Enhancing Fine-Tuning of Pre-trained Language Models for Sentiment Analysis Using Metaheuristic Algorithms

Abstract—Fine-tuning pre-trained language models for specific natural language processing tasks often leads to suboptimal performance due to the vast parameter space and the challenge of finding the most effective model configurations. Traditional fine-tuning methods can be computationally expensive and may not fully exploit the potential of these models, particularly for tasks like sentiment analysis. This work explores the application of metaheuristic techniques, focusing on sentiment analysis using the Stanford Sentiment Treebank 2 (SST2) dataset. We utilize two approaches: Genetic Algorithms (GA) and the Whale Optimization Algorithm (WOA). These methods aim to efficiently search the parameter space and optimize the fine-tuning process of pre-trained language models. To enhance efficiency, we incorporate layer freezing techniques along with different crossover strategies. Our experimental results show that GAoptimized and WOA-optimized models achieve higher accuracy compared to conventional fine-tuning methods such as Quantized Low-Rank Adaptation (QLoRA), with improvements of up to
1.45% in accuracy. Specifically, GA with single-point crossover
(SBX) and 30% layer freezing achieved 94.02% accuracy, while WOA with SBX crossover and 30% layer freezing achieved 94.67% accuracy. These outcomes highlight the potential of integrating evolutionary algorithms, nature-inspired optimization techniques, and strategic layer freezing with deep learning finetuning processes for NLP tasks.

Index Terms—Metaheuristic Techniques, Pre-trained Language Models, Fine-Tuning, Sentiment Analysis, Stanford Sentiment Treebank 2 (SST2), Genetic Algorithms (GA), Whale Optimization Algorithm (WOA), Layer Freezing.

I. Introduction

In recent years, there has been an increase in the advancement of Natural Language Processing, owing to the development of large language models like BERT, GPT, and RoBERTa [1]. These models have shown admirable performances in a variety of NLP tasks such as sentiment analysis, named entity recognition, and machine translation. Amongst these the task of sentimental analysis plays an important role in understanding the public opinion, customer feedback, and social media trend by classifying each sentence into positive or negative sentiments.

Fine-tuning pre-trained language models on tasks such as sentimental analysis is a standard approach to exploit the generalization capabilities. However, fine-tuning includes many model parameters and configurations that turn into a complex model, impacting performance most of the time. Traditional methods for parameter optimization are often incompetent in efficiently exploring the vast search space; hence, this results in substandard performance.

To address these challenges, bio-inspired computing algorithms have emerged as powerful optimization techniques. Genetic Algorithms (GAs)[2] and the Whale Optimization Algorithm (WOA) [3] are among the most prominent bioinspired approaches, known for their ability to efficiently navigate complex search spaces and find near-optimal solutions. These algorithms simulate natural evolutionary processes and social behaviors to iteratively improve candidate solutions based on their fitness.

The proposed approach focuses on combination of Genetic Algorithms with the Whale Optimization Algorithm for optimizing the fine-tuning process on the RoBERTa-base model to perform the task of sentiment analysis. In this approach, the Stanford Sentiment Treebank (SST-2) dataset is used, which has become a standard benchmark in binary sentiment classification tasks. The approach makes use of GAs for their evolutionary principles and WOA for the balance between exploration and exploitation in altering dynamically the model's parameter configurations, which best permit improving its performance in accomplishing the sentiment analysis task

The paper describes the SST2 dataset extensively and then proceeds to elaborate on the pre-processing steps required to prepare the data for training. It continues with describing the architecture of RoBERTa-base model and its effectiveness on tasks of sequence classification. Our main contribution lies in the integration of GAs and WOA to optimize the fine-tuning of the model including methods like dynamic layer freezing, crossover methods and mutation strategies.crossover

A series of experiments were conducted to evaluate the effectiveness of the model, comparing different configurations and optimization techniques. The results showcase the potential of bio-inspired algorithms in improving the finetuning process of large language models, leading to enhanced accuracy in sentiment classification.

