Journal of Information Systems and Telecommunication
Vol.14, No.2, April-June 2026, 99-116
http://jist.acecr.org
ISSN 2322-1437 / EISSN:2345-2773

Optimization of Hyperparameters of BERT Model in Sentiment Analysis Using Genetic Algorithm

Shahla Sadeghani [1], Mohammad Jafar Tarokh [2*] , Mohammad Ali Afshar Kazemi [3]

[1]. Department of Information Technology Management, SR.C.,Islamic Azad University, Tehran, Iran
[2]. Department of Industrial Engineering, K.N.,Toosi University of Technology, Tehran, Iran
[3]. Department of Industrial Management, CT.C., Islamic Azad University, Tehran, Iran

Received: 19 Sep 2025/ Revised: 04 Jun 2025/ Accepted: 20 Jul 2026

Abstract

Sentiment analysis, a vital branch of natural language processing (NLP), is progressing rapidly, and advanced models such as BERT are used to improve the accuracy of such analyses. However, tuning BERT's hyperparameters remains a major challenge, directly influencing its performance. The methods used in previous research to optimize hyperparameters often face limitations in accuracy and efficiency. This study addresses an important research knowledge gap by using genetic algorithm (GA) to present a new approach to optimize the hyperparameters of BERT model in sentiment analysis with the main goal to get better results. In this regard, this research is designed analytically and experimentally. Data were collected through Twitter API and preprocessed using NLP techniques including noise removal, tokenization, and text normalization before being analyzed. The BERT model's hyperparameter optimization was performed using GA and the optimized models were evaluated based on criteria such as accuracy, precision, recall and F-1 score. The proposed algorithm was compared with other common methods such as vanilla BERT, Long Short-Term Memory (LSTM) and convolutional neural network (CNN). The results showed that the proposed model has significantly better performance than other methods, achieving 98.1% accuracy, 98.2% precision, 98.1% recall, 98.15% F-1 score, and an area under the curve (AUC) of 98.5%. In comparison, vanilla BERT model, LSTM and CNN scored 92.8%, 91.3% and 90.5% in accuracy, respectively. These results indicate that the use of GA in optimizing the hyperparameters of BERT model can effectively increase the accuracy and efficiency of sentiment analysis. This approach not only has applicability in various fields of text data analysis, but also can be used as an effective solution for future research in optimizing deep learning models.

Keywords: Sentiment Analysis; BERT; Genetic Algorithm; Hyperparameter Optimization.

1- Introduction

Sentiment analysis has attracted the attention of many researchers and organizations in recent years as a tool for understanding public sentiment and opinions [1], which provides the ability to extract feelings and emotions from texts. This ability is especially important in social platforms such as Twitter, which publish millions of comments and views daily [2]. With the increase in the extent of text data, the use of more advanced models such as BERT for sentiment analysis has become a necessity [3]. BERT is one of the most advanced NLP models [4], which is able to significantly increase the accuracy of sentiment analysis by using a Bidirectional Pre-Training Approach [3]. Fine-tuning the hyperparameters of this model can significantly affect its final performance [4]. However, optimizing the hyperparameters of this model to achieve the best performance is a big challenge [3]. The potential consequences of not addressing this issue include reduced accuracy in sentiment analysis, higher computational resource consumption, and the inability to improve existing models. This problem can have a negative impact, especially in industrial and commercial applications where model accuracy and efficiency are of great importance. Hence, there is a strong need for a novel and efficient method for automatically optimizing hyperparameters of complex models such as BERT.

Most previous research has focused on manual optimization of hyperparameters, which is a time-consuming and computationally intensive process. In addition, conventional optimization methods such as grid search and random

✉ **Mohammad Jafar Tarokh**
mjtarokh@kntu.ac.ir

search are not efficient for complex models because they cannot automatically find optimal parameter combinations in a large search space [5]. This shortcoming not only leads to a decrease in the accuracy and performance of the models, but also inefficient optimization leads to a greater consumption of time and resources. Therefore, there is a major gap in the current research field to develop more efficient and automated optimization approaches for the hyperparameters of BERT model.

In recent years, researchers have been looking for new methods for optimizing hyperparameters. One of these methods is the use of GA, which use Darwinian Evolutionary Principles to find the best hyperparameter settings. Studies such as the research conducted by Jones et al. [6] have shown that the use of GA can effectively improve the accuracy of deep learning models. In addition, another study points to the successful application of GA in optimizing NLP models [7]. These studies indicate the importance of using advanced optimization methods to achieve higher accuracy and efficiency in complex models such as BERT.

This research seeks to solve the problem of optimizing the hyperparameters of BERT model in sentiment analysis by using GA. The main objective of this study is to present an approach based on GA to optimize the hyperparameters of BERT model and improve its performance in sentiment analysis. The main goal of the present research is to create a method that would be capable of enhancing the performance of machine learning models through automatic hyperparameter selection and the more precise optimization of the chosen ones. The current paper suggests the use of a Genetic Algorithms (GA) to face the optimization of BERT hyperparameters in sentiment analysis. The primary aim of this investigation is to develop an approach based on GA that can maximize the potential of BERT in the sentiment analysis task through hyperparameter tuning and consequently achieve better performance. The proposed approach aims to improve model performance by enabling automatic selection and more precise optimization of the model hyperparameters. In order to provide proof of the proposed method's effectiveness, we will be comparing the GA-optimized BERT model's performance to that of the baseline techniques that are often used, such as vanilla BERT (standard pre-trained model) [8], Convolutional Neural Networks (CNNs) [8], and Long Short-Term Memory networks (LSTMs) [9]. These kinds of comparisons make it possible to form an overall assessment of the strengths and weaknesses of each approach to the methods. For other deep learning models in hyperparameter optimization and help address existing knowledge gaps in this domain.

This paper is structured as follows: the "Background and Description" section provides basic information about deep learning models and GA optimization methods for BERT model. The "Literature Review" section surveys previous research in the field of sentiment analysis and hyperparameter optimization methods, providing readers with theoretical foundations and insights from existing work. Then, the "Proposed System" section provides details about the proposed research framework, methods, datasets, and how to implement the optimized model. In the "Results Analysis" section, the empirical findings of the research are analyzed and evaluated, and a comprehensive comparison between the proposed approach and existing methods is made. Finally, the paper ends with the "Conclusion" section, where the main implications of the research are discussed. This structure helps the reader to move steadily from theoretical foundations to empirical findings and practical applications.

2- Background and Explanation

2-1- BERT Algorithm

BERT is a deep learning model introduced by Google researchers in 2018 [10]. It is based on a transformer architecture and is used for NLP. Unlike previous unidirectional models, BERT's main goal is to develop language models that understand the meaning of words in the context of sentences in a bidirectional manner [11]. Before BERT, models such as Word2Vec and GloVe for words representation provided fixed representations of words without regard to the overall meaning of the sentence [12]. These models were unable to understand complex semantic dependencies. Models such as ChatGPT, although they made significant improvements, still only analyzed dependencies in one direction (left to right) [13]. BERT overcomes these limitations by using a bidirectional approach. The transformative architecture that underlies BERT consists of two main parts: an encoder and a decoder. The encoder was used to analyze the inputs and generate semantic models. This part consists of multiple layers of attention and feedforward neural networks, whose self-attention mechanism allows the model to evaluate the relationship of each word to other words in the sentence and to detect long-range semantic dependencies [10]. Self-attention uses three main components: query(Q), key(K), and value(V), which weight words based on their importance in the text.

BERT uses a set of encoder layers that include tokens, positional embeddings, and representations of input segment embedding. Its architecture has two main versions, BERT-base and BER-large, which have 12 and 24 encoder layers, respectively. BERT’s pre-training uses two main tasks: Masked Language Modeling (MLM) and Next Sentence Prediction (NSP) [8]. In MLM, 15% of the tokens are masked and the model tries to predict them, which enhances the text representations. In NSP, BERT

predicts logical relationships between sentences, which helps to understand inter-sentence dependencies.

BERT is used in many NLP tasks, including sentiment analysis, question answering, machine translation, nominal entity recognition, and text classification. The advantages of BERT include bidirectional representation, portability, and high performance in various tasks. However, its limitations include the need for high computational resources and sensitivity to tokenizers [10], [12]. Advanced research based on BERT includes models such as RoBERTa to improve pre-training methods and an optimized version of the model (DistilBERT) as a lighter version with a reduced number of parameters (ALBERT). BERT is a revolutionary model in NLP that has achieved impressive results in many NLP tasks by using bidirectional word representation and a transformer architecture, and has paved the way for the development of other advanced models. The architecture of BERT algorithm is shown in Figure 1.

