A Population-based Plagiarism Detection using DistilBERT-Generated Word Embedding

Yuqin JING*, Ying LIU
College of Electronical and Information Engineering, Chongqing Open University
Chongqing 400052, China

***Abstract*—Plagiarism is the unacknowledged use of another person's language, information, or writing without crediting the source. This manuscript presents an innovative method for detecting plagiarism utilizing attention mechanism-based LSTM and the DistilBERT model, enhanced by an enriched differential evolution (DE) algorithm for pre-training and a focal loss function for training. DistilBERT reduces BERT's size by 40% while maintaining 97% of its language comprehension abilities and being 60% quicker. Current algorithms utilize positive-negative pairs to train a two-class classifier that detects plagiarism. A positive pair consists of a source sentence and a suspicious sentence, while a negative pair comprises two dissimilar sentences. Negative pairs typically outnumber positive pairs, leading to imbalanced classification and significantly lower system performance. To combat this, a training method based on a focal loss (FL) is suggested, which carefully learns minority class examples. Another addressed issue is the training phase, which typically uses gradient-based methods like back-propagation for the learning process. As a result, the training phase has limitations, such as initialization sensitivity. A new DE algorithm is proposed to initiate the back-propagation process by employing a mutation operator based on clustering. A successful cluster for the current DE population is found, and a fresh updating approach is used to produce potential solutions. The proposed method is assessed using three datasets: SNLI, MSRP, and SemEval2014. The model attains excellent results that outperform other deep models, conventional, and population-based models. Ablation studies excluding the proposed DE and focal loss from the model confirm the independent positive incremental impact of these components on model performance.**

***Keywords*—Plagiarism detection; LSTM; imbalanced classification; DistilBERT; differential evolution; focal loss**

I. INTRODUCTION

With abundant information available online and powerful search engines, plagiarism has become a sensitive issue in various domains, including education. Plagiarism usually occurs intentionally or unknowingly [1]. In contrast, plagiarism techniques have practical uses in fields beyond detecting copied content, including retrieval of information [2] where some text is given as input and the most relevant matches returned.

Various techniques have been proposed in academic publications to address the challenge of detecting plagiarism. One prominent approach is the use of text distance methods, which aim to quantify the semantic proximity between two textual pieces by measuring the distance between them. Typically, there are three categories of text distances: length distance, distribution distance, and semantic distance [3]. Length distance methods assess the resemblance between two texts by considering their numerical attributes. Popular techniques in this category include Euclidean distance, cosine distance, and Manhattan distance [4]. These methods rely on the numerical characteristics of the texts to calculate the degree of similarity. However, distance-based methods encounter two notable limitations. Firstly, they are often suitable only for symmetrical problems, which may restrict their applicability in certain scenarios. Additionally, using distance measures without considering the statistical characteristics of the data can be risky, particularly in cases such as question answering [5]. Distribution distances, on the other hand, offer an alternative approach to estimating the semantic similarity between two items by comparing their distributions. Techniques like Jensen–Shannon divergence [6] and Kullback–Leibler divergence [7] are commonly used in this category. These methods examine the lexical and semantic similarities between texts by analyzing the distributions of words or other linguistic features. By capturing the statistical properties of the data, distribution distances provide a more nuanced and comprehensive understanding of the semantic relationship between textual items. By leveraging distribution distances, researchers can effectively assess the similarity or dissimilarity between texts based on their underlying linguistic characteristics. These approaches take into account the broader context and semantic information, contributing to more accurate plagiarism detection.

Deep learning approaches have emerged as a powerful alternative to earlier methods in various fields, thanks to their inherent advantages, such as automated feature extraction [8]. Researchers have explored different deep learning architectures and techniques to tackle the task of sentence or text similarity and representation. One approach presented in [9] involves using a recurrent neural network (RNN) with word embeddings obtained from GloVe [10]. The RNN processes the words within a sentence and generates a representation of the sentence. Cosine distance metric [11] is then applied to measure the similarity between the sentence representations. In [40], a Siamese convolutional neural network (CNN) is introduced to capture the contextual information of individual words within a sentence. This network simultaneously produces a representation of word significance and the surrounding terms. By considering the local context, this approach aims to enhance the understanding of sentence meaning. Another RNN-based approach is presented in [12], where the textual data from corresponding words between sentence pairs is combined to create an internal representation.