By combining evolutionary algorithms and nature-inspired optimization techniques with deep learning fine-tuning processes, this research aims to contribute new methodologies to the field of natural language processing, addressing the challenges of hyperparameter optimization and enhancing model performance in sentiment analysis tasks. The computational cost associated with traditional exhaustive search methods further motivates the exploration of efficient bioinspired optimization strategies, ensuring practical

scalability and performance improvements in real-world applications.

The remainder of this paper is organized as follows: Section II provides a detailed description of the dataset and preprocessing steps. Section III outlines the model architecture and its components. Section IV introduces our genetic algorithm approach, including the specific crossover techniques employed. Section V describes the experimental setup, including the training configurations and optimization strategies. Finally, Section VI presents the results of our experiments, highlighting the performance improvements achieved through our proposed approach.

Write Motivation here

II. Related Work

Recent advancements at the intersection of Large Language Models (LLMs) and Evolutionary Algorithms (EAs) have significantly enhanced various computational processes. Wang et al. [4] explore the mutual benefits of integrating LLMs with EAs, highlighting how LLMs can optimize evolutionary processes and how EAs can refine LLM architectures. Hao et al. [5] build on this by introducing LLM-assisted Evolutionary Algorithms (LAEA), where LLMs act as surrogate models for evaluating optimization quality, improving the efficiency of evolutionary methods.

The application of LLMs to enhance evolutionary computation techniques is another important area of research. Wu et al. [3] provide a comprehensive survey and roadmap on how LLMs streamline the design of EAs and automate parameter adaptation. Xiao et al. [4] further this integration by using LLMs alongside genetic algorithms to evolve event patterns for news summary generation, demonstrating the effectiveness of LLMs in refining evolutionary designs.

In the domain of hyperparameter optimization and neural architecture search, significant improvements have been achieved through the application of EAs and LLMs. Sobhanam and Prakash [5] investigate the use of Genetic Algorithms to fine-tune hyperparameters in models like RoBERTa, leading to enhanced performance in text classification. Aszemi and Dominic [6] showcase how Genetic Algorithms can optimize hyperparameters in Convolutional Neural Networks, reducing manual tuning efforts. Additionally, Garcia-Mart´ ´ınez et al. [7] highlight the benefits of integrating EAs with LLMs for neural architecture search, which enhances model adaptability and performance.

Algorithmic innovations incorporating LLMs have also been noteworthy. Gu et al. [8] propose a semantic-based layerfreezing approach for efficient fine-tuning of LLMs, offering a more effective method compared to traditional techniques. Carmona et al. [9] explore how Genetic Algorithms can finetune transformer encoders, providing insights into improving model performance through evolutionary methods. Mirjalili and Lewis [10] introduce the Whale Optimization Algorithm (WOA), showcasing its competitive performance across various optimization problems. Aljarah et al. [11] present a new training algorithm for neural networks based on WOA, demonstrating its effectiveness in optimizing connection weights.

The integration of LLMs into metaheuristic approaches has shown promise in solving complex optimization problems. Sartori et al. [12] discuss leveraging LLMs as pattern recognition tools within metaheuristics to enhance solution quality in combinatorial optimization problems. Zhan et al. [13] investigate optimization techniques for sentiment analysis using LLMs like GPT-3, demonstrating performance improvements through fine-tuning.