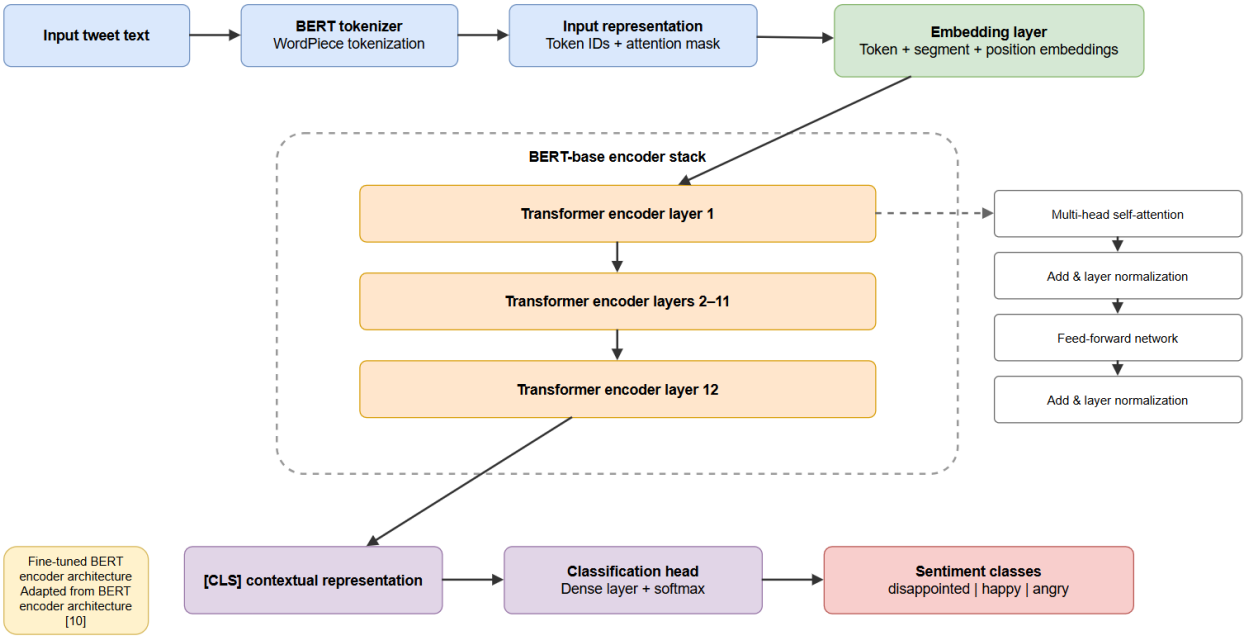

Fig. 1. Architecture of BERT algorithm [10]

2-2- Hyperparameters and the Importance of Optimization

Hyperparameters are variables that are set before the model training process begins and directly affect the model performance [14]. Examples of hyperparameters include the number of epochs, learning rate, and optimizer type. Unlike the internal weights of the network, which are optimized during the training process, hyperparameters are tuned manually or using specific optimization methods. Optimizing these variables is of particular importance, because their incorrect tuning can lead to reduced accuracy or instability in the learning process. Using intelligent methods for optimization, such as genetic algorithms, can significantly improve model performance [19].

For example, an appropriate learning rate can increase the convergence speed of the model, while an excessively high learning rate may lead to model instability. Also, choosing the appropriate number of epochs ensures that the model is well trained and prevents overfitting.

2-2-1- Hyperparameters of BERT Model

In BERT model optimization process, three key hyperparameters are set, each of which has a direct impact on the final performance of the model. Table 1 shows the hyperparameters of BERT model and their descriptions.

Table 1: Hyperparameters of BERT model

Hyperparameters	Description
Number of Epochs	Number of epochs refers to the total times a model is fully trained on the training dataset. Selecting an appropriate number of epochs enables the model to learn effectively while avoiding overfitting
Learning rate	Learning rate determines the size of weight updates in each training step. Properly tuning this parameter can improve model convergence rate
Optimizer type	Optimizer type refers to the optimization algorithm used to update model weights, such as AdamW or SGD. This parameter influences both the learning efficiency and performance of the model

The choice of these three hyperparameters (epoch, learning rate, and optimizer type) was primarily influenced by their impact on the performance of BERT during fine-tuning for sentiment analysis reported in the literatures. Along with other hyperparameters such as batch size, dropout rate, and number of hidden layers which definitely play a role, it has been confirmed through research that these three parameters significantly influence the model convergence and final performance [5] [4].The batch size was consistently set to 16 because it is considered standard when fine-tuning BERT on datasets that are about the same size [10]. Dropout rate (0.1) and the number of hidden layers (12 for BERT-base) are the default settings that have been successful in various NLP tasks and thus were not changed [10]. The optimization using a genetic algorithm was therefore carried out on the most influential hyperparameters in order to achieve a trade-off between computational efficiency and performance gains.

2-3- Genetic Algorithm

2-3-1- Introduction to Genetic Algorithm

The genetic algorithm is a method for optimization based on an evolution-like process, which features nature's way of selection, crossover, and mutation, extensively used to tackle complicated problems with a great deal of solutions to be searched through [15][14]. The algorithm starts by generating a population of random chromosomes (candidate solutions) and then each chromosome gets evaluated using a fitness function. The fittest chromosomes are then chosen to pass on their genes to the next generation as well as to produce new offspring using combinations of crossover and mutation [17], [18]. This loop of performing the same task repetitively continues until a terminating condition of either a predetermined number of generations reached or the optimum solution obtained is satisfied. In this study, GA is used to optimize the three most important parameters of the BERT model in evaluating sentiments: number of epochs, learning rate, and optimizer type. Through this technique, the importance of genetic algorithms in the optimization of the most crucial parameters becomes clear due to their ability to efficiently search through very large search spaces. The GA flowchart is shown in Figure 2.

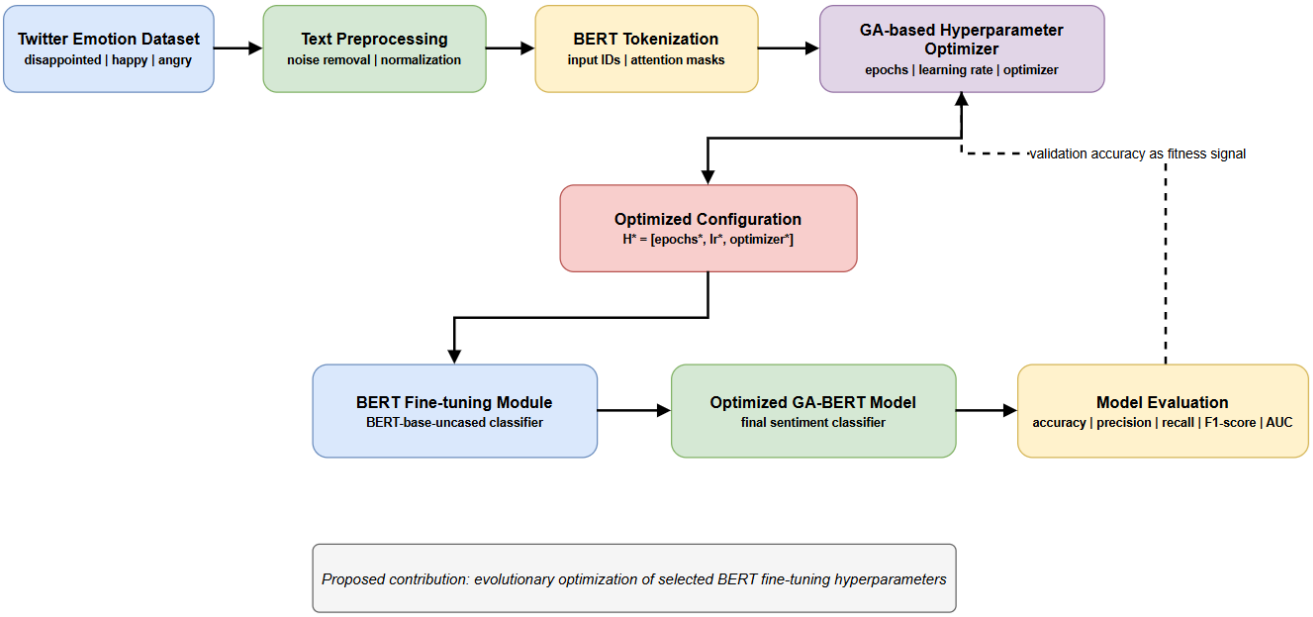

Fig. 2 Flowchart of the Genetic Algorithm [14]

2-3-2- Genetic Algorithm Parameters

he genetic algorithm adopts a predefined set of parameters, with each of them being crucial to the step of searching and optimizing. The important parameters of the GA are described in Table 2.

Table 2: Key Parameters of the Genetic Algorithm

parameters	Description
Population size	The number of individuals per generation in the algorithm, representing the number of hyperparameters combinations evaluated in each generation
Crossover rate	The probability of combining two individuals (solutions) to create a new individual inheriting features from its parents
Mutation rate	The probability of altering one or more genes (hyperparameters) in an individual to sustain diversity and prevent local optima convergence
Number of Generations	Number of iterations to produce new generations and assess hyperparameter combinations

The size of the population is what determines the number of different combinations of hyperparameters being assessed in every generation. Crossover and mutation rates control the frequency of generating new combinations through parental recombination and random alterations, respectively. The total generations represent the entire search process formed by the closure or best discovery found in the search iterations. The particular values for these parameters applied in the suggested model are stated in the Proposed Model section.