This enables the model to capture the relationship between words in different sentences, contributing to a more comprehensive understanding of semantic similarity. In [13], a Long Short-Term Memory (LSTM) network is employed to extract high-level semantic information and measure the textual similarity between two sentences. The LSTM takes unprocessed pairs of sentence and word representations as input, allowing it to capture the complex semantic relationships within sentences. Attention-based models are also utilized in the pursuit of sentence similarity. In [14], an attention-based Siamese network is employed to determine the degree of similarity in meaning among sentences. The attention mechanism enables the model to focus on important elements within the sentences, enhancing its ability to capture semantic nuances. In [15], two different methods for answer selection based on similarity are introduced. One method incorporates a single transformer encoder along with embeddings from language models such as ELMo [16] and BERT [17]. The other method utilizes two pre-trained transformer encoders to capture the semantic information. Furthermore, [18] introduces the use of two Bidirectional LSTM (BLSTM) networks to independently derive sentence embeddings. Additionally, a revised data augmentation and loss function technique is implemented to address the challenge of imbalanced data distribution, which commonly occurs in sentence similarity tasks. One significant problem is the handling of imbalanced data distribution in plagiarism detection. The current algorithms often train two-class classifiers using positive-negative pairs, where negative pairs outnumber positive pairs. This imbalance negatively impacts system performance. Another limitation pertains to the training phase, which heavily relies on gradient-based methods like back-propagation. Although widely used, these methods have their own limitations, such as initialization sensitivity.

The unequal distribution of positive (plagiarized) and negative (non-plagiarized) cases pose a major obstacle in plagiarism detection. Failing to tackle this issue can result in a notable decline in performance. Approaches to tackle imbalanced class distribution can be categorized into two main types: the methods of the algorithm level and the data level. Data-level approaches aim to rectify the imbalanced distribution of classes by leveraging techniques such as over-sampling and under-sampling. One approach to address class imbalance is through the use of techniques such as the Synthetic Minority Oversampling Technique (SMOTE) [19], which creates instances by interpolating between adjacent minority examples. Another technique, NearMiss [20], involves under-sampling majority examples using the nearest neighbor algorithm. Over-sampling approaches might result in an overfitting problem, whereas applying under-sampling methods might lead to losing some helpful information about the dominant class. Algorithmic methods amplify the influence of the minority class based on techniques like ensemble learning [21], cost-sensitive learning [22], and decision threshold adjustment [23]. In the cost-sensitive approaches, various costs are assigned for misclassifications of different classes (higher costs for minority samples). The classification issue is framed as an optimization problem that seeks to minimize the total cost. Ensemble techniques train multiple classifiers and fuse the obtained results to reach a final decision. Threshold adjustment approaches involve training a classifier and then modifying the threshold for classification during testing. Imbalanced classification has also been addressed using deep learning techniques [22, 24]. The study [25] develops a method to learn distinguishing features in unbalanced data while preserving inter-cluster and interclass margins. In [26] the author proposes a strategy that bootstraps convolutional network data into balance for each mini-batch.

Neural network methods, including deep networks, are usually based on gradient-based methods, including back-propagation, to find the appropriate network weights. Regrettably, these techniques are prone to be influenced by the initialization of parameters and may converge to suboptimal solutions. The quality of a neural network can be more significantly influenced by the initial weights than by the network structure and training samples [27]. Meta-heuristic algorithms, including differential evolution (DE) [28], have been proposed as a solution to address these issues and have demonstrated their effectiveness in optimizing the performance of the model [29, 30].

Differential Evolution (DE) is a robust method successfully utilized in various optimization tasks [31, 32]. It comprises three primary steps: mutation to create an additional candidate solution using scaling differences between solutions, crossover to integrate the produced solution with the initial solution, and selection to select the optimal solution for the subsequent iteration. The mutation operator is particularly important [33].

This article describes an original approach to plagiarism detection that employs a DE algorithm and attention-based LSTM model. The proposed model contains a feed-forward network to estimate the similarity degree between sentences and two LSTMs for source and suspicious sentences. The model is trained using pairs of sentences, including positive pairs with two similar sentences and negative pairs with two dissimilar sentences. DistilBERT word embedding is utilized, which can reduce BERT's size by 40% while maintaining 97% of its language comprehension abilities and being 60% quicker. The proposed DE algorithm utilizes clustering for weight initialization, aiming to detect an area in the exploration domain suitable for initiating the back-propagation (BP) algorithm. The best-performing solution from the top-performing cluster is selected as the starting point for the mutation operator, and a new approach for generating potential solutions is employed. Additionally, the proposed algorithm incorporates FL to address class imbalance. The model is assessed on SNLI, MSRP, and SemEval2014 datasets, demonstrating superior performance compared to other methods.

The main contributions of the article are as follows: 1) The article introduces a new DE algorithm that initiates the back-propagation process by employing a mutation operator based on clustering. This approach helps overcome limitations associated with initialization sensitivity, which is a common issue in gradient-based methods used during the training phase, 2) The article addresses the challenge of imbalanced class distribution in plagiarism detection, where negative pairs outnumber positive pairs. The proposed training method based on focal loss enables the model to better learn from minority

class examples, leading to improved system performance, 3) The article introduces the DistilBERT model that can reduce the size of BERT. This reduction in size leads to improved efficiency, making the model 60% quicker compared to the original BERT model, and 4) Ablation studies are conducted to evaluate the individual contributions of the DE algorithm and focal loss. The results confirm that these components have an independent positive incremental impact on the model's performance.