Practical applications of these advancements are evident in various case studies. Malashin et al. [14] explore the classification of economic activity descriptors using NACE codes, combining machine learning with expert evaluation to improve accuracy. Lange et al. [15] introduce EvoLLM, employing LLMs as Evolution Strategies (ES) for gradientfree optimization, achieving superior results compared to traditional methods in synthetic and neuroevolution tasks. after going through the lit, the following research gaps have been identified

III. Proposed Approach

A. Dataset Description And Preprocessing

The SST-2 dataset consists of sentences extracted from movie reviews, annotated with binary sentiment labels indicating whether the sentiment expressed in the sentence is positive or negative.The dataset contains a total of 67,349 sentences for training and validation.The sentiment labels are binary, with 0 representing negative sentiment and 1 representing positive sentiment.In the training set, there are 33,642 positive examples and 33,707 negative examples, totaling 67,349 instances. The validation set consists of 444 positive and 428 negative examples, amounting to 872 instances. The test set includes 909 positive and 912 negative examples, summing up to 1,821 instances.

Data preprocessing is a crucial step in preparing the dataset. Tokenization is performed to split the sentences into individual words or tokens. In this step, punctuation marks are also separated from words to create a tokenized representation of the text. Padding[6] is applied to ensure that all input sequences have the same length. This is necessary for batch processing and efficient model training. In this step, shorter sequences are padded with special tokens such as blank characters to match the maximum sequence length in the dataset. By performing these preprocessing steps, the raw text data from the SST-2 dataset is transformed into a clean and standardized format suitable for training sentiment analysis models using deep learning techniques.

B. Model Architecture

Our solution employs RoBERTa-base Large Language Model which is built on a robust transformer architecture consisting of 12 transformer layers, also known as transformer blocks. Each of these layers processes the input data to capture various levels of abstraction. Within each layer, there is a hidden size of 768, determining the dimensionality of the output. The model uses 12 self-attention heads per layer, which enable it to focus on different parts of the input sequence when processing specific tokens, thus capturing various contextual aspects. Following the self-attention mechanism, each transformer layer has a feed-forward neural network with an intermediate size of 3072. This feed-forward network comprises two linear transformations separated by a ReLU activation function, further enhancing the model's ability to learn complex patterns from the data.The AutoModelForSequenceClassification class is used to load RoBERTa with additional configuration for sequence classification. The model is initialized with two output labels (positive and negative), and the label mappings are defined. The RoBERTa-base model is used due to its robust performance in various NLP tasks and its ability to capture rich contextual information. RoBERTabase consists of 12 layers with a hidden size of 768 and a total of 125 million parameters.

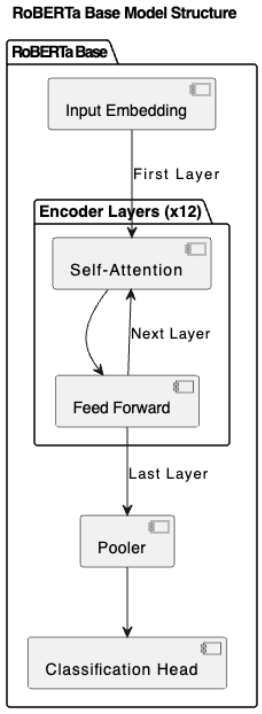

Fig. 1. RoBERTa-base model

C. Genetic Algorithm

Genetic Algorithms (GA) are optimization techniques inspired by natural selection, used to solve complex problems by iteratively improving candidate solutions. Our approach as shown in Algorithm 1. employs a Genetic Algorithm to fine-tune RoBERTa-base model on the SST-2 dataset by iteratively selecting the best-performing models, generating offspring through various crossover techniques, and refining them with mutation and dynamic layer freezing based on fitness scores to optimize accuracy. The model with the highest validation accuracy is chosen as the optimal configuration. Figure 2. represents the evolution of the initial population onto the subsequent generations of the Genetic Algorithm which eventually leads to the selection of the best individual. P1 to P4 represents the initial population while O1 to O4 represents the population of the generated offsprings. The number of individuals chosen for our experiments is 4 as shown in Figure 2. However, the initial population can contain any number of individuals.

This genetic algorithm approach leverages the principles of natural selection and evolution to find the optimal model configuration. By iteratively selecting the best individuals, performing crossover and mutation, and adjusting parameters based on fitness, the GA efficiently navigates the search space to enhance model performance.