3- Literature Review

Over the past few years, sentiment analysis has become a widely recognized and frequently used method for public opinion and sentiment detection, especially through the medium of social media where Twitter posts alone are in millions every day[1][2]. With the advancement of deep learning models, the use of advanced architectures such as BERT for sentiment analysis has grown considerably due to their high accuracy in processing large-scale data [19].
This literature review examines previous studies on optimizing BERT models for sentiment analysis to provide a comprehensive view of existing research, identify knowledge gaps, and justify the importance of the present study. Recent research on BERT-based sentiment analysis can be categorized into three main approaches: hybrid model architectures, preprocessing optimization, and domain-specific adaptations. A detailed comparison of these studies is presented in Table 3, which summarizes their objectives, methodologies, key findings, strengths, and limitations. Hybrid Model Architectures Several studies have explored combining BERT with other deep learning architectures to enhance sentiment analysis performance. Bello et al. [8] integrated BERT with CNNs, RNNs, and Bi-LSTM, achieving 93% accuracy and 95% F1-score across six datasets, demonstrating that architectural combinations can outperform word-to-vector conversion methods. Similarly, Talaat [20] developed a hybrid system combining BiGRU and BiLSTM layers with RoBERTa and DistilBERT, showing improved accuracy through effective preprocessing and layer integration. Sirisha and Chandana [9] proposed a RoBERTa-LSTM hybrid model for aspect-based sentiment and emotion analysis, achieving 94.7% accuracy on Ukraine-Russia war tweets. While these hybrid approaches demonstrated performance improvements, they did not comprehensively investigate hyperparameter optimization, which represents a significant research gap. Preprocessing and Multilingual Optimization Other research has focused on optimizing preprocessing techniques for BERT models. Pota et al. [21] conducted a comprehensive evaluation of preprocessing methods for multilingual BERT models using Twitter datasets in English, Italian, and Spanish, demonstrating that appropriate preprocessing can significantly improve model accuracy. However, this study did not address hyperparameter optimization or examine combinations of different preprocessing methods, limiting its scope for performance enhancement. Domain-Specific BERT Applications Research has also adapted BERT for specific domains and applications. Lin and Moh [22] presented Covid-Twitter-BERT using the auxiliary sentence technique for COVID-19 tweet analytics, thereby enhancing accuracy through the modification of the sentence pair. Palani et al. [23] developed T-BERT, combining Latent Dirichlet Allocation with BERT for microblog sentiment analysis, achieving 90.81% accuracy. Chiorrini et al. [24] applied BERT to sentiment analysis and emotion detection in tweets, with their accuracies being 92% and 90% respectively. Anggrainingsih et al. [25]did rumor detection on Twitter using BERT and showed the system's ability to perform automated feature extraction. Wu et al. [26] in their experiment ranking BERT through optimization techniques mainly for sentiment analysis reported that although the optimization strategies achieved promising results after fine-tuning, the cost in terms of computational resources was substantial. Research Gaps and Study Justification the gaps revealed by the reviewed literature are presented and summarized in Table 3. The systematic hyperparameter optimization through evolutionary algorithms is an area that has been very slightly touched in the past while the whole research trend has focused on combining BERT with other methods and looking at preprocessing techniques. Most studies relied on default hyperparameter settings or manual tuning, which restricts model performance potential. Furthermore, none of the reviewed studies employed genetic algorithms for automated hyperparameter optimization of BERT models in sentiment analysis tasks.
The present study proposes an integration of BERT with hyperparameters optimized via genetic algorithms in order to overcome these shortcomings with a systematic methodology to find an appropriate combination of

parameters of the number of epochs, learning rate, and optimizer. The method is more accurate than manual tuning methods and also solves the problems that were pointed out in the previous studies conducted. Table 3 provides a detailed comparison of the reviewed studies, highlighting their respective methodologies, findings, strengths, and limitations.

Table 3: Comparison of reviewed articles

Article Title	Study Objective	Methodology	Key Findings	Strengths	Weaknesses
Multilingual Evaluation of Pre-Processing for BERT-based Sentiment Analysis of Tweets[17]	Multilingual Preprocessing Evaluation to Enhance BERT-Based Sentiment Analysis on Tweets	Evaluating the Impact of Different Text Preprocessing Techniques on BERT Performance in Multiple Languages	Techniques of appropriate preprocessing were found to substantially enhance the aptness of the BERT model in performing multi-lingual sentiment analysis tasks in English, Italian, and Spanish datasets in languages other than English, though the exact figures were not reported.	Emphasizing on multilingual analysis and the importance of preprocessing in improving model performance	Lack of examining the combination of different preprocessing methods and their impact on model performance
Aspect-Based Sentiment and Emotion Analysis with ROBERTa, LSTM[9]	Aspect-Based Sentiment and Emotion Analysis Using Hybrid RoBERTa and LSTM Models	Using RoBERTa-LSTM Hybrid Model for Aspect-Based Sentiment and Emotion Analysis of Ukraine-Russia War Tweets	The RoBERTa-LSTM hybrid model scored 94.7% accuracy and exceeded the existing methods in aspect-based sentiment and emotion analysis.	Combining advanced models and using deep learning methods for more accurate sentiment analysis	Lack of comprehensive examination of the impact of different hyperparameter settings and different model applications in different environments
Sentiment Analysis Classification System Using Hybrid BERT Models[20]	A hybrid BERT-based system for sentiment classification to improve accuracy	Combination of BiLSTM and BiGRU with BERT is applied to sentiment classification on multi-purpose data, with and without emojis	The hybrid model which combines BiGRU & BiLSTM with RoBERTa and DistilBERT performed better in terms of accuracy compared to the baseline S-BERT models on the three different datasets of tweets. The preprocessing analyses indicated the effect of the removal of emojis in the dataset.	Using multi-purpose data to examine the impact of preprocessing on model performance	Limited to three datasets and lack of examination of the applications of the proposed models in other environments
A BERT Framework to Sentiment Analysis of Tweets[8]	Developing an advanced framework for Twitter sentiment analysis using BERT and different types of deep neural networks	Combining BERT model with CNNs and RNNs to improve the accuracy of sentiment analysis	The combination of BERT with CNNs, RNNs, and BiLSTM led to obtaining 93% accuracy and 95% F1-score on six datasets, thereby significantly outdoing word-to-vector conversion methods.	Using a combination of different deep learning models to improve model accuracy and performance	Reliance on internet-sourced data and restricted to basic sentiment analysis
Topic-BERT (T-BERT) Model for Sentiment Analysis of Micro-blogs[23]	Investigating the effectiveness of a hybrid T-BERT model for sentiment analysis using Twitter datasets	Utilizing a hybrid T-BERT model for improving sentiment analysis accuracy	The T-BERT hybrid model that integrated LDA topic modeling with BERT has achieved 90.81% accuracy on a 42,000-tweet dataset, thus proving its effectiveness in the sentiment analysis of microblogs.	Integrating topic models with BERT to improve sentiment analysis performance	Limited scalability to complex data and high computational requirements

Article Title	Study Objective	Methodology	Key Findings	Strengths	Weaknesses
Research on the Application of Deep Learning-based BERT Model in Sentiment Analysis[26]	Investigating the application of BERT models in sentiment analysis and improving its performance through model fine-tuning	Applying BERT to optimize performance through experimental evaluation	BERT demonstrated effective sentiment analysis performance with significant improvements after optimization and fine-tuning, though specific quantitative metrics were not reported in the reviewed study.	Optimizing the performance of BERT model and using it in high-accuracy sentiment analysis	Requires high computational resources for large-scale and time-consuming data processing
Emotion and Sentiment Analysis of Tweets using BERT[24]	Exploring the use of BERT models to identify emotions and sentiments in tweets	Utilizing BERT Models for High-Accuracy Sentiment Analysis and Emotion Recognition	BERT achieved 92% accuracy in sentiment analysis and 90% accuracy in emotion identification on Twitter data, demonstrating strong performance in both tasks.	High accuracy in sentiment and emotion recognition in tweets	Requires more data for improved performance and has limitations in analyzing complex or multilingual texts
BERT based classification system for detecting rumors on Twitter[25]	Rumor detection on Twitter using BERT model	Rumor detection on Twitter using BERT model with extracted tweet content features	The BERT-based rumor detection delivered better performance and faster processing time than the conventional techniques through the automatic feature extraction, therefore, without the need for manual feature engineering (exact accuracy figures not disclosed).	Improving time efficiency with BERT by removing manual feature extraction	Requires processing large datasets and high-volume tweets
Sentiment Analysis on COVID Tweets Using COVID-Twitter-BERT with Auxiliary Sentence Approach [22]	COVID tweet sentiment analysis using COVID-Twitter-BERT and the auxiliary sentence method	Using Covid-Twitter-BERT model and the auxiliary sentence method to analyze the sentiment of Covid-19 tweets	The Covid-Twitter-BERT with auxiliary sentence strategy realized higher accuracy and F1-score in the COVID-19 tweet sentiment analysis than that of the standard classification methods (exact percentage improvements given in the original study).	Combining COVID-Twitter-BERT and auxiliary sentences enhances model accuracy	Improving model performance requires larger and more diverse data

A literature review showed that previous research in sentiment analysis has only focused on combining BERT models with classical methods and has not considered the issue of hyperparameter optimization and the use of evolutionary algorithms in combination with advanced models such as BERT. Considering the limitations observed in the reviewed papers, this study, by combining the BERT model and GA, presents a proposed framework that improves the performance of the model, which was one of the major challenges in previous papers. This approach, by utilizing evolutionary algorithms, provides more accurate optimization of the model hyperparameters and effectively addresses the gaps identified in previous studies.

4- Proposed GA-BERT Optimization Model

This section presents the proposed GA-BERT model for sentiment classification. The main innovation of this research is the integration of the fine-tuning process of BERT model with a hyperparameter optimization mechanism based on genetic algorithms. The model is designed to automatically identify the appropriate combination of three hyperparameters that influence the fine-tuning of BERT: number of training epochs, learning rate, and optimizer type.

Unlike the standard BERT training process, the proposed model introduces an evolutionary search process in which different combinations of hyperparameters are evaluated

and iteratively improved based on their performance on the validation set.

4-1- Overview of the Proposed Model

The proposed model consists of two interconnected parts. The first part is a BERT-based sentiment classification module that receives preprocessed tweets and classifies them into three sentiment categories: Disappointed, Happy, and Angry.
The second part is a GA-based optimization module that searches the hyperparameter search space and determines the best configuration for fine-tuning BERT model.

In this paper, text preprocessing, tokenization, data partitioning, BERT model training, and model evaluation are considered as standard implementation steps. These steps are necessary to ensure the reproducibility of the research, but do not constitute the main contribution and algorithmic innovation of this study.
The main and innovative component of this research is the GA-based optimization mechanism that controls the fine-tuning process of BERT model and determines the optimal values of the number of training epochs, learning rate, and optimizer type. The overall architecture of the proposed model is shown in Figure 3.

