Journal of Information and Organizational Sciences
Volume 49, Number 1 (2025)
Journal homepage: jios.foi.hr

JIOS

DOI: 10.31341/jios.49.1.6
Open Access

UDC 004.8:004.021:316.774:659.3
Original Scientific Paper

Hierarchical Deep Learning Model Optimization Using Enhanced Evolutionary-based Approach for Fake News Detection

Deepti Nikumbh[1]* and Anuradha Thakare[1]

[1]Department of Computer Engineering, Pimpri Chinchwad College of Engineering, Pune, Savitribai Phule Pune University, Pune, India 411044
*Correspondence: deepti.nikumbh@gmail.com

PAPER INFO

Paper history:
Received 23 November 2024
Accepted 31 March 2025

Citation:
Nikumbh D. & Thakare, A. (2025). Hierarchical Deep Learning Model Optimization Using Enhanced Evolutionary-based Approach for Fake News Detection. In Journal of Information and Organizational Sciences, vol. 49, no. 1, pp. 83-99

ABSTRACT

Multimodal fake information on social media is a growing concern worldwide. Existing deep learning-based solutions typically involve designing hierarchical models that capture relevant features from each modality, which are then fused for final classification. However, these models are often complex, with numerous trainable parameters, making them resource-intensive. This work introduces the Deep Learning Model with Evolutionary Computing Approach (DLECA), a novel method for compressing and optimizing hierarchical deep learning models (HDLM). It employs an enhanced genetic algorithm (GA) with a unique fitness function, dynamic crossover, and adaptive mutation strategies to achieve model compression, maintain accuracy, and balance exploration and exploitation during evolution. In comparison to manually designed HDLM, the proposed approach achieves up to 97.86% model compression with a 0.34% accuracy improvement, while a variant achieves 96.24% compression with a 0.23% accuracy improvement. Comparative analysis shows that DLECA outperforms Random Walk and Bayesian Optimization in multimodal fake news detection, offering a more efficient and accurate solution.

Keywords: Evolutionary computing, Deep Learning, model compression, model optimization, Genetic Algorithms

1. Introduction

Social media has revolutionized how people stay connected and access information. Its vast reach has become the primary platform for disseminating content to a broad audience. However, this widespread connectivity also presents a significant challenge: the propagation of misinformation, including fake news. Malicious actors exploit social media platforms to spread false information, often to mislead or manipulate the public. The impact of misinformation on social media is profound, affecting both individuals and society according to Shu et al.[1]. It has the potential to change public opinion, influence decision-making, and even cause civil unrest. This misinformation is frequently multimodal, combining text, images, and metadata to enhance its credibility and appeal.

Traditional fake news detection models, which rely on a single modality, often fail to capture the complexity of multimodal content. For instance, Pérez-Rosas et al.[2] explored a linguistic approach for detecting fake news, achieving 76% accuracy. While linguistic features showed promising results, the authors emphasized the need to integrate metadata and visual features for improved performance. Similarly, P. Qi et al.[27] analyzed the visual content of fake news using pixel and frequency domain features but highlighted the importance of exploring the relationship between textual and visual cues.

Fake news often blends deceptive text with manipulated images, making detection challenging when analyzing either modality in isolation. P. Li et al. [28] provided examples of multimodal news and emphasized the necessity of jointly analyzing text and images to enhance fake news detection accuracy.

Additionally, user metadata, such as the profile information of individuals sharing the post, can offer valuable insights according to Shu et al.[3]. Fake news is often propagated through networks of fake or bot accounts, which further complicates detection. Incorporating text, images, and metadata into a deep learning model is essential to address these challenges. However, the resulting model has complex hierarchical structures.

Training these models is time-consuming and demands significant storage and computational requirements. Additionally, the success of these models is largely reliant on the tuning of numerous hyperparameters. Manually selecting and optimizing these hyperparameters is a challenging and labor-intensive task, often requiring extensive experimentation to identify the best configurations.

Herein, we introduce a novel method that performs hierarchical deep learning model hyperparameter optimization and compression, offering a solution that not only reduces model size but also enhances accuracy.

The major contributions of this research are as follows:

1. **Model Compression** and **Hyperparameter Optimization**: A unique multi-objective fitness function is proposed which helps in reducing the model architecture without compromising the models' accuracy.
2. **Dynamic crossover strategy:** A novel technique that calculates crossover probability based on the fitness of the current population and updates it as evolution progresses.
3. **Adaptive mutation strategy**: A technique that computes mutation probability based on the crossover probability of the current population also gets updated as evolution progresses.
4. **Genetic Evolutionary Approach**: Introduce a novel DLECA, which performs evolution by genetic operators and optimizes model parameters dynamically.

A comprehensive performance evaluation of the proposed method against optimization algorithms from the literature is presented. By addressing the optimization of both compression and performance, our work aims to enhance the practicality and effectiveness of deep learning models for multimodal fake news detection on social media platforms.

The rest of the papers is organized as follows: Section 2, presents related work. Section 3 introduces the proposed methodology. In Section 4, we present the dataset under consideration, experiment setup and results and comparative analysis. Finally, Section 5 concludes the paper.

2. Related work

Existing literature explores complex deep learning models that integrate multiple modalities to address multimodal fake news detection. Many studies, such as S. Singhal et al. [12] and Khattar D. et al. [29], have developed deep learning architectures combining text and image modalities, whereas Raza et al. [30] incorporate social context alongside textual features of the news. Combining multiple modalities such as text and image with social context provides a more comprehensive analysis of fake news. However, integrating diverse architectures into a single complex model presents significant challenges. These include synchronizing feature extraction across modalities, and managing computational complexity. The resulting models are highly intricate, requiring extensive tuning and computational resources.

In literature, optimizing Deep Learning architectures such as Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTMs)has been extensively explored. Galván et al. [4] discuss neuroevolution, which is the process of using evolutionary-based approaches for deep neural network architecture optimization and training. Researchers have developed various strategies to enhance the efficiency and effectiveness of GA in this context, addressing the model performance, fitness evaluation, and the balance between exploration and exploitation-related challenges.

Baldominos et al.[5] explored optimizing the topology of CNNs using both GA and Grammatical Evolution (GE). The optimized CNN ensemble was applied to the MNIST dataset, and transfer learning was performed on the EMNIST dataset. However, this approach was limited to basic datasets and did not explore more complex or multimodal data.