D. Whale Optimisation

Whale Optimization Algorithm (WOA) is used to optimize the training process of a sequence classification model. The process begins with initializing a population of models, each with randomly frozen layers to reduce trainable parameters. The objective function trains each model for one epoch and evaluates its performance based on the negative F1 score. The algorithm iteratively updates the models over several generations, balancing exploration and exploitation using a parameter a that decreases linearly. Models move towards the best solution found so far or towards a randomly chosen model, simulating the encircling behavior of whales. Additionally, the bubble-net hunting strategy is mimicked by either performing shrinking encircling or spiral updating. The encircling prey mechanism involves updating the position of a whale towards the best solution found so far. The spiral updating position mechanism simulates the helix-shaped movement of whales around their prey.The above methods are the ones employed in the traditional approach of the WOA.

Our appraoch varies slightly from the traditional one where in it consists of introducing crossover mechanism along with the traditional Algorithm 1 Genetic Algorithm for Fine-Tuning RoBERTa on SST-2

1: Initialize the population with K copies of the pre-trained RoBERTa-base models each with different set of frozen layers
2: for each model in the population do
3: Fine-tune the model on the SST-2 dataset
4: Measure accuracy on the validation set
5: Define fitness as the negative of the model's accuracy
6: end for
7: for generation = 1 to N do
8: Select the two models with the best fitness scores as parents
9: for each crossover technique in {SBX, Single Point, Arithmetic, Two Point} do
10: Generate new offspring from the selected parents using the current crossover technique
11: end for
12: Introduce small random changes to the parameters of the offspring through mutation
13: for each offspring do
14: Fine-tune the offspring on the SST-2 dataset
15: Measure accuracy on the validation set
16: Define fitness as the negative of the model's accuracy
17: end for
18: Update the population with the new individuals based on their fitness scores
19: Perform relative freezing of the offsprings with highest fitness model getting least number of layers frozen
20: end for
21: Select the model with the highest accuracy on the validation set as the optimal configuration

approach of WOA to ensure variability in the models. After each generation of spiral updation crossover mechanism is applied between two individuals and offsprings are produced. Algorithm 2 starts by initializing a population of K pretrained RoBERTa models, each with different frozen layers, and evaluates their initial fitness based on validation accuracy. Over N generations, the algorithm dynamically adjusts the positions of individuals towards the best solution or random individuals using parameters A and C, derived from random numbers. Individuals are also updated using a spiral equation when necessary. After updating positions, crossover techniques (SBX, Single Point, Arithmetic, Two Point) generate offsprings, which are then fine-tuned and evaluated.

E. Crossover techniques

Crossover is a crucial genetic operator in genetic algorithms (GAs) that combines the genetic information of two parent solutions to generate new offspring. Below are some of the traditionally used crossover techniques: Single Point Crossover, Arithmetic Crossover, Simulated Binary Crossover(SBX) and Two Point Crossover.