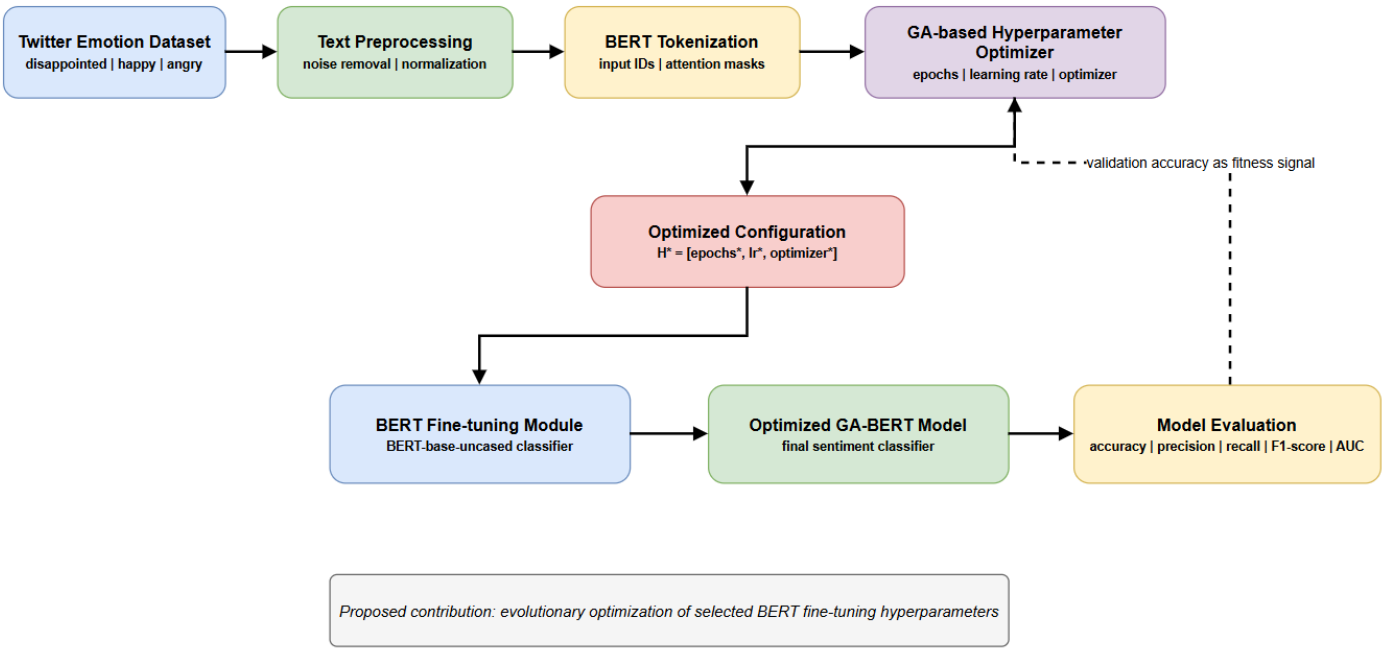

Fig. 3 Architecture of the Proposed GA-BERT Optimization Model

4-2- Main Components of the Proposed Model

The proposed model includes the following main components:

1. BERT-based sentiment classifier: In this section, BERT-base-uncased model is used as the core of the transformer architecture-based classifier. This model receives tokenized tweets and provides sentiment predictions for three target classes. BERT fine-tuning process is performed using a combination of hyperparameters selected by the genetic algorithm.
2. Hyperparameter Search Space: The search space consists of three main hyperparameters in BERT fine-tuning process: the number of training epochs, the learning rate, and the type of optimizer. Each solution in the genetic algorithm represents a possible combination of these hyperparameters.
3. GA-based optimizer: The genetic algorithm first generates an initial population of different combinations of hyperparameters. It then evaluates each combination based on the accuracy obtained from the validation of the tuned BERT model and gradually improves the population through selection, crossover and mutation operators.
4. Fitness Evaluation Module: For each solution, BERT model is fine-tuned using the training data and then evaluated with the validation data. The fitness value is calculated based on the relationship (1 - Accuracy); therefore, the optimization problem is formulated as a minimization problem.

5. Optimized GA-BERT Model: After the evolutionary search process is completed, the best combination of hyperparameters is selected and used for the final training of BERT-based sentiment classifier. . Finally, the model performance is evaluated using the criteria of Accuracy, Precision, Recall, F1 score and Area Under the ROC Curve (AUC).

4-3- GA-based Optimization Workflow

The optimization process begins by encoding each combination of hyperparameters as a chromosome. Each chromosome contains three genes: the number of training epochs, the learning rate, and the optimizer type. An initial population is then randomly generated within predetermined ranges for the search. Each chromosome is transformed into a configuration for fine-tuning BERT model, and BERT model is trained and validated against that configuration. The accuracy obtained from the validation is then used to calculate a fitness value.

After evaluating the fitness, the genetic algorithm selects the chromosomes that perform better as parents. The crossover operator combines the parent chromosomes together to create new configurations, while the mutation operator maintains the population diversity and reduces the likelihood of premature convergence by making random changes to the selected genes.

This evolutionary process continues until a predetermined number of generations are reached. Finally, the chromosome with the lowest fitness value is selected as the optimal configuration of hyperparameters. The optimization process based on the proposed genetic algorithm is presented in Figure 4.

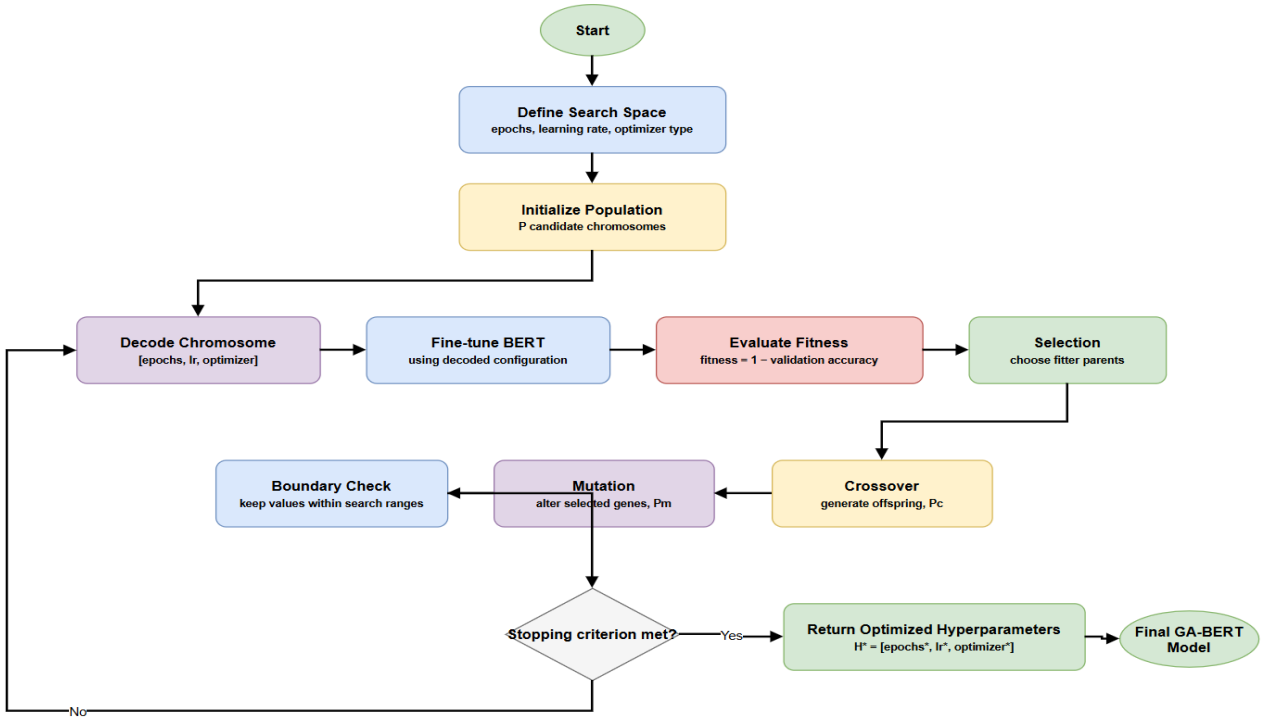

Fig. 4 GA-based Hyperparameter Optimization Workflow

4-4- Assumptions and Scope of the Proposed Model

The proposed model is designed based on two main assumptions. First, it is assumed that the computational environment, including hardware and software settings, remains constant during the training and evaluation stages to ensure the repeatability of the experimental results. Second, it is assumed that the selected hyperparameters, including the number of training epochs, learning rate, and optimizer type, have a direct impact on the fine-tuning performance of BERT model.

Other parameters, including the batch size, dropout rate, and number of hidden layers of BERT, were kept constant based on the conventional settings of BERT-base model to reduce the computational complexity and focus the optimization process on the selected search space.

These assumptions determine the scope of the proposed model. Therefore, the innovation and main contribution of

this research should be interpreted as the optimization of the selected hyperparameters in BERT fine-tuning process using a genetic algorithm, not as a change in the internal architecture of BERT model.

4-5- Experimental Dataset

In this study, we used the public dataset "Cleaned Emotion Extraction Dataset from Twitter" available on the Kaggle platform. This dataset was originally collected from Twitter and contains 916,575 tweets categorized into three emotion classes: disappointment, happiness, and anger. The distribution of classes is approximately balanced, with 313,714 tweets in the disappointment class (34.2%), 301,871 tweets in the happiness class (32.9%), and 300,990 tweets in the anger class (32.9%).

This balanced distribution reduces the risk of bias towards a particular class in the model training process and justifies the use of the accuracy criterion as a fitness criterion in the GA-based optimization process.

The emotion labels were already present in the original dataset and were used without any manual relabeling. The dataset consists of three main columns: sentiment tags, cleaned tweet content, and original tweet content.

The cleaned content was used as the main input to BERT model, while the original tweet content was retained to maintain transparency and allow for verification of the accuracy of the preprocessing steps.

This is an even spread of cases across the categories of sentiment, which means that the training of the model will not be affected by any one emotion in particular. The balanced distribution across the categories of sentiment, which means that the training of the model will not be affected by any one emotion in particular.