Another approach to optimizing CNN architecture was provided by Sun et al.[6]. They used a variable-length encoding strategy for GA, a skip layer for deeper CNN, and a cache component to improve the efficiency of fitness evaluations. Naik et al.[7] used an adaptive tournament strategy, where the tournament size was dynamically adjusted based on the population's state. Jebraeily et al.[8] obtained an optimal structure for convolutional neural networks in terms of the number of layers and other parameters using GA. The authors further would like to design full parameter models using GA and use newer optimization algorithms for better accuracy. Shrestha et al.[9] also optimized the architecture of the deep learning model through enhanced GA with mtDNA to track individuals in the population with the same ancestry.

Wu et al. [10] performed hyperparameter optimization of LSTM architecture through the concept of parallel GAs. They created subgroups of individuals in the population and ran GA in parallel. Subgroup exchange of individuals is done to enhance diversity. Farrag et al. [11] performed optimization of stacked LSTM. The architecture parameters are considered single-objective multi-parameters discrete optimization problem and MAPE was used as evaluation measure.

Apart from GA other optimization algorithms such as random walk by Singhal et al. [12] and Bayesian optimization algorithm by Puentes et al. [13] are used for optimizing the hyperparameters of deep learning models. A graph-based version of Bayesian optimization was used by Ma et al. [14] for searching best deep neural architecture. Here the search space was made up of attributed graph each representing a neural architecture. Bakhshi et al. [15] also used a deep evolutionary approach for finding the best architecture of Convolutional Neural Network. A Particle Swarm Optimization (PSO) and GA-based optimization of three layer network was conducted by Mandal et al. [16].

Khan et al. [17] employed meta-heuristic algorithms, utilizing the Grey Wolf Optimization (GWO) algorithm for weight and bias optimization and the Strawberry (SB) algorithm for optimizing the learning rate of deep neural networks. Further, the authors would like to explore hybrid combinations of other meta-heuristic algorithms and keep a balance between exploration and exploitation. Hybrid approaches of GA and machine learning (ML) are used by Shannaq et al.[18] and Choudhury et al. [19].[18]applied GA to find optimal hyperparameters for support vector machine (SVM) and XGBoost ML models.[19] created various combinations such as GA and logistic regression, GA and SVM, GA and naïve bayes, GA and random forest for optimizing model parameters.

Kumar et al. [20] used a meta-heuristic approach in the context of fake news detection. A term frequency-inverse document frequency (TF-IDF) technique was used to extract text features traits of news. The most salient traits were selected using a modified grasshopper optimization (MGO) algorithm and a CNN performed the final classification of news. Zaheer et al. [21] combine metaheuristic approach with a hybrid filter and wrapper feature selection technique to obtain the important features from textual news. Uppada, et al. [22] extract optimal features from textual news and associated metadata using GA. Shah et al. [23] extracted optimal feature sets from news text and image using a cultural algorithm. Marsili-Libelli et al.[24]and Lin et al.[25] designed an adaptive mutation technique with [25] focused on both adaptive mutation and crossover in GA.

In the literature, deep learning architecture optimization, particularly for CNNs and LSTMs, has been performed across various domains.GA has been the primary optimization algorithm, various modifications to GA operators, such as adaptive tournament selection and modified crossover and mutation operations, have been proposed to improve optimization efficiency. Additionally, metaheuristic algorithms like GWO and SB have been utilized to optimize model weights and biases. However, there remains a need to design methods that accelerate fitness evaluation, develop full-parameter models using GA, and explore newer optimization algorithms for better accuracy. Furthermore, a balance between exploration and exploitation of the search space is crucial, especially when performing multi-objective optimization that evaluates multiple performance metrics of the model. These insights guided the development of the proposed DLECA.

Table 1 provides details about the configurations of GA operators used in the literature.

Reference number	**Algorithm/ Method/ Technique**	**Fitness**	**Population Size**	**Crossover rate**	**Mutation rate**	**Generation**	**Convergence criteria**
[5]	Genetic Algorithm and Grammatical Evolution	classification error with niching strategy		-	-	20	-
[6]	Genetic Algorithm with variable length encoding.	classification accuracy	20	0.9	0.2	20	classification accuracy does not change over a generation

[10]	Parallel genetic algorithms	RMSE, MAPE, R2	12	0.5	0.5	25	evolution reaches a set number.
[7]	Adaptive Genetic Algorithm.	Mean Average Precision	-	0.5 -1.0	0.5 -1.0	1500	specified no. of generations
[8]	Genetic Algorithm	Accuracy	50	0.52	0.24	250	specified no. of generations
[17]	Grey Wolf Optimization and Strawberry Metaheuristic Algorithms.	MSE	30	-	-	100	-
[18]	Genetic Algorithms with XGBoost and SVM.	Accuracy	-	-	-	100	Maximum generations reached
[11]	Genetic Algorithm	MAPE	50	10%	10%	100	-
[20]	Modified grasshopper optimization algorithm.	Error rate	200	-	-	300	Terminates after running fixed no. of iterations.

Table1. Summary of Genetic Algorithm Parameters Used in Literature

3. Proposed methodology

In this section, we discuss the DLECA approach used for the performance and architecture optimization of an HDLM. We begin by introducing the HDLM which is designed for multimodal fake news detection and comprises various deep learning components. Due to the complexity and heaviness of such architectures, selecting the right configuration of parameters is crucial to maintain accuracy without adding unnecessary computational overhead. Next, we review the GA, widely used in the literature for optimization processes. Finally, we present our proposed DLECA algorithm, specifically developed to address the challenges and achieve optimal performance.

3.1. Hierarchical Deep Learning Model (HDLM)for Fake News Detection

To effectively capture the multimodal features of news posts on social media, our hierarchical deep learning model incorporates three specialized feature extractors. The text feature extractor is made up of Bidirectional Encoder Representations from Transformers (BERT) and LSTM to extract long-term dependency and contextual information from text. For images, the image feature extractor employs a CNN to recognize and learn significant visual features. Additionally, the social context feature extractor uses a Multi-Layer Perceptron (MLP) to analyze cues and insights generated from user profiles and post metadata. These extracted features are then fused using an inter-attention mechanism, which effectively integrates the diverse modalities into a comprehensive multimodal feature vector. This vector is subsequently classified as either fake or real. Figure 1 depicts the architecture of the hierarchical deep learning model.

3.2. Genetic Algorithm

A Genetic Algorithm (GA) is an evolutionary computing approach based on natural selection and evolution. It is especially well-suited to handling difficult optimization issues, where traditional methods may fail. In a GA, potential problem solutions are represented as population individuals. The algorithm evolves this population over successive generations using GA operators: encoding, selection, crossover, and mutation.

- Encoding transforms potential solutions into a format that can be easily processed by GA, typically as chromosomes.