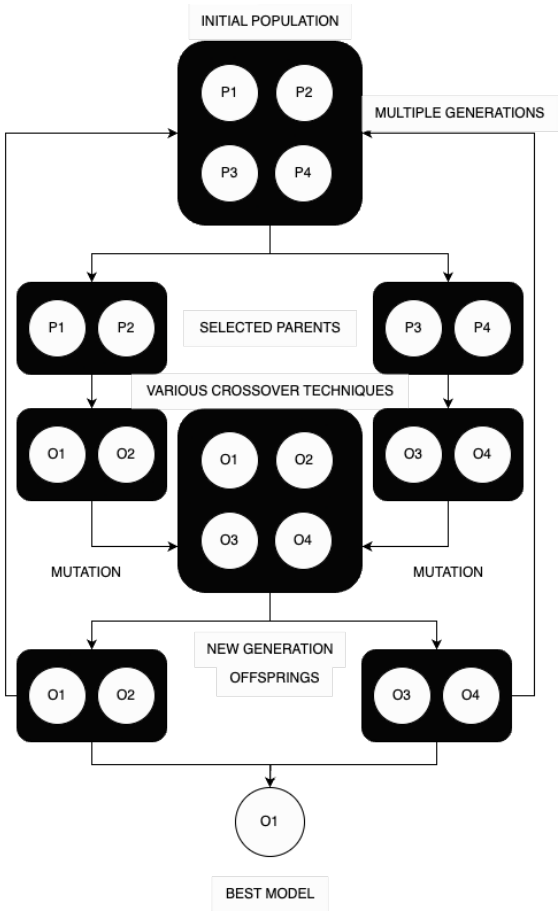

Fig. 2. Genetic algorithm

Single Point Crossover is one of the simplest crossover techniques. In this method, a single crossover point is selected randomly on the parent parameters. The genetic material from the beginning to the crossover point is taken from one parent, and the rest is taken from the other parent.

$$\Theta_1 = [P_1[1:cp] \oplus P_2[cp+1:n]]$$
$$\Theta_2 = [P_2[1:cp] \oplus P_1[cp+1:n]] \quad (1)$$

Where P_1 and P_2 represent the parents and Θ_1 and Θ_2 are the resulting offspring. cp represents the crossover point

In the Arithmetic Crossover technique, offspring are generated by combining corresponding genes from two parents using a weighted average. It is particularly suited for

problems where the values of genes represent real numbers or continuous variables.

$$\Theta_1 = \alpha P_1 + (1 - \alpha) P_2$$
$$\Theta_2 = (1 - \alpha) P_1 + \alpha P_2 \quad (2)$$

Where α is a random weight usually between 0 and 1. P_1 and P_2 represent the parents and Θ_1 and Θ_2 are the resulting offspring.

Two-point crossover selects two random points on the parent chromosomes and then exchanges the genetic material between Algorithm 2 Whale Optimization Algorithm for Fine-Tuning RoBERTa on SST-2 with Crossover and Dynamic Freezing

1: Initialize population with K pre-trained RoBERTa models with different frozen layers.
2: for each individual do
3: Fine-tune on SST-2 and measure validation accuracy.
4: Define fitness as the negative of accuracy.
5: end for
6: Initialize best individual and best fitness.
7: for generation = 1 to N do
8: Calculate a as $a_{\max} - \frac{}{generation N \times a_{\max}}$.
9: for each individual do
10: Generate random numbers $r1$ and $r2$.
11: Calculate $A = 2 \times a \times r1 - a$ and $C = 2 \times r2$.
12: if random < 0.5 then
13: if $|A| < 1$ then
14: Update position towards the best individual.
15: else
16: Update position towards a random individual.
17: end if
18: else
19: Update position using the spiral equation.
20: end if
21: end for
22: Perform crossover:
23: for each pair of individuals do
24: for each crossover technique in {SBX, Single Point, Arithmetic, Two Point} do 25: Generate and fine-tune offspring.
26: Evaluate offspring and define fitness.
27: end for
28: end for
29: Update population and best individual based on fitness.
30: Apply Dynamic Freezing:
31: for each individual in population do
32: Determine the proportion of layers to freeze based on fitness.
33: Randomly freeze the determined proportion of layers.
34: end for
35: end for
36: Select the best individual as the optimal configuration.

these points to create offspring. This method allows for more diversity in the offspring compared to single-point crossover, as it can disrupt fewer building blocks of good solutions.

$$\Theta_1 = [P_1[1 : x_1] \oplus P_2[x_1 : x_2] \oplus P_1[x_2 : end]]$$
$$\Theta_2 = [P_2[1 : x_1] \oplus P_1[x_1 : x_2] \oplus P_2[x_2 : end]] \quad (3)$$

Where x_1 and x_2 are the crossover points, $\oplus$ denotes concatenation, P_1 and P_2 represent the parents and Θ_1 and Θ_2 are the resulting offspring.