The article's residual parts are structured as follows. Section II provides a number of contextual information, whereas Section III outlines the proposed method for identifying plagiarism. Section IV gives the prediction of the study made. In Section V, the results of the experiments are presented, and Section VI summarizes the paper.

II. Differential Evolution

Differential evolution [28] has effectively optimized various problems [35, 36]. DE starts with an initial population, usually drawn from a random distribution, and comprises three primary operations: mutation, crossover, and selection. The mutation operation generates a mutant vector as

$$\vec{v}_{i,g} = \vec{x}_{r_1,g} + F\,(\vec{x}_{r_2,g} - \vec{x}_{r_3,g}) \quad (1)$$

where $\vec{x}_{r_1,g}$, $\vec{x}_{r_2,g}$ and $\vec{x}_{r_3}$ are three randomly chosen candidate solutions from the available population, and F shows a factor scaling.

Crossover incorporates the mutant and target vectors. A well-known crossover operator is a binomial crossover, which does this as

$$u_{i,j,g} = \begin{cases} v_{i,j,g} \text{ if rand}(0,1) \leq \text{ CR or j } = j_{rand} \\ x_{i,j,g} \qquad \text{otherwise} \end{cases} \quad (2)$$

Where CR denotes the rate of crossover, and j_{rand} is a number chosen randomly from the set $\{1,2,\ldots,D\}$, where D is the dimensionality of a candidate solution.

Lastly, the selection operator elects the superior solution from the target and trial vectors.

III. Proposed Approach

The overall structure of the suggested method is displayed in Fig. 1. As seen, it comprises three main stages, pre-processing, word embedding, and prediction. First, redundant words and symbols are removed from the sentences. Next, the embedding vector of each word is obtained using BERT, and ultimately, the model predicts the similarity between the two sentences. The proposed model incorporates a clustering-based differential evolution algorithm to find the initial seeds of the network weights while using focal loss to handle class imbalance.

A. Pre-Processing

Data pre-processing is a crucial aspect of any NLP system as the fundamental characters, words, and sentences extracted in this phase are forwarded to the subsequent stages. Consequently, they considerably impact the outcome. Conversely, an unsuitable pre-processing technique can decrease the model's performance [37]. Common stop-word elimination and stemming techniques are used in the approach.

Stop words are part of sentences that can be regarded as overhead. The most common stop words are articles, prepositions, pronouns, etc. They should thus be removed as they cannot function as keywords in text mining applications [38] and decrease the number of dimensions in the term space

Stemming is employed to determine the base form of a word. For instance, the terms ‘watch’, ‘watched’, ‘watching’, ‘watcher’, etc., can all be reduced to the stem word "watch" by stemming. Stemming reduces ambiguity, decreases the number of words, and minimizes time and memory requirements [37].

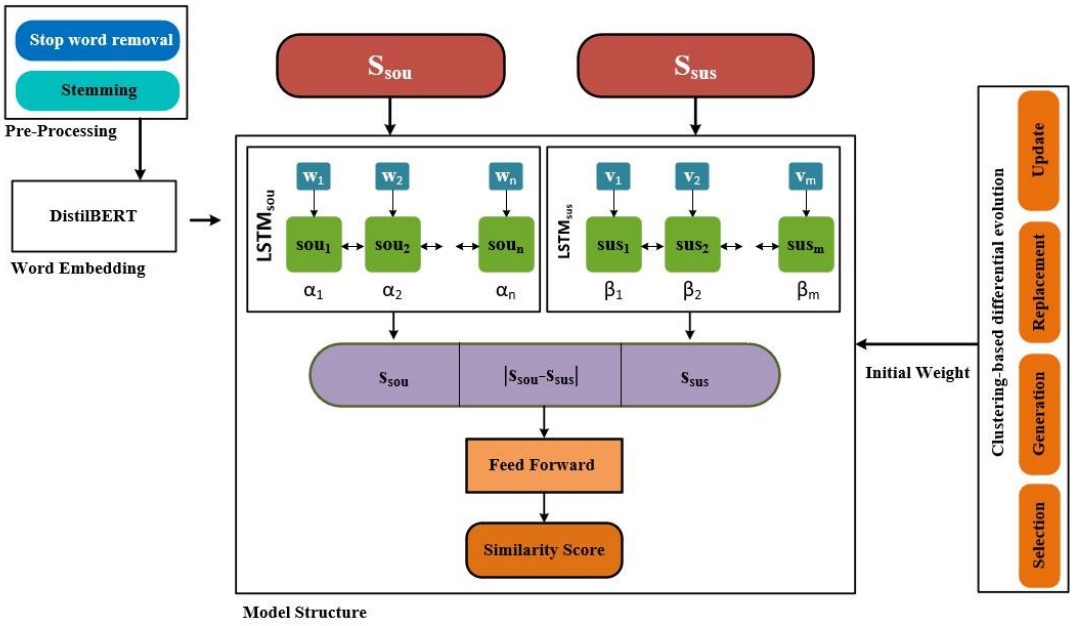

Fig. 1. Architecture of the suggested model as a whole.