Data preprocessing for this study was conducted through the use of automated Python scripts. The original dataset had a cleaned version of tweets already, but further preprocessing steps were applied in a planned manner so that the data could be input to the BERT model. Programmatic removal of URLs using regular expressions, user mentions (@) and hashtags (#) elimination, conversion to lowercase, and punctuation marks removal were among the steps taken. All the preprocessing operations were carried out in the same way throughout the whole dataset by using Python libraries like re (regular expressions), pandas, and NLTK. The data has been categorized in three major columns such as emotions, content, and original content. The emotions column includes the sentiment labels that are connected to every tweet. The content column has the preprocessed and cleaned version of tweets that are ready for the deep learning model to process. In the original content column, the raw tweets collected from Twitter are kept, which consist of emojis, hashtags, and symbols, and this column is the main source for preprocessing. All tweets were anonymized to ensure user privacy protection by removing user identifiers, and only publicly available content was used which is in keeping with the ethical research standards and data privacy regulations. The researchers followed ethical research practices while working with social media data. The data was all collected from tweets that were publicly available and this was done according to Twitter's terms of service and data usage policies. Regarding potential biases, we acknowledge that Twitter data may reflect demographic and geographic biases inherent to the platform's user base. The balanced class distribution reduces the risk of model bias toward any particular sentiment category. However, the applicability of findings to other social media platforms or offline contexts might be restricted and this should be considered when drawing conclusions. Table 4 provides a summary of the dataset structure and representative samples from each sentiment category.

Table 4: Dataset Content and Data Sample

Column	Description	Sample
Emotions	Refers to the emotion linked to the tweet text (e.g., "disappointed," "happy," "angry"), used as sentiment labels for data analysis	disappointed: 313,714; happy: 301,871; angry: 300,990
Content	Preprocessed tweet text, cleaned of noise and irrelevant elements, prepared for deep learning model processing and analysis	imagine if that reaction guy that called jj kfc saw this my man would ve started crying lmao
Original Content	The original and unprocessed tweets containing all essential details, including emojis, hashtags, and symbols, used as the primary data source	"@KSIOlajidebt imagine if that reaction guy that called JJ KFC saw this. my man would've started crying lmao"

4-6- Implementation Process

In this section, the implementation process used to train, validate, and optimize the proposed GA-BERT model is described. The implementation process includes the steps of data collection, data preprocessing, data partitioning, data tokenization using BERT, generating data set in PyTorch, model fine-tuning, model performance evaluation, and GA-based hyperparameter optimization. The general implementation steps are presented to ensure the reproducibility of the research, while the algorithmic innovation of this research specifically lies in the GA-based hyperparameter optimization process, which is described in Algorithm 1.

4-6-1- Step 1: Dataset Selection and Acquisition

The selection of an appropriate dataset for sentiment analysis is the basis of this research. This study used the Cleaned Emotion Extraction Dataset from Twitter, which is found on Kaggle. The selection of the data was determined by some robust criteria, which included the quality, balanced class distribution, relevance to the research, and the availability of pre-labeled sentiment categories. The data consists of a cumulative total of 916,575 tweets, which fall under the following three categories: Disappointed – 313,714 (34.2%), Happy – 301,871 (32.9%), and Angry – 300,990 (32.9%). The almost equal distribution was a major reason for the selection because it reduces the problems of class imbalance and allows the model to recognize the patterns in all sentiment categories equally. The dataset was first acquired by Kosweet using the Twitter API and it was done following Twitter's Developer Agreement and Policy, hence the data collection was done ethically. The dataset structure includes three columns: emotions (sentiment labels), content (cleaned text), and original content (raw tweets). The choice of having both raw and pre-cleaned versions gave the opportunity to implement custom preprocessing steps according to the BERT model requirements. Moreover, the dataset being publicly accessible on Kaggle is an assurance of this research's reproducibility, other researchers will thus be able to confirm and further develop the findings. Detailed dataset characteristics are presented in the Dataset section (Section 4.4).

4-6-2- Step 2: Data preprocessing

The text preprocessing was performed by means of automated Python scripts in order to keep the whole dataset uniform and repeatable. The dataset in question was one that had previously assigned sentiment labels (disappointed, happy, angry) that were used in the experiment without any changes. These labels were checked through the original data set and no further manual coding was needed. The pipeline was also automated via the Python libraries used to carry out regular expressions for pattern matching, pandas for manipulation of data, and NLTK for text processing utilities. The preprocessing operations on tweet text included extracting URLs through the use of regular expressions, deleting user mentions (the ones beginning with @) and hashtags (the ones beginning with #), making all text lowercase to treat it uniformly, and getting rid of punctuation signs while content of the tweets still being conveyed. The labels of the sentiments were converted to a numerical form through the application of scikit-learn's LabelEncoder. The labels included emotions of "disappointed," "happy," "angry" and were converted to integer labels "0," "1," "2" that were ideal for the learning process. This process of encoding was done systematically all over the dataset with the encoder fitted on the training data to keep proper data handling practices. All the preprocessing steps were done programmatically without any manual intervention, thus ensuring scalability and reproducibility of data preparation process.

4-6-3- Step 3: Data Division

The preprocessed data was split into two parts - training (80%) and evaluation (20%) which helped in the determination of the generalizability of the model and thus, overfitting was avoided.

4-6-4- Step 4: Data Tokenization

The BERT-base-uncased tokenizer was used to perform the text tokenization that transforms the text into sequences of tokens. The process of tokenization entails the use of the truncation process and padding that gives uniform length of sequences and aids uniform input to the model. This version is designed to process English texts and considers all letters in lowercase. This tokenizer was designed to maintain a balance between accuracy and efficiency and is widely used in natural language processing research. The tokenization process involves converting each text into a sequence of tokens and then applying complementary operations such as truncation and padding to ensure that the sequences are of equal length. These operations help the model process input texts in a consistent and homogeneous format and avoid defects that can arise due to differences in text length.

4-6-5- Step 5: Creating PyTorch Dataset Objects

The data after tokenization was transformed into objects of PyTorch Dataset to allow processing by batches in an efficient manner during model training and evaluation. The implementation of a custom dataset class was done, which inherits the torch.utils.data. Dataset base class, providing standardized methods for data access and manipulation. The custom Sentiment Dataset class contained the tokenized encodings along with the respective sentiment values that facilitated the definition of the necessary __getitem__ method supporting the return of individual samples as tensors along with the definition of the __len__ method providing the dataset's length. This allowed PyTorch's DataLoader module to perform the operations of batch generation, shuffling of data, along with the iteration over the dataset efficiently while the model was being trained. The training and validation split dataset objects were instantiated independently with the help of the generated encodings and labels of the tokens of the training and validation splits, in turn, creating the necessary data separation in the development pipeline.

4-6-6- Step 6: Model Training

The BERT-base-uncased model was loaded from the Hugging Face transformers library and configured for sequence classification with three output classes corresponding to the sentiment categories. Model training was executed on NVIDIA GeForce RTX 3090 GPU hardware to accelerate computation, with all tensors automatically transferred to CUDA device during processing. Training was performed using the AdamW optimizer with the learning rate determined by the genetic algorithm optimization process (detailed in Step 8). The model was trained using PyTorch DataLoader objects with a batch size of 16, iterating through the specified number of epochs as optimized by the GA. For each training batch, the following operations were executed: optimizer gradients were reset using zero_grad(), input tensors (input_ids, attention_mask) and labels were moved to GPU memory, a forward pass computed the model outputs and classification logits, cross-entropy loss was calculated between predictions and true labels, backward propagation computed gradients via loss.backward(), and the optimizer updated model parameters through optimizer.step(). Training progress was monitored through loss values across batches and epochs to ensure proper convergence. The model weights were preserved after training for subsequent evaluation and hyperparameter optimization iterations.

4-6-7- Step 7: Model Evaluation

The performance of the BERT model was evaluated on the validation dataset, which was the model's assessment to unseen data. The BERT model was set to the evaluation mode with the help of model.eval() function. Hence, dropouts were turned off and predictions were made consistently. In the process of evaluation, no gradients were calculated (by means of the torch.no_grad() context) and model weights were kept unaltered. For each validation batch, input tensors were transferred to GPU memory and passed through the model to generate predictions. The predicted classes were found out by applying the argmax function to the output logits of the model. The predictions were eventually accumulated for all validation batches, together with the actual classes, for comprehensive model assessment. Model performance was quantified using five key evaluation metrics: accuracy (overall percentage of correct predictions), precision (ratio of true positive predictions to all positive predictions), recall (ratio of true positives to all actual positives), F1-score (harmonic mean of precision and recall), and area under the ROC curve (AUC, measuring the model's ability to discriminate between classes). The above-mentioned calculation of the metrics has been performed by using the classification metrics provided by the Scikit-learn library. The evaluation results of this phase were considered for model convergence assessment and for taking decisions regarding the genetic algorithm optimization process in the subsequent iterations.

4-6-8- Step 8: Hyperparameter Optimization

Hyperparameter optimization using GA is used as a new approach to improve the performance of BERT model. Genetic algorithm, as an evolutionary optimization method, imitates natural selection processes to find an optimal combination of hyperparameters such as epochs, learning rate, and optimizer type. This optimization is based on iterated evaluations of the model's performance under different conditions, so that each time a different combination of parameters is tested and the results are measured using accuracy criteria. The optimization process is designed in such a way that its ultimate goal is to reduce the mistake and increase the model's accuracy at the same time. This approach based on genetic evolution is very effective and efficient due to its ability to search large and complex parameter spaces, especially in deep learning models such as BERT. Here, important parameters of the genetic algorithm such as population size, crossover rate, and mutation rate are finely tuned to optimize the search in the parameter space. The exact values of the genetic algorithm parameters used in this optimization step are given in Table 5. The fitness function was defined as (1 - accuracy) to minimize error and maximize accuracy. Although F1-score is often preferred for imbalanced datasets, our dataset largely has a balanced distribution of classes (disappointed: 313,714; happy: 301,871; angry: 300,990), with a maximum class imbalance ratio of approximately 1.04:1. For the balanced environment that was created, accuracy was proven to serve as an apt and efficiently calculated optimization function. The end result of the entire evaluation of the model would consider different aspects of the dataset including precision, recall, and the F1-score (as seen in Tables 9 and 10) to provide a comprehensive assessment of model performance across all classes.