SBX (Simulated Binary Crossover) is a genetic algorithm operator designed for real-valued optimization problems, mimicking the binary crossover used in binary-coded genetic algorithms. In SBX, two parent solutions are selected from the population, and for each dimension, a random number is generated uniformly from [0, 1]. This random number determines the calculation of a distribution parameter.

$$c_{1,k} = 0.5[(1 + \beta_k) p_{1,k} + (1 - \beta_k) p_{2,k}]$$
$$c_{2,k} = 0.5[(1 - \beta_k) p_{1,k} + (1 + \beta_k) p_{2,k}] \quad (4)$$

where $c_{1,k}$ and $c_{2,k}$ are the k-th components of the offspring, $p_{1,k}$ and $p_{2,k}$ are the k-th components of the parents, and β_k is a random variable with the density:

$$P(\beta) = \begin{cases} 0.5(\eta_c + 1)\beta^{\eta_c}, & \text{if } 0 \le \beta \le 1 \\ 0.5(\eta_c + 1)\frac{1}{\beta^{\eta_c + 2}}, & \text{if } \beta > 1 \end{cases} \quad (5)$$

where η_c is the distribution index.

F. Freezing technique

The solution proposes an approach to freeze the parameters of the model dynamically through generations based on the fitness of the individual

The random freezing function is used to randomly select and freeze a subset of layers in a neural network model, preventing their parameters from being updated during training. It first creates a list of all layers, excluding the classifier, to ensure the classifier remains trainable. It then calculates the number of layers to freeze, of the total layers. Using random.sample, it randomly selects the layers to freeze and sets their requires grad attribute to False, leaving the rest as trainable. This function helps introduce diversity and reduces training complexity, with the names of the frozen layers printed for transparency.

The adjust freezing function dynamically adjusts the proportion of frozen layers in each model based on their fitness, promoting an adaptive freezing mechanism. It identifies the best and worst fitness values in the population and calculates a freezing proportion for each model, ranging from 30% to 50% based on its relative fitness. Where the best model will have least layers frozen while the worst performing model will have the most number of frozen layers. The best performing models will have less layers frozen while the poorly performing models will have a higher amount of frozen layers. This is done to ensure the selection of the best models and allow for the best models to improve and evolve more efficiently. For example if a model has 10 layers the best performing model will have only 3 layers frozen while the worst performing model will have 5 layers frozen while any intermediately performing model will have between 3 to 5 layers frozen proportionately. This adaptive approach helps balance exploration and exploitation, improving the efficiency and performance of the genetic algorithm.

$$p_i = 0.3 + \frac{(f_i - f_{\text{best}})}{(f_{\text{worst}} - f_{\text{best}})} \cdot 0.2 \tag{6}$$

where p_i is the percentage of layers frozen while f_i is the fitness index of the individual.

IV. Experimental Setup

The experiment leverages the SST-2 (Stanford Sentiment Treebank) dataset, part of the GLUE benchmark suite. The dataset contains sentences extracted from movie reviews, along with their respective binary sentiment labels: positive or negative. The pre-trained large language model used in this experiment is RoBERTa-base, that is a robust transformer model known for performing well in various Natural Language Processing tasks. The model is fine-tuned for a binary sentiment classification task, with two output labels: ”Positive” and ”Negative”. The training setup is aimed at optimizing the model using a genetic algorithm. The genetic algorithm begins with choosing an initial population which is a set of pre-trained models, each with randomly frozen layers to introduce variability. The genetic algorithm involves the evolution of the population over several generations, while using selection, crossover, and mutation to iteratively improve the model performance. The experiments are conducted using a CUDA-enabled L4 GPU in order to accelerate the training and evaluation processes.