IV. PREDICTION

The prediction model comprises two attention-based LSTM networks as extractors of the embeddings of the source and suspicious sentences and a feed-forward network as a predictor of the similarity of the two sentences. Given $s_{sus} = \{v_1, v_2, \dots, v_m\}$ and $s_{sou} = \{w_1, w_2, \dots, w_n\}$ as the sentence of the suspicious and source, where v_i and w_i denote the i-th word in the suspicious and source sentences, respectively. s_{sou} and s_{sus} are restricted to n and m words due to the length limitation in BLSTM (in the work, $n = m$). s_{sou} and s_{sus} are nourished separately into an LSTM network. These sentences embeddings are computed using the mechanism of attention, as

$$s_{sou} = \sum_{i=1}^{n} \alpha_i \, h_{sou_i} \quad (3)$$

And

$$s_{sus} = \sum_{i=1}^{m} \beta_i \, h_{sus_i} \quad (4)$$

where the i-th hidden vectors are represented in the BLSTM by $h_{sou_i} = [\overleftarrow{h}_{sou_i}, \vec{h}_{sou_i}]$ and $h_{sus_i} = [\overleftarrow{h}_{sus_i}, \vec{h}_{sus_i}]$, and the i-th attention weight for every section is shown in the BLSTM by $\alpha_i, \beta_i \in [0,1]$, computed as

$$\alpha_i = \frac{e^{u_i}}{\sum_{j=1}^{n} e^{u_j}} \quad (5)$$

And

$$\beta_i = \frac{e^{v_i}}{\sum_{j=1}^{m} e^{v_j}} \quad (6)$$

With

$$u_i = tanh(W_u h_{sou_i} + b_u) \quad (7)$$

And

$$v_i = tanh(W_v h_{sus_i} + b_v) \quad (8)$$

where W_u, W_v, b_u and b_v are the weight matrices and biases to the attention mechanisms. The fully-connected network's input is the connection of the s_{sou}, s_{sus} and $|s_{sou} - s_{sus}|$ as shown in Fig. 1. The dataset used for training consists of positive and negative pairs, where positive pairs contain a source sentence and a copied sentence and negative pairs comprise a source sentence and a different sentence.

The model has two training phases, pre-training and fine-tuning. In pre-training, an appropriate starting configuration is found. The weights obtained in pre-training are then the initial weights of the fine-tuning phase. In the pre-training phase, the enhanced differential evolution algorithm is employed.

A. Pre-Training

At this stage, the weights of the LSTM, attention mechanism, and feed-forward neural network are initialized. For this, an enhanced differential evolution method is introduced, boosted by a clustering scheme and a novel fitness function.

1) Clustering-based differential evolution: A clustering-based mutation and updating scheme is employed in the enhanced DE algorithm to improve the optimization performance.

The suggested mutation operator, which takes inspiration from [39] pinpoints a propitious area in the search space. The *k*-means clustering technique is used to partition the current population *P* into k clusters, each defining a distinct section of the search space. From [2, *N*], a random integer is chosen to depict the clusters number. The cluster with the lowest mean fitness of its samples is the best cluster after clustering.

The suggested mutation based on clustering is described as follows:

$$\overrightarrow{v^{clu}}_{\iota} = \overrightarrow{win}_g + F\,(\vec{x}_{r_1,g} - \vec{x}_{r_2,g}) \quad (9)$$

where $\overrightarrow{win}_g$ is the most acceptable solution in the promising region, and $\vec{x}_{r_1,g}$ and $\vec{x}_{r_2,g}$ are two randomly determined solutions from the available population. It should be noted that wing is not always the population's most acceptable solution. The procedure of the mutation on the basis of the clustering is implemented *M* times.

When *M* new solutions have been provoked through clustering-based mutation, the current population is updated. The steps are as follows:

- Selection: Generate k individuals randomly as the starting points of *k*-means;
- Generation: Generate the solutions of the*M* by applying clustering-based mutation as the collection v^{clu};
- Replacement: Choose *M* solutions at random and determine as *B*;
- Update: The best *M* solutions from the $v^{clu} \cup B$ determined as the B'. The novel population is afterwards calculated as $(P - B) \cup B'$.

2) Encoding strategy: The primary structure of the proposed model includes two LSTM networks along with their attention mechanisms and a feed-forward network. As illustrated in Fig. 2, all weights and bias terms are arranged into a vector to form a candidate solution in the proposed DE algorithm.

3) Fitness function: To calculate the quality of a candidate solution, the fitness function is as

$$Fitness = \frac{1}{\sum_{i=1}^{N}(y_i - \tilde{y}_i)^2} \quad (10)$$

Where *N* is the number of training examples, y_i and $\tilde{y}_i$ show the i-th target and output predicted by the model, respectively.