Table 5: Parameter values of the genetic algorithm used in optimization

Parameter	Value Used
Population Size	10
Crossover Rate	0.8
Mutation Rate	0.2
Number of Generations	2

In the optimization process of BERT model, key hyperparameters including epochs, learning rate, and optimizer type are set. The permissible range of these hyperparameters is presented in Table 6.

Table 6: The permissible ranges of BERT model hyperparameters used in optimization

Hyperparameter	Range of Values
Number of Epochs	[10, 100]
Learning Rate	[1e-6, 1e-5]
Optimizer Type	[0 (AdamW), 1 (SGD)]

Here, each solution in the GA is considered as a specific combination of hyperparameters including epochs, learning rate,

and optimizer type. The algorithm searches for the optimal combination by evaluating the performance of each combination through the accuracy of the model. The goal of this optimization process is to maximize the accuracy of the model by minimizing the error (defined as 1 minus the accuracy).

The hyperparameters allowed would be set according to the conventional best practices defined in the BERT fine-tuning literature. The epoch range of 10-100 corresponds to suggestions for transformer-based models, where fewer than 10 epochs usually lead to underfitting, and more than 100 epochs seldom produce any performance improvements and rather increase the risk of overfitting [5][4]. The learning rate range of 1e-5to 1e-6 is in line with the generally accepted range for BERT fine-tuning since learning rates outside this range are likely to result in either unstable training or slow convergence [10].

The hyperparameter values selected for BERT model after optimization using the GA are given in Table 7. These optimized values have provided the best performance for the model in sentiment analysis.

Table 7: Optimized hyperparameter values of BERT Model after genetic algorithm optimization

Hyperparameter	Selected Value	Description
Number of epochs	90	Setting the number of epochs to 90 improves model learning, particularly for training on diverse datasets
Learning Rate	1e-5	A learning rate of 1e-5 is selected as an optimal learning rate to accelerate model convergence
Optimizer Type	AdamW (0)	The AdamW optimizer is chosen for its efficiency and enhanced performance in many deep learning models

The number of epochs is set to 90 so that the model can be well optimized on the training dataset without overfitting, which is suitable for models that do not require more iterations for convergence. The learning rate is chosen to be 1e-5, which is a relatively small value and allows the model to gradually update its weights and avoid large and sudden changes. Also, the AdamW optimizer is chosen as the model optimizer due to its high efficiency in adjusting the weights and its ability to avoid convergence problems, which allow the model to achieve more accurate results in sentiment prediction.

Ultimately, this optimization method increases the accuracy of BERT model and improves the hyperparameter tuning process in an efficient and effective manner. This improvement allows researchers to create models with better performance and higher generalizability, which can be useful in various fields of text data analysis and natural language processing applications. The mechanism of GA-based optimization, which constitutes the main algorithmic innovation of this research, is summarized in Algorithm 1.

4-6-9- Step 9: Final Training Using Optimal Hyperparameters

After the GA-based optimization process identified the best combination of hyperparameters, BERT model was fine-tuned using the resulting optimized configuration. The performance of the final model was then evaluated using the criteria of Accuracy, Precision, Recall, F1 score, and Area Under the ROC Curve (AUC) on the validation dataset.

The results were used to compare with the baseline models, including standard BERT (Vanilla BERT), LSTM, and CNN. The algorithmic innovation of this research is summarized in Algorithm 1, which specifically focuses on the GA-based hyperparameter optimization process.

The innovation of this research in the general sentiment classification pipeline does not include dataset preparation, preprocessing, tokenization, BERT model training, or model evaluation. These operations are standard implementation steps required for training transformer-based classifiers.

The innovative part of this research is the GA-based optimization mechanism that automatically searches and determines the best combination of hyperparameters used in fine-tuning BERT model. Therefore, the pseudocode presented in Algorithm 1 is limited only to the proposed process of GA-based hyperparameter optimization and includes chromosome representation, fitness evaluation, selection, crossover, mutation, and extraction of the optimal configuration of BERT model.

Table 8. Pseudocode of the Proposed algorithm for GA-based Hyperparameter Optimization

Algorithm 1. GA-based Hyperparameter Optimization for BERT Sentiment Classification

Input:

- Training dataset D_train
- Validation dataset D_val
- Search space H = {epochs, learning rate, optimizer type}
- Population size P
- Crossover rate Pc
- Mutation rate Pm
- Maximum number of generations G

Output:

- Optimized hyperparameter set H*
- Best validation accuracy Acc_best
- Final GA-optimized BERT model

1. Define the chromosome representation:
chromosome_i = [epochs_i, learning_rate_i, optimizer_i]
2. Initialize a population Pop = {chromosome_1, chromosome_2, ..., chromosome_P}
by randomly sampling each chromosome from the predefined hyperparameter search space H.
3. For each chromosome_i in Pop:
a. Decode chromosome_i into:
epochs_i
learning_rate_i

optimizer_i
b. Fine-tune BERT on D_train using the decoded hyperparameters.
c. Evaluate the fine-tuned BERT model on D_val.
d. Compute the fitness value:
fitness_i = 1 - Accuracy_i
4. Set the best chromosome as the chromosome with the minimum fitness value.
5. For generation = 1 to G do:
a. Select parent chromosomes from Pop based on their fitness values.
b. Apply crossover with probability Pc to generate offspring chromosomes.
c. Apply mutation with probability Pm to selected genes in the offspring chromosomes.
d. Ensure that all generated hyperparameter values remain within the predefined search space H.
e. For each offspring chromosome:
i. Decode the chromosome into BERT hyperparameters.
ii. Fine-tune BERT using the decoded configuration.
iii. Evaluate the model on D_val.
iv. Compute fitness = 1 - Accuracy.
f. Update the population using the newly evaluated offspring.
g. Update the best chromosome if a lower fitness value is obtained.
6. Return the best chromosome as the optimized hyperparameter set H*.
7. Fine-tune the final BERT sentiment classifier using H*.
8. Report the final performance metrics, including accuracy, precision, recall, F1-score, and AUC.

In Algorithm 1, the proposed part of the research is the GA-based optimization and search process used to fine-tune BERT model. The standard implementation steps, including data set loading, text preprocessing, data partitioning, tokenization, generating data set in PyTorch, model training, and model performance evaluation, are intentionally omitted from the pseudocode because these steps are not considered algorithmic innovations.
These steps are described separately in the implementation process section to ensure the reproducibility of the research, while Algorithm 1 focuses solely on the proposed optimization mechanism.

4-7- Analysis Methods

To evaluate the performance of the proposed hybrid algorithm, the developed model was compared with the CNN algorithms, LSTM, and Vanilla BERT. This comparison was made based on the main criteria for evaluating machine learning and deep learning models, which include precision, accuracy, recall, F-1 score, Receiver Operating Characteristic (ROC) curve, and the Area Under the Curve (AUC) [22, 23, 24].

- Accuracy: Accuracy is one of the main criteria for evaluating the performance of classification models and indicates the percentage of samples that are correctly classified. Accuracy is generally a suitable metric for data that has a balanced distribution between classes. Accuracy is calculated as follows [27],[30]:

$$\text{Accuracy}=\frac{\text{True Positive}+\text{True Negative}}{\text{True Positive}+\text{True Negative}+\text{False Positive}+\text{False Negative}} \quad (1)$$

In this equation, the number of correct predictions includes the number of samples that the model correctly classified into the corresponding classes. This metric is particularly suitable for multi-class classification problems where the data distribution is balanced.
- Precision: Precision is the percentage of correctly predicted positive samples. In other words, this metric shows how many of all samples predicted as positive are actually positive. This metric is especially important in situations where the cost of false positive predictions is high (e.g., in fraud detection). The equation for calculating precision is as follows [28],[30]:

$$\text{Precision}=\frac{\text{True Positive}}{\text{True Positive}+\text{False Positive}} \quad (2)$$

where, the number of true positive predictions refers to the examples that are truly positive and correctly predicted by the model.
- Recall: Recall represents the percentage of true positives that are correctly identified by the model. In other words, this metric looks at the problem from the perspective of true positives and determines how many true positives the model correctly identified. Recall is important in situations where the cost of missing positives is high (e.g., in cancer diagnosis). Recall is calculated as follows [29]:

$$\text{Recall}=\frac{\text{True Positive}}{\text{True Positive}+\text{False Negative}} \quad (3)$$

here, the number of true positives includes all samples that actually belong to the positive category, whether correctly predicted or incorrectly.
- F-1 Score: A composite metric that measures the trade-off between precision and recall. This metric is particularly useful when there is a need to strike a balance between precision and recall, and we do not want either metric to be overly influential. The F-1 score is defined as the harmonic mean of precision and recall, and it is expressed as follows [28]:

$$\text{F1 Score}=\frac{2\times\text{Precision}\times\text{Recall}}{\text{Precision}+\text{Recall}} \quad (4)$$

This metric will be high when both precision and recall are high. Hence, the F-1 score is used as a comprehensive and useful metric to evaluate the performance of models in unbalanced classification problems.