The training process of the RoBERTa-based LLM involved configuring several parameters to optimize its performance for the sentiment analysis task on the SST-2 dataset. The learning rate was set at 0.00002, and both the training and evaluation batch sizes were set to 128 to handle larger data batches efficiently. The training was conducted for only one epoch to speed up training the process, and a weight decay of 0.01 was applied to prevent overfitting. Evaluation and logging were performed every 500 steps to monitor progress and save the model. Mixed precision training was enabled to accelerate the training process and reduce memory usage. Early stopping was also implemented with a patience of two evaluation steps to stop training if no improvement was observed. This configuration aimed to balance between training efficiency and model performance.

A. Genetic Algorithm

A Genetic Algorithm (GA) was employed to optimize the layers of the RoBERTa-base LLM. The GA was designed with a population size of 4 with 2 generations, and a mutation rate of 0.1 was used. Crossover methods such as Simulated Binary Crossover (SBX), uniform crossover, arithmetic crossover, and single-point crossover were explored to generate offspring. The fitness of each model, defined as the negative accuracy on the validation set, guided the selection of parents for the next generation. The two individuals with the highest fitness were chosen as parents for the next generation.

B. Whale Optimisation

The Whale Optimization Algorithm (WOA) is also used to fine-tune a RoBERTa model. The optimization process begins with the initialization of a population of models. During each generation, models are refined using WOA’s encircling prey and bubble-net feeding methods. The encircling prey method involves adjusting model parameters towards the bestknown solution, whereas the bubble-net method modifies the parameters using exponential and trigonometric functions. The parameters r1 and r2 are randomly generated values between 0 and 1, used to introduce variability into the search process across generations. These values dictate the stochastic behavior of the algorithm, influencing decisions such as whether to explore the current best solution or a randomly selected individual from the population. The parameter A is computed based on these random values and a linearly decreasing coefficient a, which scales its impact on the adjustment of individual solutions. C is derived directly from r2 and serves to modulate the magnitude of adjustments made based on the chosen direction (either towards the best solution or a random individual). Then the crossover mechanisms were introduced to further improve the variablilty of the models, Testing with all the crossover mechanisms applied in the GA. Together, these parameters dynamically influence the exploration and exploitation phases of the algorithm, guiding how each individual in the population adapts and improves over successive generations of the optimization process.

The freezing algorithm implemented in the model in this experiment initially employed 30% frozen layers and subsequently applied 50% and 70% respectively. After the model has evaluated the first generation of parents a secondary freezing algorithm is applied which freezes between 30% to 50% of the layers based on their fitness.The best

performing models will have less layers frozen while the poorly performing models will have a higher amount of frozen layers as mentioned in section III.

V. Results

The experimental results of our approach to applying metaheuristic algorithms to fine-tune RoBERTa-base for sequence classification are presented here. The performance of the Whale Optimization Algorithm (WOA) and Genetic Algorithm (GA) for the fine-tuning task is presented in this section. We investigated the effects of varying the percentage of frozen layers and different crossover methods within the GA framework.

A. Performance Metrics

To evaluate the effectiveness of our approach, we measured the accuracy, precision, recall and F1 score of the model across various configurations. Table 1 summarizes our key findings, presenting the accuracy scores for different combinations of layer freezing percentages and crossover methods.

Table 1 shows the results of the experiments performed on the Genetic Algorithm. The 'Percent' column represents the initial percentage of frozen RoBERTa-base layers during the initial generation of fine-tuning. This enables us to investigate the trade-off between computational efficiency and model adaptability. The 'crossover' column specifies the genetic algorithm crossover method used, which includes the following crossover methods - Simulated Binary Crossover (SBX), Single Point, Two Point, and Arithmetic Crossover. Table 2 has a similar structure to Table 1 and shows the results of the Whale Optimisation Algorithm.