B. Focal Loss

The plagiarism problem is defined as a two-class classification problem based on positive and negative classes. As an imbalanced problem, with few samples in the negative class, focal loss (FL) [34] is used to address this.

FL is a modification of binary cross-entropy (CE) that focuses training on harder (i.e., minority class) samples [40].CE is defined as

$$CE = \begin{cases} -log\,(p), & y = 1 \\ -log(1-p), & \text{otherwise} \end{cases} \quad (11)$$

where $y \in \{-1,1\}$ is the actual class label, and $p \in [0,1]$ is the predicted probability of the model for the class with target $y = 1$. The probability is

$$p_t = \begin{cases} -p, & y = 1 \\ 1-p, & \text{otherwise} \end{cases} \quad (12)$$

and hence

$$CE(p,y) = CE(p_t) = -\log(p_t) \quad (13)$$

FL tends to add a modulating component to cross-entropy loss, leading to

$$FL(p_t) = -\alpha_t(1-p_t)^{\gamma} \log(p_t) \quad (14)$$

Where $\gamma > 0$ (if $\gamma = 1$, then FL is similar to CE loss), and $\alpha \in [0,1]$ is the inverse class frequency.

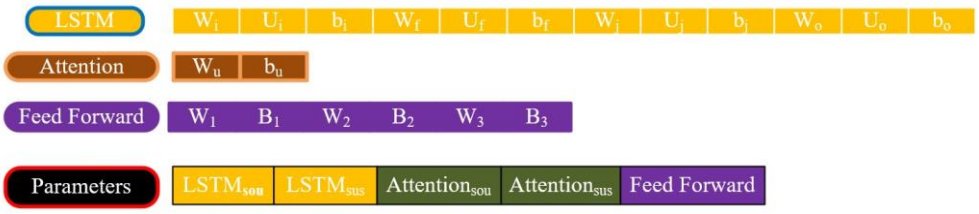

Fig. 2. Encoding strategy in the proposed algorithm.

V. Results

A. Datasets

In the tests, the following three benchmark datasets are utilized:

- SNLI: the Stanford Natural Language Inference (SNLI) corpus [41] is a large dataset consisting of pairs of labelled sentences with three classes, including contradiction, entailment, and semantic independence. It comprises 550,152 sentence pairs for training and 10,000 pairs of sentences each for testing and validation.
- MSRP: A rephrasing set of data from Internet news articles called the Microsoft Research Paraphrase Corpus (MSRP) [42], divided into training and testing, with pairs of positive and negative sentences by several experts. Of the whole collection, about 67% of paraphrases are present. The test and training datasets have 1,726 and 4,076 examples, respectively, out of which 1,147 and 2,753 are paraphrases, respectively.
- SemEval2014: the Semantic Evaluation Database (SemEval) [13] is a widely-used benchmark for evaluating STS, presented in various versions. The Compositional Knowledge (SICK) dataset [43] from 2014 is employed to assess the semantic similarity of sentences. The dataset includes 10,000 sentence pairs, distributed as 4,500 pairs for training, 500 for validation, and 5,000 for testing.

B. Model Performance

The algorithm is compared to seven deep learning methods, namely RNN [9], Siamese CNN+LSTM [44] , CA-RNN [14], AttSiaBiLSTM [13], LSTM+FNN+attention [14], CETE [15] and STS-AM [18]. The results are given in Tables I, II and III for SNLI, MSRP, and SemEval2014, correspondingly. For the suggested approach, outcomes are presented based on accidental weight initialization, the use of FL, and the full proposed model. The proposed model demonstrates superior performance compared to other models, including CETE, the best-performing competitor, across all metrics for SNLI. The error rate is reduced by over 50% and 54% in the two primary metrics, F-measure and G-means. Comparing the proposed model with Proposed+random weights and Proposed+random weights+FL, the mistake percentage is reduced by around 67%, highlighting the significance of improved DE and FL methodologies. The proposed model achieved the most significant improvement for the MSRP dataset, followed by the CETE algorithm. The error rate improvement for this dataset is about 27.41% and 26.69% for both the F-measure and G-means criteria, respectively. In the SemEval2014 dataset, the proposed method reduces the classification mistake by over 18% and 37% compared to CETE and STS-AM, respectively.

TABLE I. Comparative Performance of Deep Learning Models on the SNLI Dataset. The Performance Metrics are Represented in Fractions, Multiplied by 10^(-3), for Concise Presentation