- Calculating the area under the curve (AUC): The area under the curve is a numerical measure for evaluating the performance of binary classifier models that calculates the area under Receiver Operating Characteristic (ROC) curve. The value of the area under the curve varies between 0.5

and 1, where a value of 0.5 indicates random performance and a value of 1 indicates excellent model performance. In general, the higher the value of the area under the curve, the more capable the model is in distinguishing between classes. The area under the curve is a comprehensive indicator because it considers all possible thresholds and takes into account the fluctuations caused by threshold changes, so it can be an effective and accurate measure for evaluating the overall performance of classifier models.

5- Analysis of Results

The accuracy, precision, recall, F1-score, and AUC metrics were used to evaluate the proposed GA-optimized BERT model on both training and testing datasets. The performance results of the model are reported in Table 9.

Table 9: Evaluation results of the hybrid model using five key metrics

Evaluation Metric	Proposed Model (Training)	Proposed Model (Testing)
Accuracy	98.2%	98.1%
Precision	98.4%	98.2%
Recall	98.3%	98.1%
F1-Score	98.35%	98.15%
AUC	98.8%	98.5%

The purpose of Table 9 is to show model performance in both training and testing datasets which is needed to evaluate the generalization capability and to identify potential overfitting. The minimal performance gap between training and testing phases (98.2% vs 98.1% accuracy) clearly shows that the GA-optimal BERT model generalizes extremely well on new examples without any overfitting on the training examples.

This small difference of 0.1% in accuracy, along with similarly minimal variations in other metrics (precision: 0.2%, recall: 0.2%, F1-score: 0.2%, AUC: 0.3%), confirms that the hyperparameter optimization successfully balanced model complexity with generalization capability. Performance of the highest level on all examples of both steps validates the efficiency of genetic algorithm optimization approach. The training phase results (98.2% accuracy, 98.4% precision, 98.3% recall, 98.35% F1-score, 98.8% AUC) demonstrate the model's learning capacity, while the testing phase results (98.1% accuracy, 98.2% precision, 98.1% recall, 98.15% F1-score, 98.5% AUC) confirm robust performance on unseen data. These findings indicate that optimized hyperparameters allowed the model to extract significant patterns from the training data and at the same time to maintain strong predictive power against new samples.

Next, the performance of the proposed hybrid BERT model with genetic optimization was compared with other standard sentiment analysis algorithms, including vanilla BERT, LSTM, and CNN. This comparison is performed to demonstrate the superiority of the proposed model and accurately evaluate its performance against other well-known models. The results of this comparison are shown in Table 10.

Table 10: Comparative results between the proposed model and other algorithms

Evaluation metric	Proposed Model	BERT	LSTM	CNN
Accuracy	98.1%	92.8%	91.3%	90.5%
Precision	98.2%	91.2%	91.1%	90.2%
Recall	98.1%	92.0%	90.7%	90.1%
F1-Score	98.15%	91.6%	90.9%	90.0%
AUC	98.5%	93.2%	91.0%	89.6%

As shown in Table 10, the accuracy of the model indicates the percentage of test data samples that are correctly classified. The proposed model, with an accuracy of 98.1%, significantly outperform the other models. In comparison, the accuracies of Vanilla BERT, LSTM, and CNN models are 92.8%, 91.3%, and 90.5%, respectively. This significant difference in accuracy indicates that genetic optimization had a positive impact on the performance of BERT model. The precision indicates how many of all samples that are predicted as a specific positive class are correctly classified. The precision of the proposed model is 98.2%, which is higher than that of Vanilla BERT (91.2%), LSTM (91.1%), and CNN (90.2%). This indicates the proposed model is able to identify positive samples more accurately and has fewer false positive errors. The recall or sensitivity of the model indicates what percent of all true positive samples are correctly identified. The proposed model achieved 98.1% recall, outperforming BERT (92.0%), LSTM (90.7%), and CNN (90.1%), indicating its superior detection of true positives. The F-1 score provides a balance between precision and recall. This measure is especially useful when the data is unbalanced. The proposed model's F1-score of 98.15% outperforms BERT (91.6%), LSTM (90.9%), and CNN (90.0%), demonstrating its strength in detecting true positives while reducing false positives.

The area under the curve is a more comprehensive measure of the model's performance, which indicates its ability to discriminate between different classes. The proposed model with an area under the curve of 98.5% had the best performance compared to other models (BERT 93.2%, LSTM 91.0%, CNN 89.6%). The high area under the curve means the proposed model has a high ability to discriminate between samples and generally performs better in all classification scenarios. This comparison demonstrates that the proposed model significantly outperforms other models in precision, recall, F1-score, and especially in AUC. This optimal performance is the result of improving the model parameters through the use of genetic algorithm to fine-tune hyperparameters, which led to the development of a model with very high accuracy and efficiency in sentiment analysis.

One of the important tools for measuring the accuracy and precision of models in sentiment analysis is the confusion matrix. The results of examining the performance of different models in sentiment analysis using this tool are shown in Figure 6. This figure clearly shows how each model placed the samples in the correct or incorrect categories and reveals the difference in performance between them.

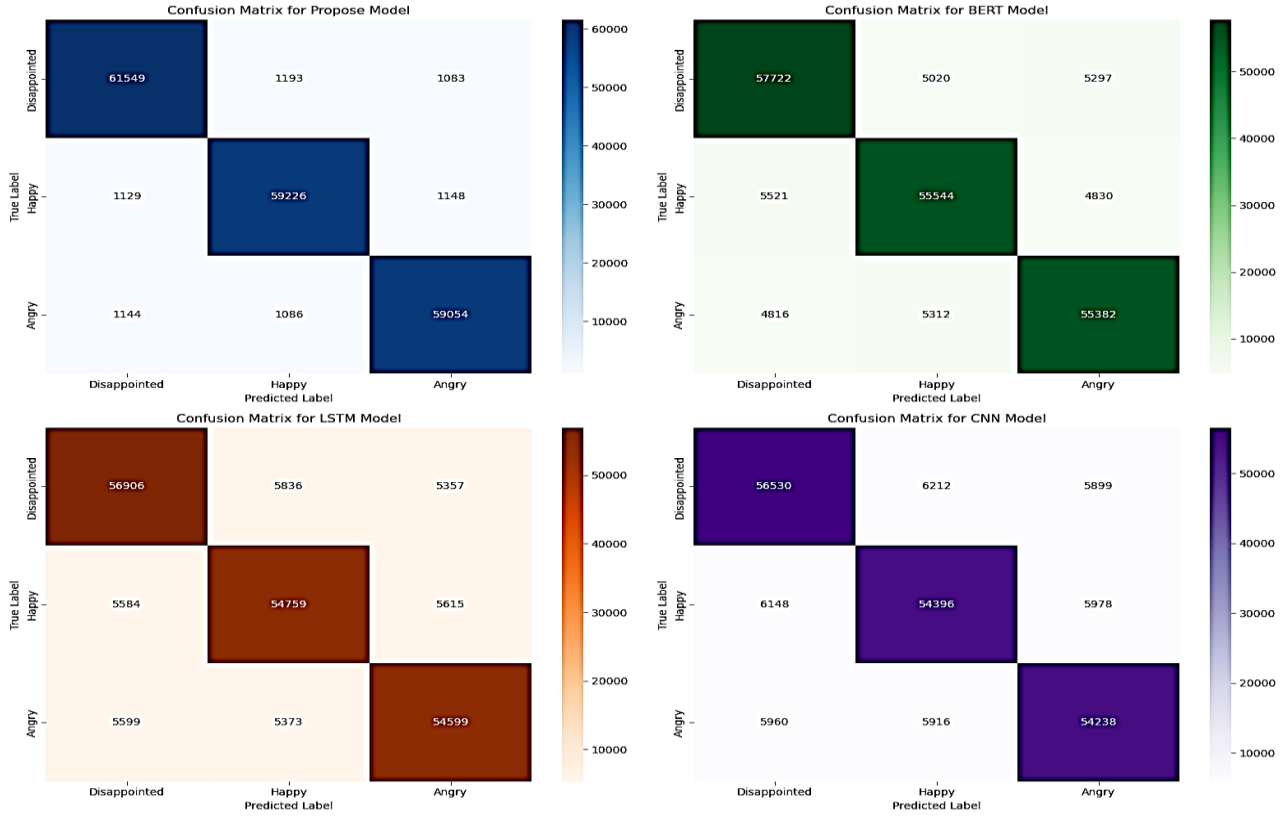

Fig. 6 Confusion matrices of four different models for sentiment analysis

The optimized BERT model (proposed model) using hyperparameter optimization showed excellent performance in all metrics including accuracy, precision, recall, and F-1 score. This optimized model was able to effectively reduce the type I and type II error rates, especially in the classes labeled "disappointed" and "angry", which indicates the high accuracy of this model in distinguishing between these emotions. In comparison, BERT model without optimization has more errors in recognizing the "happy" classes, which indicates the weakness of this model in the optimal tuning of hyperparameters. These differences demonstrate the importance of fine-tuning hyperparameters in improving the performance of deep learning models.

LSTM model had a lower performance than BERT models. It is particularly weak in detecting the "disappointed" class, with a higher type II error rate in this class. This may be due to the limitations of LSTM model in learning complex long-term dependencies and semantic differences between various classes. CNN model performs similarly to LSTM model, but performs slightly better in detecting the "happy" class. However, it had a higher error rate in detecting the "angry" class, which indicates the limitations of CNN in understanding long-term dependencies in text data.

Overall, these results show that the optimized BERT model using metaheuristic algorithms such as genetic algorithm can achieve significant performance improvements over more conventional models such as LSTM and CNNs, and even vanilla BERT. These findings confirm that fine-tuning hyperparameters can have a major impact on the accuracy and efficiency of deep learning models in sentiment analysis and can be used as an effective strategy to improve other deep learning models in this field.

