no. 14, p. 8209, 2023.
15. F. K. Alarfaj and J. A. Khan, "Deep dive into fake news detection: Feature-centric classification with ensemble and deep learning methods," *Algorithms*, vol. 16, no. 11, p. 507, 2023.
16. M. I. Nadeem, S. A. H. Mohsan, K. Ahmed, D. Li, Z. Zheng, M. Shafiq, F. K. Karim, and S. M. Mostafa, "HyproBert: A fake news detection model based on deep hypercontext," *Symmetry,* vol. 15, no. 2, p. 296, 2023.
17. M. Y. Chen, Y. W. Lai, and J. W. Lian, "Using deep learning models to detect fake news about COVID-19," *ACM Trans. Internet Technol.*, vol. 23, no. 2, pp. 1–23, 2023.
18. A. Kishwar and A. Zafar, "Fake news detection on Pakistani news using machine learning and deep learning," *Expert Syst. Appl.*, vol. 211, no. 1, p. 118558, 2023.
19. D. Choudhury and T. Acharjee, "A novel approach to fake news detection in social networks using genetic algorithm applying machine learning classifiers," *Multimedia Tools Appl.*, vol. 82, no. 6, pp. 9029–9045, 2023.

20. S. Kumar, A. Kumar, A. Mallik, and R. R. Singh, "OptNet-Fake: Fake news detection in socio-cyber platforms using grasshopper optimization and deep neural network," *IEEE Trans. Comput. Social Syst.*, vol. 14, no. 8, pp. 1-12, 2023.
21. N. Aslam, I. U. Khan, F. S. Alotaibi, L. A. Aldaej, and A. K. Aldubaikil, "Fake detect: A deep learning ensemble model for fake news detection," *Complexity*, vol. 2021, no. 1, pp. 1–8, 2021.
22. W. Chen, Z. Xu, X. Zheng, Q. Yu, and Y. Luo, "Research on sentiment classification of online travel review text," *Appl. Sci.*, vol. 10, no. 15, p. 5275, 2020.
23. S. K. Bharti, S. Varadhaganapathy, R. K. Gupta, P. K. Shukla, M. Bouye, S. K. Hingaa, and A. Mahmoud, "Text-based emotion recognition using deep learning approach," *Computational Intelligence and Neuroscience*, vol. 2022, no. 1, pp. 1–12, 2022.
24. S. L. B. Pentakota, "Unveiling deepfake and fraudulent content generation in GPT models and countermeasures," *AVE Trends in Intelligent Computer Letters*, vol. 1, no. 1, pp. 41–50, 2025.
25. P. Li, P. Zhong, K. Mao, D. Wang, X. Yang, Y. Liu, J. Yin, and S. See, "ACT: An attentive convolutional transformer for efficient text classification," *in Proceedings of the AAAI Technical Track on Speech and Natural Language Processing II*, vol. 35, no. 15, pp. 13261–13269, 2021.
26. A. R. P. Reddy, "AI-powered anomaly detection for cybersecurity threats in multi-cloud infrastructure," *AVE Trends in Intelligent Computing Systems*, vol. 2, no. 2, pp. 77–86, 2025.
27. O. Akinola, O. N. Oyelade, and A. E. Ezugwu, "Binary Ebola optimization search algorithm for feature selection and classification problems," *Appl. Sci.*, vol. 12, no. 22, p. 11787, 2022.
28. M. A. Hasan, M. T. R. Mazumder, M. C. Motari, M. S. H. Shourov, and M. J. Howlader, "Assessing AI-enabled fraud detection and business intelligence dashboards for trust and ROI in U.S. e-commerce: A data-driven study," *AVE Trends in Intelligent Technoprise Letters*, vol. 2, no. 1, pp. 1–14, 2025.
29. A. Thirumalraj, T. Rajesh, and R. M. Das, "An improved ARO model for task offloading in vehicular cloud computing in VANET," *Research Square,* 2023. Available: https://assets-eu.researchsquare.com/files/rs-3291507/v1_covered_2b8de597-e0d0-42be-b5d7-f6de74e7356f.pdf?c=1697017963 [Accessed by 24/11/2024].
30. A. Thirunagalingam, "Bias detection and mitigation in data pipelines: Ensuring fairness and accuracy in machine learning," *AVE Trends in Intelligent Computing Systems*, vol. 1, no. 2, pp. 116–127, 2024.
31. T. M. Aruna, P. Kumar, N. N. Srinidhi, G. N. Divyaraj, K. Asha, A. Thirumalraj, and E. Naresh, "Effective utilisation of geospatial data for peer-to-peer communication among autonomous vehicles using optimized machine learning algorithm," *Research Square,* 2023. Available: https://www.researchsquare.com/article/peer_communication_among_autonomous_vehicles_using_Optimiz ed_Machi ne_learning_algorithm [Accessed by 12/11/2024].