Model	Accuracy	Recall	Precision	F-measure	G-means
RNN [9]	687×10^{-3}	594×10^{-3}	540×10^{-3}	566×10^{-3}	661×10^{-3}
Siamese CNN+LSTM [44]	850×10^{-3}	763×10^{-3}	792×10^{-3}	777×10^{-3}	826×10^{-3}
CA-RNN [14]	790×10^{-3}	667×10^{-3}	704×10^{-3}	685×10^{-3}	754×10^{-3}
AttSiaBiLSTM [13]	695×10^{-3}	569×10^{-3}	554×10^{-3}	561×10^{-3}	658×10^{-3}
LSTM+FNN+attention [14]	818×10^{-3}	781×10^{-3}	715×10^{-3}	747×10^{-3}	809×10^{-3}
CETE [15]	874×10^{-3}	855×10^{-3}	795×10^{-3}	824×10^{-3}	870×10^{-3}
STS-AM [18]	756×10^{-3}	625×10^{-3}	650×10^{-3}	637×10^{-3}	718×10^{-3}
Proposed+random weights	808×10^{-3}	777×10^{-3}	698×	735×	801×
Proposed+random weights+FL	815×10^{-3}	784×10^{-3}	708×	744×	808×
Proposed	930×10^{-3}	920×10^{-3}	881×	900×	927×

TABLE II. COMPARATIVE PERFORMANCE OF DEEP LEARNING MODELS ON THE MSRP DATASET. THE PERFORMANCE METRICS ARE REPRESENTED IN FRACTIONS, MULTIPLIED BY 10^(-3), FOR CONCISE PRESENTATION

Model	Accuracy	Recall	Precision	F-measure	G-means
RNN [9]	853×10^{-3}	922×10^{-3}	866×10^{-3}	893×10^{-3}	812×10^{-3}
Siamese CNN+LSTM [44]	863×10^{-3}	916×10^{-3}	882×10^{-3}	899×10^{-3}	833×10^{-3}
CA-RNN [14]	880×10^{-3}	928×10^{-3}	896×10^{-3}	912×10^{-3}	854×10^{-3}
AttSiaBiLSTM [13]	874×10^{-3}	927×10^{-3}	889×10^{-3}	908×10^{-3}	845×10^{-3}
LSTM+FNN+attention [14]	889×10^{-3}	917×10^{-3}	916×10^{-3}	916×10^{-3}	873×10^{-3}
CETE [15]	916×10^{-3}	949×10^{-3}	926×10^{-3}	937×10^{-3}	898×10^{-3}
STS-AM [18]	899×10^{-3}	940×10^{-3}	910×10^{-3}	925×10^{-3}	876×10^{-3}
Proposed+random weights	875×10^{-3}	908×10^{-3}	905×10^{-3}	906×10^{-3}	858×10^{-3}
Proposed+randomweights+FL	895×10^{-3}	926×10^{-3}	917×10^{-3}	921×10^{-3}	879×10^{-3}
Proposed	937×10^{-3}	961×10^{-3}	946×10^{-3}	953×10^{-3}	925×10^{-3}

TABLE III. COMPARATIVE PERFORMANCE OF DEEP LEARNING MODELS ON THE SEMEVAL2014 DATASET. THE PERFORMANCE METRICS ARE REPRESENTED IN FRACTIONS, MULTIPLIED BY 10^(-3), FOR CONCISE PRESENTATION

Model	Accuracy	Recall	Precision	F-measure	G-means
RNN [9]	809×10^{-3}	822×10^{-3}	963×10^{-3}	887×10^{-3}	750×10^{-3}
Siamese CNN+LSTM [44]	775×10^{-3}	787×10^{-3}	958×10^{-3}	864×10^{-3}	720×10^{-3}
CA-RNN [14]	811×10^{-3}	826×10^{-3}	961×10^{-3}	888×10^{-3}	742×10^{-3}
AttSiaBiLSTM [13]	799×10^{-3}	816×10^{-3}	957×10^{-3}	881×10^{-3}	720×10^{-3}
LSTM+FNN+attention [14]	733×10^{-3}	746×10^{-3}	949×10^{-3}	835×10^{-3}	670×10^{-3}
CETE [15]	854×10^{-3}	868×10^{-3}	969×10^{-3}	916×10^{-3}	791×10^{-3}
STS-AM [18]	823×10^{-3}	834×10^{-3}	966×10^{-3}	895×10^{-3}	768×10^{-3}
Proposed +random weights	839×10^{-3}	855×10^{-3}	964×10^{-3}	906×10^{-3}	766×10^{-3}
Proposed +random weights+FL	849×10^{-3}	863×10^{-3}	967×10^{-3}	912×10^{-3}	783×10^{-3}
Proposed	876×10^{-3}	884×10^{-3}	977×10^{-3}	928×10^{-3}	838×10^{-3}

C. Comparison with other Metaheuristics

The enhanced DE algorithm is contrasted with several metaheuristic optimization algorithms in the subsequent experiment. Different metaheuristics are used to obtain the initial model parameters while keeping the same as the other model components, i.e., pre-processing, word embedding, LSTM and network structure, and loss function. Eight different algorithms, namely (standard) DE [45], FA [46], BA [47], COA [48], ABC [49], GWO [50], WOA [51], and SSA [52], are used. The obtained results are reported in Tables IV, V and VI for the SNLI, MSRP, and SemEval2014 datasets, respectively. For the SNLI dataset, the suggested model reduces error by about 44% compared to the standard DE. It clearly shows that the proposed model has a substantial ability compared to the standard one. Also, DE offers more acceptable results than other algorithms, including ABC, GWO, and BAT. There is a minor improvement for the other two datasets, so the error rate for MSRP and SemEval2014 is reduced by around 19.17% and 8.82%, respectively.