- The selection operator chooses individuals based on fitness value.
- The crossover operator then recombines pairs of selected individuals to produce offspring, while the mutation operator introduces random variations.

This evolution through selection, crossover, and mutation iteratively improves the population, driving it towards the optimal solution. GA explores the solution space efficiently converging on to quality solutions, making it a powerful tool for optimizing complex hierarchical deep learning models.

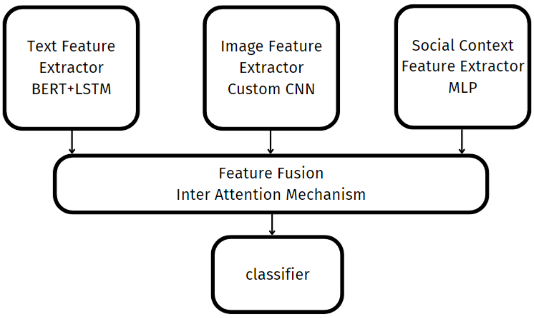

Figure 1. Hierarchical Deep Learning Model Architecture

3.3. Proposed Deep Learning Model with Evolutionary Computing Approach (DLECA)

The proposed approach is an enhanced GA to optimize a hierarchical deep learning model, focusing on both model accuracy and architecture compression. To facilitate this optimization process, a unique fitness function has been designed, balancing the trade-offs between model accuracy and compression. The algorithm begins by identifying the key hyperparameters of the model. An initial population of individual solutions is formed out of the hyperparameters which is further subjected to optimization. The optimization is performed using genetic operator's selection, crossover, and mutation through which the population evolves. Unlike many conventional methods where the crossover and mutation probabilities remain constant throughout the evolution process, this work introduces a dynamic crossover strategy. This strategy adjusts the crossover probability dynamically based on the current state of the population allowing the algorithm to better explore and exploit the solution space. Additionally, an adaptive mutation strategy is employed, which adjusts mutation probability in response to changes in crossover probability. This adaptive approach ensures a more efficient search for optimal solutions, leading to a compact and efficient model without compromising performance. The flow chart and of the proposed DLECA is presented in figure 2.

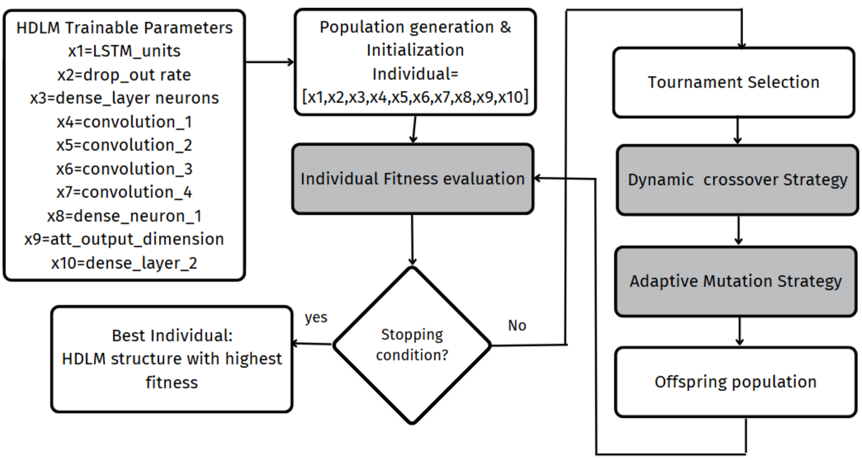

Figure 2. Flowchart of Proposed DLECA

3.3.1. Population Initialization:

The performance of a HDLM heavily relies on various hyperparameters. During the population initialization phase, individuals in the population are created using the hyperparameters of the hierarchical deep learning model. The key hyperparameters identified for this process and their respective ranges used for initializing the population are outlined in Table 2. These ranges are carefully determined to ensure that the model parameters do not increase excessively. The dropout rate is capped at 50%. Figure 3 demonstrates the individuals created in the population initialization phase. Each individual presents one set of hyperparameters of HDLM. Color of each element in the individual solution provides modality information, wherein blue represents text feature extractor, green for image feature extractor, grey for social context feature extractor and pink for attention mechanism. Consider the equation(1)

$$P_0 = \{I_1, I_2, I_3, \dots . I_N\} \tag{1}$$

WhereP_0 is initial population and N are individuals in population. Equation(2) I_i represent the ith individual in the population. The I_i is a vector of hyperparameters.

$$I_i = [x_{i1}, x_{i2}, x_{i3}, \dots . x_{ik}] \tag{2}$$

where k is the number of hyperparameters. The hyperparameter x_{ij} is selected from a discrete set of predefined values S_j

$$S_j = \{v_1, v_2, v_3, \dots . v_m\} \tag{3}$$

where m is the number of discrete values available for the j-th hyperparameter.

$$x_{ij} = S_j[(randomInt(1, m))] \tag{4}$$

Hyperparameter	**Description of the hyperparameter.**	**Range for population Initialization.**
LSTM units	No. of LSTM units used in the model	[25,50,75,100]
Drop out rate	No. of neurons to be dropped during training.	[10,20,30,40,50]
Convolution layer	No. of filters used in the convolution layer	[16,32,64,128]
Dense layer	No. of neurons in a dense layer	[128,256,512,1024]
output dimension	Output dimension after attention module.	[16,32,64]