Crossover	Percent	Accuracy	F1	Precision	Recall
SBX	30%	94.02	94.1	94.56	93.92
	50%	93.67	93.2	92.11	94.34
	70%	92.6	92.8	92.4	93.2
Single Point	30%	93.69	93.85	93.12	94.59
	50%	93.00	93.16	92.65	93.69
	70%	92.7	93.12	92.83	93.81
Two Point	30%	93.69	93.85	93.12	94.59
	50%	92.88	92.91	92.93	94.06
	70%	93.03	93.08	92.25	93.91
Arithmetic	30%	93.66	93.82	93.52	94.11
	50%	93.11	93.33	93.13	93.93
	70%	92.65	92.84	91.22	93.81

TABLE I
GENETIC ALGORITHM

Crossover	Percent	Accuracy	F1	Precision	Recall
SBX	30%	94.67	94.6	95.21	94.14
	50%	94.12	94.21	93.69	93.88
	70%	93.92	94.06	93.54	94.59
Single Point	30%	93.34	93.58	91.77	95.49
	50%	93.80	93.90	94.11	93.69
	70%	94.15	94.26	94.15	94.36
Two Point	30%	94.26	94.40	93.77	95.04
	50%	93.57	93.76	92.73	94.81
	70%	93.84	93.54	93.54	94.59
Arithmetic	30%	93.96	94.02	94.13	93.91
	50%	93.46	93.64	92.71	94.17
	70%	93.57	93.72	93.30	94.14

TABLE II
WHALE OPTIMISATION

For GA, the SBX crossover method with a 30% initial freezing displayed the highest accuracy of 94.02%, while the same method achieved the highest F1 score of 94.1%.For WOA,the traditional approach gave an accuracy of 93.8%.While the approach with the SBX crossover with 30% freezing displayed the highest accuracy of 94.67%, and the highest F1 score of 94.6%.For the Genetic Algorithm, the lowest accuracy of 92.60% was observed with SBX crossover at 70% freezing, and the lowest F1 score of 92.25% was observed with Arithmetic crossover at the same freezing percentage. For the Whale Optimization Algorithm(WOA), the lowest accuracy was 93.34% with Single Point crossover at 30% freezing, while the lowest F1 score was 93.54% using Two Point crossover at 70% freezing.The results obtained were similar to that of the traditional approach for the WOA and no significant improvements were observed.

The results obtained from the performed experiments show that out of all crossovers the SBX crossover method performed the best in both the algorthims by consistently producing reliable results. Out of all the freezing percentage variations 30% was proven to be more effective which resulted in a more adaptable model.One anomalous observation was observed where the 50% and 70% freezing performed better than the 30% in the WOA with Single Point crossover. Overall we observe that WOA performed slightly better than GA by achieving 0.65% higher accuracy and 0.5% higher F1 score.

VI. Conclusion

In conclusion the proposed approach involving the use of Genetic Algorithm and Whale Optimisation Algorithm to fine

tune the RoBERTa-base LLM produced commendable results. The achieved results are comparable to traditional approaches to fine tuning LLMs. The freezing strategy was critical in both optimization algorithms ensuring faster training times. The dynamic layer freezing strategy allowed the best performing models to remain more adaptable by freezing fewer layers through generations. Achieving an accuracy of 94.67% in the Whale Optimisation Algorithm and 94.02% in the Genetic Algorithm highlights the effectiveness of metaheuristic algorithms for fine tuning tasks.

Although the proposed algorithms provide significant insights into various ways of fine tuning LLMs, there is scope for further advancements in the same. Various other metaheuristic algorithms could be used to tailor to specifc tasks. Additionally, developing adaptive learning rate strategies and employing ensemble methods to combine multiple optimized models could enhance accuracy and generalization. Exploring the scalability of these approaches across diverse and multilingual datasets will be crucial, as will the application of transfer learning to quickly adapt fine-tuned models to related tasks with minimal retraining. By continuing to innovate in these areas, the potential of meta-heuristic algorithms in optimizing LLMs for a broad spectrum of NLP applications can be fully realized.

References