D. Word Embeddings

Word embedding is a crucial component of In-depth learning-based models since the input is read as a vector, and if the embedding is erroneous, the model might be misled. This study used the DistilBERT model as a word embedding, one of the most recent embedding models. Five more-word embeddings are used to compare various word embeddings to the model: One-Hot encoding One-Hot encoding [53], CBOW, Skip-gram [54], GloVe [10] ,and FastText [55]. One-Hot encoding is a crucial step in changing the collected data variables fed to In-depth learning methods, enhancing the accuracy of predictions and classifications. It generates a binary feature for every class, and each sample's feature is given a value of 1 corresponding to its original class. Skip-gram and CBOW are techniques that transform a word into its corresponding representation vector using neural networks. The GloVe is a method for aggregating global word-word co-occurrence data from a corpus. The Skip-gram paradigm is expanded by the word embedding technique known as FastText. This approach encodes each word as an n-gram of letters rather than learning word vectors. The outcomes of this experiment can be found in Tables VII, VIII and IX for the SNLI, MSRP, and SemEval2014 datasets, respectively. The worst-performing word embedding method was One Hot encoding. In the MSRP dataset, the proposed model showed an improvement of approximately 85.81% and 83.51% for the two criteria, F-measure and G-means, respectively. Skip-gram and CBOW operate nearly similarly across the three datasets because of similar architecture, which is superior to the Glove model. FastText performs better than other models but poorly on BERT. The error rate is reduced by more than 18%, 15%, and 24% for the SNLI, MSRP, and SemEval2014 datasets, respectively, when utilizing BERT instead of FastText.

E. Loss Functions

Finally, to justify the selection of focal loss in the approach, the comparison is made with four other loss functions, namely weighted cross-entropy (WCE) [56], balanced cross-entropy (BCE) [57], Dice loss (DL) [58], and Tversky loss (TL) [59].

The results of these experiments are given in Tables X, XI and XII for the SNLI, MSRP, and SemEval2014 datasets, respectively. The use of focal loss gives the best results for all measures on the SNLI and MSRP datasets and yields the best G-means results for all three datasets. The results of this experiment are given in Tables X, XI and XII for the SNLI, MSRP, and SemEval2014 datasets, respectively. Generally speaking, the reduction of FL error compared to TL for SNLI and MSRP datasets is about 19% and 27%. However, these two functions are slightly different in the SemEval2014 dataset, so the improvement rate for this dataset is about 12%.

F. Examples

A qualitative example is provided to demonstrate the important contributions of both the improved DE algorithm and the use of FL in the approach. The source sentence "Two people are kickboxing, and spectators are watching" from the SemEval2014 dataset is used for this purpose. Fig. 3 gives the results of the top five sentences retrieved by the BPD model with random weight initialization and focal loss, without FL, and the full approach. As is apparent, the full model extracts suspicious sentences most similar to the source sentence, while the other two models retrieve these only in the lower rankings.

TABLE IV. COMPARATIVE PERFORMANCE OF METAHEURISTIC ALGORITHMS ON THE SNLI DATASET. THE PERFORMANCE METRICS ARE REPRESENTED IN FRACTIONS, MULTIPLIED BY 10^(-3), FOR CONCISE PRESENTATION

Algorithm	Accuracy	Recall	Precision	F-measure	G-means
DE	897×10^{-3}	889×10^{-3}	824×10^{-3}	855×10^{-3}	895×10^{-3}
FA	864×10^{-3}	803×10^{-3}	801×10^{-3}	802×10^{-3}	848×10^{-3}
BA	876×10^{-3}	850×10^{-3}	801×10^{-3}	825×10^{-3}	870×10^{-3}
COA	860×10^{-3}	811×10^{-3}	787×10^{-3}	799×10^{-3}	847×10^{-3}
ABC	885×10^{-3}	869×10^{-3}	809×10^{-3}	838×10^{-3}	881×10^{-3}
GWO	842×10^{-3}	780×10^{-3}	763×10^{-3}	771×10^{-3}	826×10^{-3}
WOA	883×10^{-3}	832×10^{-3}	828×10^{-3}	830×10^{-3}	870×10^{-3}
SSA	863×10^{-3}	820×10^{-3}	789×10^{-3}	804×10^{-3}	852×10^{-3}

TABLE V. COMPARATIVE PERFORMANCE OF METAHEURISTIC ALGORITHMS ON THE MSRP DATASET. THE PERFORMANCE METRICS ARE REPRESENTED IN FRACTIONS, MULTIPLIED BY 10^(-3), FOR CONCISE PRESENTATION