Table 2. HDLM's Hyperparameter and population initialization

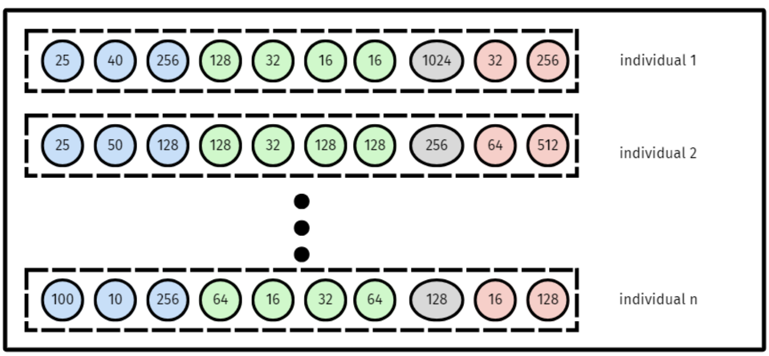

Figure 3. Population Initialization of DLECA

3.3.2. Fitness Function

The individual or solutions are evaluated by a fitness function, which can also be viewed as an objective function that the model seeks to optimize. In this work, we design a multi-objective fitness function aimed at reducing computational complexity by decreasing the model size without compromising accuracy. In this fitness function objective 1 focuses on model performance and is expressed in terms of confusion matrix parameters (refer to equation(6)). Objective 2 addresses model compression shown in equation(7), quantified by the model compression ratio. The formula is inspired by the image compression ratio. This ratio compares the total trainable parameters in the original or base model with those in the compressed model. Specifically, it is calculated by dividing the number of parameters in the original model by the number of parameters in the compressed model. This method provides a measure of how many parameters were eliminated or reduced in the compressed model compared to the original model. Combining the two objectives, Equation (5) presents the novel multi-objective fitness function designed to optimize both the performance and compression of the model. Here, w_{acc} is the weight assigned to the model's performance measured using accuracy, and w_{comp} is the weight for model compression. The weights are selected based on experimental observations balancing the tradeoff between model performance and computational efficiency. The values of w_{acc} and w_{comp} are empirically set to 0.7 and 0.3 to ensure that compression does not significantly degrade accuracy aiming for a small improvement or maintenance of performance.

$$fitness = w_{acc}\left(R_{performance}\right) + w_{comp}\left(R_{compression}\right) \tag{5}$$

Where,

$$R_{performance} = \frac{(TN+TP)}{(TP+FN+TN+FP)} \tag{6}$$

$$R_{compression} = \frac{No.of\ Original\ model\ parameters}{No.of\ compressed\ model\ parameters} \tag{7}$$

Since higher values of both objectives are desirable, the resultant function aims to maximize both accuracy and model compression, striking a balance between maintaining high performance and reducing model complexity.

3.3.3. Evolution Operations

In this section three evolution operations are performed on the initial population. Tournament selection is performed for the best parent selection. Further, two-point crossover and uniform mutation is performed using a unique dynamic crossover strategy and adaptive mutation technique.

1. Dynamic crossover strategy:

In approaches found in the literature, the crossover rate or probability is either fixed or varies within a predefined range during the evolution process (refer to Table 1). However, we introduce a unique technique where the crossover probability adapts or changes dynamically throughout the evolutionary process. The core idea behind this approach is to encourage crossovers when the population has fitter individuals and reduce them when the population is more homogenous. This method aims to dynamically balance the exploitation based on the current state of the population's fitness. The formula for dynamic crossover probability is provided in equation(8).

$$crossover_{prob} = \frac{\max\left(fitness_{of_{population}}\right)}{\sum_{i=1}^{N} fitness(i)} * initial\_cxpb \tag{8}$$

The ratio in the formula provides the measure of how dominant the best individual is compared to the entire population. If the best individual is significantly better , than this ratio will be higher, indicating that some individuals are outperforming others substantially. Conversely, if all individuals are performing similarly, the ratio will be lower.

By multiplying this ratio by the initial crossover probability (initial_cxpb), the crossover probability is adjusted dynamically. When the best individual significantly outperforms others, the crossover probability increases, promoting the sharing of superior genes. On the other hand, when all individuals perform similarly, the crossover probability decreases, reducing excessive mixing and allowing for more focused exploitation of the current solutions.

2. Adaptive Mutation Strategy:

Mutation probability is adjusted inversely with respect to crossover probability. By adjusting the mutation probability inversely with the crossover probability, we can balance exploration and exploitation. In case of high crossover probability, the algorithm relies more on recombining existing solutions, so the mutation probability is reduced to avoid introducing too much randomness. In case of low crossover probability, the algorithm relies less on recombination, so the mutation probability is increased to introduce more variations and explore the solution space more broadly. The equation (9) depicts the process.

$$mutation\_prob = base\_mutation\_prob * (1 - crossover\_prob) \tag{9}$$

4. Results and analysis

4.1. Dataset for Fake New Detection:

In this work for training and validating the model, FakeNewsNet dataset provided by Shu et al.[26] is used. FakeNewsNet is one of the very few datasets that provide multimodal information about a news on social media. The news is collected from fact checking websites like PolitiFact and GossipCop This multimodal information includes news title and news body, image related to the news and other social context and meta data about the post.

4.2. Manual Designed HDLM for Fake News Detection:

As mentioned in the previous section, this work is about optimizing a HDLM in the context of fake news detection. Table 3 presents manually designed architecture details of the HDLM along with trainable parameters in the sub-models.

Modality /Mechanism	**Sub-model**	**Trainable parameters**
Text	Fixed BERT followed by LSTM and dense layer	399312
Image	Convolutional Neural Network with four consecutive convolution and max pooling layers followed by dense layers.	9678528
Social Context	Two-layer perceptron model.	530944
Attention unit	Scale dot product attention with concatenation	98496
Downsampling	Dense layer	12352
Total Trainable parameters		10719632

Table 3. Total Trainable parameters of HDLM

4.3. Experimental Results

This section presents the detailed implementation of the proposed DLECA. The proposed DLECA is used to optimize the HDLM. The performance of DLECA is compared against the traditional GA, Bayesian optimizer and random walk.

4.3.1. Experimental setup

The proposed work was implemented using Google Colab Pro subscription, with hardware accelerators like high-performance GPUs (L4 GPU, A100 GPU) and TPUs (v2-8 TPU) along with high RAM configurations. Distributed Evolutionary Algorithms in Python (DEAP) library in python was used for performing operations involved in GA. This robust setup ensures efficient handling of memory-intensive tasks and speed up the execution.

4.3.2. Parameter configuration

Table 4 provides a comprehensive list of parameters and their assigned values required for implementing the DLECA. These values have been carefully determined by reviewing existing literature and selecting those that best support the optimization objectives of the proposed method.

Parameters	**Values**
Population Size	20
Selection technique	Tournament selection
Crossover technique	Two-point crossover
Mutation technique	Uniform mutation
Generations	10
Stopping criteria	Specified generations are reached.

Table 4. Parameter setting for DLECA

4.3.3. Performance Evaluation

Here, we present performance comparison of the proposed DLECA method and traditional GA with proposed fitness with manual parameter settings and other existing optimization algorithms from the literature [12][13]. We further present the best architectures identified by each algorithm, which are then integrated into the HDLM and trained on the FakeNewNet dataset. The results are evaluated by comparing the percentage of model compression and the percentage change in accuracy for each optimized architecture against the original manually tuned model. Additionally, we provide performance metrics and confusion matrix for each architecture.

To analyze the performance of the proposed DLECA, various performance plots such as the convergence plot, average fitness plot, population diversity plot, and crossover and mutation rate plots are presented. The convergence plot from figure 4,5 demonstrates that, compared to traditional GA with proposed fitness, the proposed DLECA algorithm tends to converge quickly, typically after just 8 generations. Additionally, the difference between the fitness of the best individual found by DLECA and that found by traditional GA with proposed fitness is substantial, highlighting the superior optimization capability of DLECA.