6- Conclusion

This study addresses the problem of hyperparameters optimization of BERT model for sentiment classification

in Twitter texts using a genetic algorithm. In many previous studies, default values of hyperparameters or their manual tuning have been used, which can limit the performance of transformer-based sentiment classifiers. In order to overcome this limitation, this study proposed a GA-BERT optimization model, in which the genetic algorithm automatically searches for an effective combination of hyperparameters used in fine-tuning BERT model, including the number of training epochs, learning rate, and optimizer type.

The experimental results showed that BERT model optimized with the genetic algorithm performed very well in sentiment classification. The model achieved a precision of 98.1%, a precision of 98.2%, a recall of 98.1%, an F1 score of 98.15%, and an area under the ROC curve (AUC) of 98.5%.

Also, compared with standard BERT (Vanilla BERT), CNN, and LSTM models, the proposed model outperformed all the main performance evaluation criteria. These findings indicate that the GA-based hyperparameter optimization can improve the effectiveness of BERT model fine-tuning process in sentiment classification based on Twitter texts.

Despite the results obtained, this study has several limitations. First, the tests were conducted on a Twitter-based dataset consisting of three emotion classes, namely disappointment, happiness, and anger; therefore, the generalizability of the obtained optimal configuration to other platforms, application domains, languages, or emotion classification systems requires further investigation.

Second, the optimization process focused on only three selected hyperparameters, while other influential parameters, including Batch Size, Dropout Rate, Weight Decay, Maximum Sequence Length, and Learning-Rate Scheduling, were considered constant.

Third, the GA-based optimization requires repeated execution of BERT model fine-tuning process, which increases the computational cost and training time.

Future research should extend the proposed GA-BERT optimization model in several directions. First, the hyperparameter search space can be expanded to include batch size, dropout rate, weight decay, maximum sequence length, and learning-rate scheduling strategies. Second, the proposed GA-based optimization strategy should be compared with other automated hyperparameter optimization methods, such as Bayesian optimization, particle swarm optimization, random search, grid search, and differential evolution, to evaluate its relative performance and computational efficiency. Third, the generalizability of the optimized model should be assessed using cross-domain, multilingual, and non-Twitter sentiment datasets. Fourth, future studies should investigate computationally efficient variants of the proposed approach, including early stopping, parallel fitness evaluation, lightweight transformer models such as DistilBERT or ALBERT, and surrogate-assisted optimization. Finally, alternative fitness functions, including macro-F1, weighted-F1, AUC, and multi-objective criteria, should be examined, particularly for imbalanced or heterogeneous sentiment datasets.

References

[1] M. Li, L. Chen, J. Zhao, and Q. Li, “Sentiment analysis of Chinese stock reviews based on BERT model,” Applied Intelligence, vol. 51, no. 7, 2021, doi: 10.1007/s10489-020-02101-8.

[2] K. Naithani and Y. P. Raiwani, “Realization of natural language processing and machine learning approaches for text-based sentiment analysis,” Expert Syst, vol. 40, no. 5, 2023, doi: 10.1111/exsy.13114.

[3] A. Mewada and R. K. Dewang, “SA-ASBA: a hybrid model for aspect-based sentiment analysis using synthetic attention in pre-trained language BERT model with extreme gradient boosting,” Journal of Supercomputing, vol. 79, no. 5, 2023, doi: 10.1007/s11227-022-04881-x.

[4] X. Li, X. Wang, and H. Liu, “Research on fine-tuning strategy of sentiment analysis model based on BERT,” in 2021 IEEE 3rd International Conference on Communications, Information System and Computer Engineering, CISCE 2021, 2021. doi: 10.1109/CISCE52179.2021.9445882.

[5] R. Qasim, W. H. Bangyal, M. A. Alqarni, and A. Ali Almazroi, “A Fine-Tuned BERT-Based Transfer Learning Approach for Text Classification,” J Healthc Eng, vol. 2022, 2022, doi: 10.1155/2022/3498123.

[6] J. Martinez-Gil, “Optimizing readability using genetic algorithms,” Knowl Based Syst, vol. 284, 2024, doi: 10.1016/j.knosys.2023.111273.

[7] D. Choudhury and T. Acharjee, “A novel approach to fake news detection in social networks using genetic algorithm applying machine learning classifiers,” Multimed Tools Appl, vol. 82, no. 6, 2023, doi: 10.1007/s11042-022-12788-1.

[8] A. Bello, S. C. Ng, and M. F. Leung, “A BERT Framework to Sentiment Analysis of Tweets,” Sensors, vol. 23, no. 1, 2023, doi: 10.3390/s23010506.

[9] U. Sirisha and B. S. Chandana, “Aspect based Sentiment and Emotion Analysis with ROBERTa, LSTM,” International Journal of Advanced Computer Science and Applications, vol. 13, no. 11, 2022, doi: 10.14569/IJACSA.2022.0131189.

[10] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “Bert: Bidirectional encoder representations from transformers,” arXiv preprint arXiv:1810.04805, p. 15, 2018.

[11] H. Liu et al., “Use of BERT (Bidirectional Encoder Representations from Transformers)-Based Deep Learning Method for Extracting Evidences in Chinese Radiology Reports: Development of a Computer-Aided Liver Cancer Diagnosis Framework,” J Med Internet Res, vol. 23, no. 1, 2021, doi: 10.2196/19689.

[12] L. B. Cesar, M. Á. Manso-Callejo, and C. I. Cira, “BERT (Bidirectional Encoder Representations from Transformers) for Missing Data Imputation in Solar Irradiance Time Series †,” Engineering Proceedings, vol. 39, no. 1, 2023, doi: 10.3390/engproc2023039026.

[13] A. Areshey and H. Mathkour, "Transfer Learning for Sentiment Classification Using Bidirectional Encoder Representations from Transformers (BERT) Model," Sensors, vol. 23, no. 11, 2023, doi: 10.3390/s23115232.

[14] L. Yang and A. Shami, "On hyperparameter optimization of machine learning algorithms: Theory and practice," Neurocomputing, vol. 415, 2020, doi: 10.1016/j.neucom.2020.07.061.

[15] A. Pfob, S. C. Lu, and C. Sidey-Gibbons, "Machine learning in medicine: a practical introduction to techniques for data pre-processing, hyperparameter tuning, and model comparison," BMC Med Res Methodol, vol. 22, no. 1, 2022, doi: 10.1186/s12874-022-01758-8.

[16] S. Katoch, S. S. Chauhan, and V. Kumar, "A review on genetic algorithm: past, present, and future," Multimed Tools Appl, vol. 80, no. 5, 2021, doi: 10.1007/s11042-020-10139-6.

[17] M. Dehghani, E. Trojovská, and P. Trojovský, "A new human-based metaheuristic algorithm for solving optimization problems on the base of simulation of driving training process," Sci Rep, vol. 12, no. 1, 2022, doi: 10.1038/s41598-022-14225-7.

[18] H. Su et al., "RIME: A physics-based optimization," Neurocomputing, vol. 532, 2023, doi: 10.1016/j.neucom.2023.02.010.

[19] M. Pota, M. Ventura, R. Catelli, and M. Esposito, "An effective bert-based pipeline for twitter sentiment analysis: A case study in Italian," Sensors (Switzerland), vol. 21, no. 1, 2021, doi: 10.3390/s21010133.

[20] A. S. Talaat, "Sentiment analysis classification system using hybrid BERT models," J Big Data, vol. 10, no. 1, 2023, doi: 10.1186/s40537-023-00781-w.

[21] M. Pota, M. Ventura, H. Fujita, and M. Esposito, "Multilingual evaluation of pre-processing for BERT-based sentiment analysis of tweets," Expert Syst Appl, vol. 181, 2021, doi: 10.1016/j.eswa.2021.115119.

[22] H. Y. Lin and T. S. Moh, "Sentiment analysis on COVID tweets using COVID-Twitter-BERT with auxiliary sentence approach," in Proceedings of the 2021 ACMSE Conference - ACMSE 2021: The Annual ACM Southeast Conference, 2021. doi: 10.1145/3409334.3452074.

[23] S. Palani, P. Rajagopal, and S. Pancholi, "T-BERT--Model for Sentiment Analysis of Micro-blogs Integrating Topic Model and BERT," arXiv preprint arXiv:2106.01097, 2021.

[24] A. Chiorrini, C. Diamantini, A. Mircoli, and D. Potena, "Emotion and sentiment analysis of tweets using BERT," in CEUR Workshop Proceedings, 2021.

[25] R. Anggrainingsih, G. M. Hassan, and A. Datta, "BERT based classification system for detecting rumours on Twitter," IEEE Access, vol. 11, pp. 80207-80217, 2023., 2021, doi: https://doi.org/10.1109/ACCESS.2023.3299858.

[26] Y. Wu, Z. Jin, C. Shi, P. Liang, and T. Zhan, "Research on the Application of Deep Learning-based BERT Model in Sentiment Analysis," arXiv preprint arXiv:2403.08217, 2024.

[27] H. A. Samir, L. Abd-Elmegid, and M. Marie, "Sentiment analysis model for Airline customers' feedback using deep learning techniques," 2023. doi: 10.1177/18479790231206019.

[28] S. Kardakis, I. Perikos, F. Grivokostopoulou, and I. Hatzilygeroudis, "Examining attention mechanisms in deep learning models for sentiment analysis," Applied Sciences (Switzerland), vol. 11, no. 9, 2021, doi: 10.3390/app11093883.

[29] S. Seo, C. Kim, H. Kim, K. Mo, and P. Kang, "Comparative Study of Deep Learning-Based Sentiment Classification," IEEE Access, vol. 8, 2020, doi: 10.1109/ACCESS.2019.2963426.

[30] M. Sedighimanesh, A. Sedighimanesh, and H. Zandhessami, "Optimizing Hyperparameters for Customer Churn Prediction with PSO-Enhanced Composite Deep Learning Techniques," Journal of Information Systems and Telecommunication (JIST), vol. 13, no. 50, pp. 91–110, 2025, doi: 10.61882/jist.48088.13.50.91.