Figure 6 presents the diversity plots for the traditional GA with proposed fitness and optimized DLECA. Diversity plots visualize the diversity of population solutions in each generation. In the work, the diversity of the population for each generation is calculated based on the Euclidean distance between pairs of solution's parameter values. Then an average of all the Euclidean distance is taken to calculate the final diversity measure of population. The diversity values are then plotted against the generations to visualize diversity of population in each generation.

Diversity plots provide valuable insights, higher the values indicate more diversity among the individuals in population. From the plot of figure 6 it is evident that the individuals in each generation of the DLECA are more diverse compared to those in traditional GA, suggesting that the algorithm performs better exploration of the solution space and providing a solution without premature convergence.

Figure 7 illustrates the crossover and mutation probability plots throughout the evolutionary process. The algorithm starts with high initial crossover probability of 0.8 and low initial mutation probability of 0.5. Initially, as the population is randomly generated, the crossover and mutation probabilities are high indicating the exploration. As the generations progress and the fitness of the population improves, the algorithm shifts towards exploitation by increasing the crossover probability. The plots clearly demonstrate this transition, showing a high mutation rate in the early stages of DLECA and higher crossover probability in later stages.

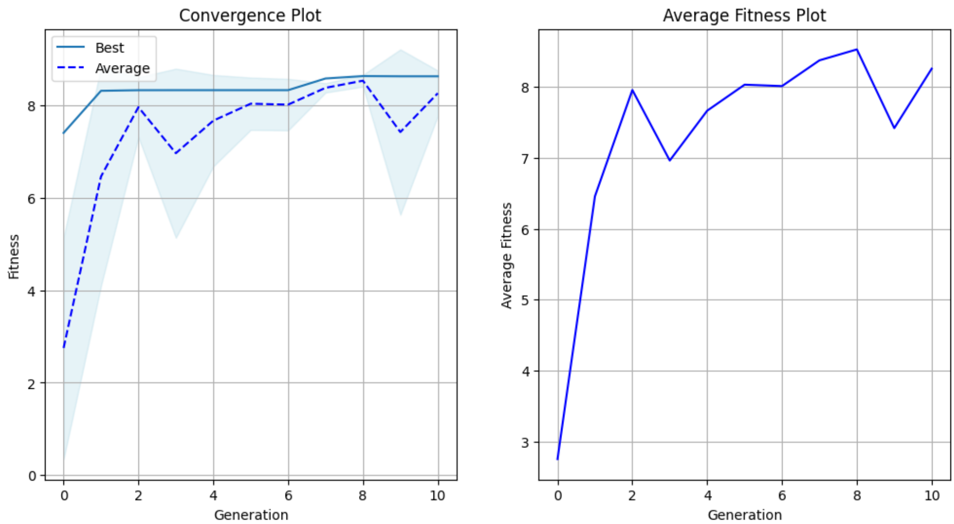

Figure 4. Convergence plot and Average Fitness Plot for traditional GA with Proposed Fitness(Left to right) .

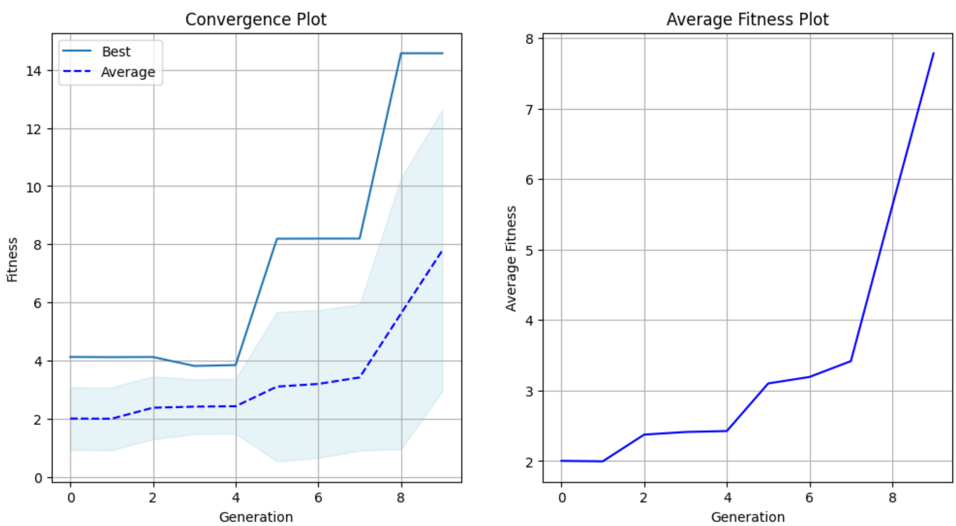

Figure 5. Convergence plot and Average Fitness Plot for Proposed DLECA (Left to right)

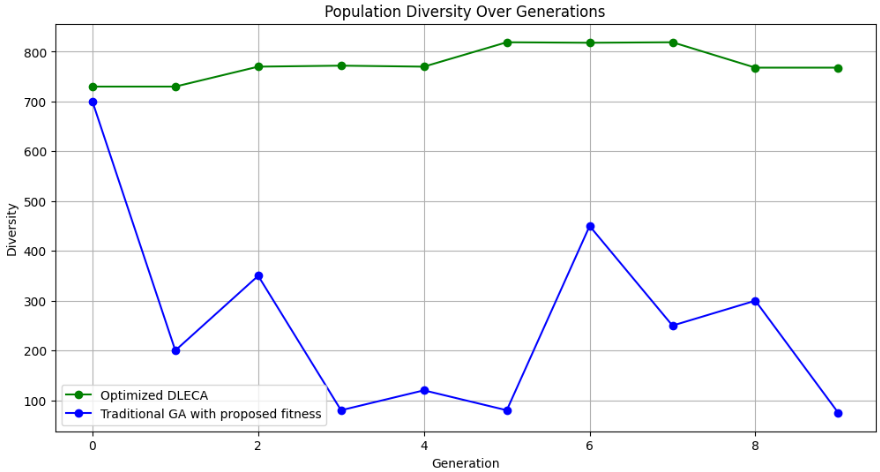

Figure 6. The Diversity Plot for the proposed approaches

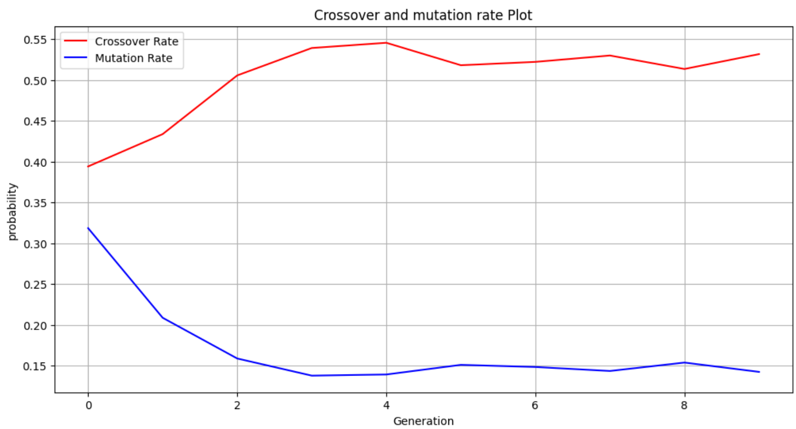

Figure 7. The crossover and mutation probability rate plot for DLECA.

Table 5 presents the optimized best solutions produced by the proposed DLECA and traditional GA using the proposed fitness function. The model1 is the best HDLM architecture given by DLECA and model 2 is best HDLM architecture given by traditional GA. The table5 details the structure of feature extractors within the HDLM, including output shapes of each layer and the number of trainable parameters for each extractor. Additionally, it summarizes the total trainable parameters in HDLM and highlights the accuracy and compression ratio achieved by the optimized architecture.

4.3.4. Result Analysis

In this section, we provide a detailed analysis of the two best HDLM architectures generated by the proposed techniques. Additionally, we compare the performance of these techniques against traditional optimization algorithms found in the literature. The optimized solutions from Table 5 are integrated into the HDLM, which is then trained and validated on the FakeNewsNet dataset. Table 6 presents the performance of the two models, evaluated in terms of accuracy, confusion matrix, precision, recall, and F1 score. The table 6 also shows the percentage improvement in accuracy and the percentage reduction in model parameters compared to the base model architecture detailed in Table 3. Equation (10) presents the formula used to calculate the percentage improvement in both model compression and accuracy.

$$\%\ improvement = \frac{value_{before}\ - Value_{after}}{|value_{before}|} \times 100 \tag{10}$$

Model	Best Individual	Architecture	output shape	Trainable parameters	Total Parameters	Accuracy	Compression ratio
1	[11,45,47,43,39,32,23,16,22,36]	lstm_layer dropout dense_1	(none,11) (none,11) (none,47)	34320 0 564	34884	91.59	46.43
		con2d_1 con2d_2 con2d_3 con2d_4 dense_2	(none,222,222,43) (none,109,109,39) (none,52,52,32) (none,24,24,23) (none,47)	1204 15132 11264 6647 155711	189958		
		dense_3 dense_4	(none,16) (none,47)	96 799	895		
		output_dimension1 dense_5	(none,22) (none,36)	2618 828	3446		
2	[25, 27, 128, 16, 16, 32, 16, 32, 16, 70]	lstm_layer dropout dense_1	(none,25) (none,25) (none,128)	79400 0 3328	82728	91.48	26.81
		con2d_1 con2d_2 con2d_3 con2d_4 dense_2	(none,222,222,16) (none,109,109,16) (none,52,52,32) (none,24,24,16) (none,128)	448 2320 4640 4624 295040	307072		
		dense_3 dense_4	(none,32) (none,128)	192 4224	4416		
		output_dimension1 dense_5	(none,16) (none,70)	4400 1190	5590		

Table 5. Best solutions provided by DLECA and Traditional GA

Model 1, created using the best configuration from the proposed DLECA, achieves a significant compression rate of 97.86%, indicating a 97.86% reduction in trainable parameters in the optimized architecture, while also slightly improving the accuracy by 0.34%. In contrast, Model 2, the best configuration from the traditional GA with the proposed fitness function, achieves a compression rate of 96.23% but provides a slightly less accuracy improvement of 0.23% and the highest precision of 90.79%.

Model	Percentage Improvement in Accuracy	Percentage reduction in model parameters	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)
1	**0.34**	**97.86%**	**91.59**	89.67	**92.94**	**91.27**
2	0.23	96.24%	91.48	**90.79**	91.24	91.01

Table 6. Performance analysis of best solutions

The confusion matrix for both the models are shown in figure 8 and figure 9. The confusion matrix is presented to provide a clearer understanding of the model's performance at the individual instance level. It is evident that the true positives (TPs) and true negatives (TNs) are significantly higher than the false negatives (FNs) and false positives (FPs), highlighting the model's high accuracy. Additionally, the model demonstrates a balanced performance, as reflected in the consistent values of TPs and TNs, indicating it is not biased towards a particular class.

FNs, which represent fake news instances misclassified as real, are minimal, resulting in a low false-negative rate and high recall. Similarly, FPs, where real news is misclassified as fake, are also low, leading to a low false-positive rate and reinforcing the model's precision.

This pattern is consistently observed in both confusion matrices, confirming that the model achieves an unbiased performance with strong precision and recall metrics.

Figure 8. Performance plots and confusion matrix for HDLM through DLECA

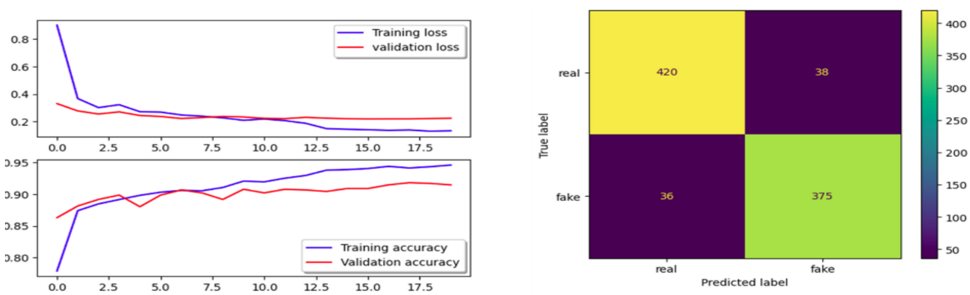

Figure 9. Performance plots and confusion matrix of traditional GA with proposed fitness

4.3.5. Comparative Analysis of proposed techniques

In this section, we present a comparative analysis of the proposed DLECA and traditional GA with the proposed fitness function against classical optimization algorithms from the literature. Additionally, we evaluate the performance of these approaches against the manual trial-and-error technique commonly used for hyperparameter selection.

1. Random Walk Optimization:

Random Walk optimization is a well-known method for hyperparameter optimization[12], particularly known for its simplicity and effectiveness in large and complex hyperparameter spaces. In this work, the algorithm starts by defining a search space like the population initialization phase of DLECA. Then an initial set of hyperparameters are chosen randomly from the search space. The performance of the HDLM is evaluated using these initial hyperparameters which will be considered as baseline performance. Random walk then iteratively introduces random perturbations to these hyperparameters, evaluating the model's performance after each perturbation. If the performance improves, the current set of hyperparameters is considered as the new baseline, and the process continues until the specified iterations is completed.

The performance of the HDLM using the best solution provided by Random Walk optimization is presented in the table7. The Random Walk algorithm achieved an F1 score of 82.33%, demonstrating its ability to find reasonable hyperparameter settings in a relatively straightforward manner.

The random search is implemented using talos library, the hyperparameter search space is same as that used for DLECA and the number of iterations was fixed to 10.

2. Bayesian optimization (BO):

Bayesian optimization is a powerful method for hyperparameter optimization. The algorithm starts by defining an objective function which in current work is same as the proposed fitness function. It then evaluates the objective function using random hyperparameter set values. Gaussian process(GP) models is then built to understand the performance over hyper parameters. Further an acquisition function would suggest new hyper parameter values to improve the performance. The process continues till maximum number of evaluations are done.

In this work, scikit-optimize library (skopt) in Python is used, the performance of the HDLM with the hyperparameter set provided by Bayesian optimization process is shown in table 7. The model achieved an F1 score of 83.76%.

DLECA outperforms other optimization methods like random walk and bayesian optimization because it dynamically adjusts the crossover and mutation probabilities based on the population's fitness. This adaptive approach enables DLECA to explore diverse solutions in the early stages and focus on refining high quality solutions later in the process. In contrast random walk explore the search space more randomly which often leading to suboptimal results while bayesian lacks flexibility to dynamically adjust based on population fitness.

Beyond fake news detection, this approach can be generalizable to any domain where deep learning models are used and lightweight architectures are needed. The hyperparameters targeted in this study like learning rate, dropout, and layer dimensions are fundamental to any deep learning model, making DLECA applicable to various multimodal tasks. In future we would like to experiment to ensure the adaptability of the algorithm to compress models in diverse use cases without compromising on accuracy.

Sr. No.	Algorithm	Accuracy	Precision	Recall	F1 Score
1	Random Walk	85.44	84.56	80.22	82.33
2	Bayesian Optimization	86.22	85.12	82.45	83.76
3	Manually Setting of hyper parameters.	91.25	89.59	92.21	90.88
2	**DLECA with fixed crossover and mutation**	**91.48**	**90.79**	**91.24**	**91.01**
3	**DLECA with adaptive crossover and mutation**	**91.59**	**89.67**	**92.94**	**91.27**

Table 7. Comparative Analysis of proposed technique against other optimization techniques

5. Conclusion

Multimodal fake news detection is a critical challenge in today's digital world, as misinformation spreads across various media formats. A common approach to tackling this issue involves designing hierarchical deep learning models that capture features from all available modalities, such as text, images, and social context. However, these resultant multimodal models tend to be highly complex, making them resource-intensive to train and deploy. In this work, we aimed to reduce the complexity of such hierarchical deep learning models using an innovative approach based on evolutionary computing.

We introduced DLECA, which employs a customized fitness function along with adaptive crossover and mutation strategies. The fitness function is specifically designed to minimize the number of trainable parameters in the hierarchical deep learning model without compromising its accuracy. The adaptive crossover and mutation strategies dynamically adjust the crossover and mutation probabilities based on the current state of the population.

The proposed DLECA achieved significant results, providing a best solution that reduced the model size by an impressive 97.86% while maintaining, and even slightly improving, model accuracy. We also tested a variation of the DLECA with fixed crossover and mutation rates, which resulted in a model compression of 96.24% with a comparable improvement in accuracy. Both approaches demonstrated a performance enhancement, with accuracy improvements of 0.34% and 0.23%, respectively.

Furthermore, the proposed DLECA is compared against other optimization techniques, including random walk, bayesian optimization, and manual hyperparameter tuning. The DLECA outperformed all these methods, highlighting its efficacy in optimizing complex models. Given the resource constraints, this study was conducted over 10 generations; however, in future will explore running the approach for more generations and leveraging multiprocessing environments to evaluate the fitness of individuals in the population simultaneously.

The proposed DLECA not only provides a powerful tool for optimizing hierarchical deep learning models in multimodal fake news detection but also offers potential for broader applications where model architecture and accuracy are critical.

Reference

[1] Shu, Kai & Sliva, Amy & Wang, Suhang & Tang, Jiliang & Liu, Huan. (2017). Fake News Detection on Social Media: A Data Mining Perspective. ACM SIGKDD Explorations Newsletter. 19. 10.1145/3137597.3137600.

[2] Verónica Pérez-Rosas, Bennett Kleinberg, Alexandra Lefevre, and Rada Mihalcea. (2018). Automatic Detection of Fake News. In Proceedings of the 27th International Conference on Computational Linguistics, 3391–3401.

[3] Shu, K., Wang, S. and Liu, H. (2018). Understanding User Profiles on Social Media for Fake News Detection. 2018 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR), Miami, FL, USA, 430-435.

[4] Galván, E., and Mooney, P. (2021). Neuroevolution in Deep Neural Networks: Current Trends and Future Challenges. in IEEE Transactions on Artificial Intelligence, 2(6), 476-493. doi: 10.1109/TAI.2021.3067574.

[5] Baldominos, Alejandro & Sáez, Yago & Isasi, Pedro. (2019). Hybridizing Evolutionary Computation and Deep Neural Networks: An Approach to Handwriting Recognition Using Committees and Transfer Learning. Complexity, 2952304. 10.1155/2019/2952304.

[6] Sun, Y., Xue, B., Zhang, M., Yen, G.G. and Lv, J. (2020). Automatically Designing CNN Architectures Using the Genetic Algorithm for Image Classification. in IEEE Transactions on Cybernetics, 50(9), 3840-3854. doi: 10.1109/TCYB.2020.2983860.

[7] Naik, P. M. and Rudra, B. (2023). Classification of Arecanut X-Ray Images for Quality Assessment Using Adaptive Genetic Algorithm and Deep Learning. in IEEE Access, 11, 127619-127636. doi: 10.1109/ACCESS.2023.3332215.

[8] Jebraeily, Yashar & Sharafi, Yousef & Teshnehlab, Mohammad. (2024). Driver Drowsiness Detection Based on Convolutional Neural Network Architecture Optimization Using Genetic Algorithm. IEEE Access, 12, 1-1. 10.1109/ACCESS.2024.3381999.

[9] Shrestha, A. and Mahmood, A. (2019). Optimizing Deep Neural Network Architecture with Enhanced Genetic Algorithm. 2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA), Boca Raton, FL, USA, 1365-1370. doi: 10.1109/ICMLA.2019.00222.

[10] Wu, D., Guan, Q., Fan, Z., Deng, H. and Wu, T. (2023). AutoML With Parallel Genetic Algorithm for Fast Hyperparameters Optimization in Efficient IoT Time Series Prediction," in IEEE Transactions on Industrial Informatics, 19(9), 9555-9564. doi: 10.1109/TII.2022.3231419.

[11] Farrag, T. A., and Elattar, E. E. (2021). Optimized Deep Stacked Long Short-Term Memory Network for Long-Term Load Forecasting. in IEEE Access, 9, 68511-68522. doi: 10.1109/ACCESS.2021.3077275.

[12] Singhal, S., Shah, R. R., Chakraborty, T., Kumaraguru, P. and Satoh, S. (2019). SpotFake: A Multi-modal Framework for Fake News Detection. 2019 IEEE Fifth International Conference on Multimedia Big Data (BigMM), Singapore, 39-47. doi: 10.1109/BigMM.2019.00-44.

[13] Navaux, Philippe, O.A. (2022). Hyperparameter optimization for convolutional neural networks with genetic algorithms and Bayesian optimization. 2022 IEEE Latin American Conference on Computational Intelligence (LA-CCI). IEEE.

[14] Ma, L., Cui, J. and Yang, B. (2019). Deep Neural Architecture Search with Deep Graph Bayesian Optimization. 2019 IEEE/WIC/ACM International Conference on Web Intelligence (WI), Thessaloniki, Greece, 500-507.

[15] Bakhshi, A., Noman, N., Chen, Z., Zamani, M. and Chalup, S. (2019). Fast Automatic Optimisation of CNN Architectures for Image Classification Using Genetic Algorithm. 2019 IEEE Congress on Evolutionary Computation (CEC), Wellington, New Zealand, 1283-1290. doi: 10.1109/CEC.2019.8790197.

[16] Mandal, R. and Verma, B. (2023). Parameter Optimisation for Context-Adaptive Deep Layered Network for Semantic Segmentation. 2023 IEEE Symposium Series on Computational Intelligence (SSCI), Mexico City, Mexico, 258-263. doi: 10.1109/SSCI52147.2023.10371960.

[17] Khan, M. S., Jabeen, F., Ghouzali, S., Rehman, Z., Naz, S. and Abdul, W. (2021). Metaheuristic Algorithms in Optimizing Deep Neural Network Model for Software Effort Estimation. in IEEE Access, 9, 60309-60327. doi: 10.1109/ACCESS.2021.3072380.

[18] Shannaq, F., Hammo, B., Faris, H. and Castillo-Valdivieso, P. A. (2022). Offensive Language Detection in Arabic Social Networks Using Evolutionary-Based Classifiers Learned From Fine-Tuned Embeddings. in IEEE Access, 10, 75018-75039. doi: 10.1109/ACCESS.2022.3190960.

[19] Choudhury, D., Acharjee, T. (2023). A novel approach to fake news detection in social networks using genetic algorithm applying machine learning classifiers. Multimed Tools Appl., 82, 9029–9045. https://doi.org/10.1007/s11042-022-12788-1.

[20] Kumar, S., Kumar, A., Mallik, A. and Singh, R. R. (2024). OptNet-Fake: Fake News Detection in Socio-Cyber Platforms Using Grasshopper Optimization and Deep Neural Network. in IEEE Transactions on Computational Social Systems, 11(4), 4965-4974. doi: 10.1109/TCSS.2023.3246479.

[21] Zaheer, H., Rehman, S.U., Bashir, M. et al. (2024). A metaheuristic based filter-wrapper approach to feature selection for fake news detection. Multimed Tools Appl. https://doi.org/10.1007/s11042-024-18734-7

[22] Uppada, S.K., Ashwin, B.S. & Sivaselvan, B. (2024). A novel evolutionary approach-based multimodal model to detect fake news in OSNs using text and metadata. J Supercomput., 80, 1522–1553. https://doi.org/10.1007/s11227-023-05531-6

[23] Shah, P. and Kobti, Z. (2020). Multimodal fake news detection using a Cultural Algorithm with situational and normative knowledge. 2020 IEEE Congress on Evolutionary Computation (CEC), Glasgow, UK, 1-7. doi: 10.1109/CEC48606.2020.9185643.

[24] Marsili-Libelli, Stefano & Alba, P. (2000). Adaptive mutation in genetic algorithms. Soft Computing, 4, 76-80. 10.1007/s005000000042.

[25] Lin, Wen-Yang & Lee, Wen-Yung & Hong, Tzung-Pei. (2003). Adapting Crossover and Mutation Rates in Genetic Algorithms. J. Inf. Sci. Eng. 19, 889-903.

[26] Shu, Kai and Mahudeswaran, Deepak and Wang, Suhang and Lee, Dongwon and Liu, Huan. (2018). FakeNewsNet: A Data Repository with News Content, Social Context and Dynamic Information for Studying Fake News on Social Media, arXiv preprint arXiv:1809.01286},2018.

[27] Qi, P., Cao, J., Yang, T., Guo, J. and Li, J. (2019). Exploiting Multi-domain Visual Information for Fake News Detection. 2019 IEEE International Conference on Data Mining (ICDM), Beijing, China, 518-527. doi: 10.1109/ICDM.2019.00062.

[28] Li, P., Sun, X., Yu, H., Tian, Y., Yao, F. and Xu, G. (2022). Entity-Oriented Multi-Modal Alignment and Fusion Network for Fake News Detection. in IEEE Transactions on Multimedia, 24, 3455-3468. doi: 10.1109/TMM.2021.3098988.

[29] Dhruv Khattar, Jaipal Singh Goud, Manish Gupta, and Vasudeva Varma. (2019). MVAE: Multimodal Variational Autoencoder for Fake News Detection. In The World Wide Web Conference (WWW '19). Association for Computing Machinery, New York, NY, USA, 2915–2921. https://doi.org/10.1145/3308558.3313552

[30] Raza, S., Ding, C. (2022). Fake news detection based on news content and social contexts: a transformer-based approach. Int J Data Sci Anal., 13, 335–362. https://doi.org/10.1007/s41060-021-00302-z.