Algorithm	Accuracy	Recall	Precision	F-measure	G-means
DE	925×10^{-3}	959×10^{-3}	930×10^{-3}	944×10^{-3}	906×10^{-3}
FA	897×10^{-3}	942×10^{-3}	908×10^{-3}	925×10^{-3}	874×10^{-3}
BA	910×10^{-3}	944×10^{-3}	922×10^{-3}	933×10^{-3}	892×10^{-3}
COA	902×10^{-3}	936×10^{-3}	918×10^{-3}	927×10^{-3}	884×10^{-3}
ABC	899×10^{-3}	946×10^{-3}	906×10^{-3}	926×10^{-3}	873×10^{-3}
GWO	884×10^{-3}	926×10^{-3}	902×10^{-3}	914×10^{-3}	861×10^{-3}
WOA	901×10^{-3}	938×10^{-3}	915×10^{-3}	926×10^{-3}	881×10^{-3}
SSA	890×10^{-3}	929×10^{-3}	908×10^{-3}	918×10^{-3}	869×10^{-3}

TABLE VI. COMPARATIVE PERFORMANCE OF METAHEURISTIC ALGORITHMS ON THE SEMEVAL2014 DATASET. THE PERFORMANCE METRICS ARE REPRESENTED IN FRACTIONS, MULTIPLIED BY 10^(-3), FOR CONCISE PRESENTATION

Algorithm	Accuracy	Recall	Precision	F-measure	G-means
DE	864×10^{-3}	873×10^{-3}	975×10^{-3}	921×10^{-3}	822×10^{-3}
FA	854×10^{-3}	865×10^{-3}	972×10^{-3}	915×10^{-3}	807×10^{-3}
BA	860×10^{-3}	869×10^{-3}	973×10^{-3}	918×10^{-3}	814×10^{-3}
COA	856×10^{-3}	867×10^{-3}	971×10^{-3}	916×10^{-3}	805×10^{-3}
ABC	851×10^{-3}	863×10^{-3}	970×10^{-3}	913×10^{-3}	796×10^{-3}
GWO	848×10^{-3}	857×10^{-3}	972×10^{-3}	911×10^{-3}	805×10^{-3}
WOA	844×10^{-3}	856×10^{-3}	969×10^{-3}	909×10^{-3}	788×10^{-3}
SSA	843×10^{-3}	857×10^{-3}	966×10^{-3}	908×10^{-3}	777×10^{-3}

TABLE VII. COMPARATIVE PERFORMANCE OF WORD EMBEDDINGS ON THE SNLI DATASET. THE PERFORMANCE METRICS ARE REPRESENTED IN FRACTIONS, MULTIPLIED BY 10^(-3), FOR CONCISE PRESENTATION

Word embedding	Accuracy	Recall	Precision	F-measure	G-means
One-Hot encoding	650×10^{-3}	473×10^{-3}	489×10^{-3}	481×10^{-3}	592×10^{-3}
CBOW	856×10^{-3}	779×10^{-3}	796×10^{-3}	787×10^{-3}	835×10^{-3}
Skip-gram	871×10^{-3}	817×10^{-3}	808×10^{-3}	812×10^{-3}	857×10^{-3}
GloVe	845×10^{-3}	798×10^{-3}	762×10^{-3}	780×10^{-3}	833×10^{-3}
FastText	905×10^{-3}	861×10^{-3}	861×10^{-3}	861×10^{-3}	893×10^{-3}
BERT	912×10^{-3}	892×10^{-3}	910×10^{-3}	886×10^{-3}	902×10^{-3}

TABLE VIII. COMPARATIVE PERFORMANCE OF WORD EMBEDDINGS ON THE MSRP DATASET. THE PERFORMANCE METRICS ARE REPRESENTED IN FRACTIONS, MULTIPLIED BY 10^(-3), FOR CONCISE PRESENTATION

Word embedding	Accuracy	Recall	Precision	F-measure	G-means
One-Hot encoding	604×10^{-3}	659×10^{-3}	721×10^{-3}	689×10^{-3}	571×10^{-3}
CBOW	802×10^{-3}	840×10^{-3}	859×10^{-3}	849×10^{-3}	781×10^{-3}
Skip-gram	830×10^{-3}	856×10^{-3}	884×10^{-3}	870×10^{-3}	816×10^{-3}
GloVe	781×10^{-3}	824×10^{-3}	844×10^{-3}	834×10^{-3}	758×10^{-3}
FastText	864×10^{-3}	880×10^{-3}	913×10^{-3}	896×10^{-3}	857×10^{-3}
BERT	912×10^{-3}	900×10^{-3}	922×10^{-3}	910×10^{-3}	913×10^{-3}

TABLE IX. COMPARATIVE PERFORMANCE OF WORD EMBEDDINGS ON THE SEMEVAL2014 DATASET. THE PERFORMANCE METRICS ARE REPRESENTED IN FRACTIONS, MULTIPLIED BY 10^(-3), FOR CONCISE PRESENTATION

Word embedding	Accuracy	Recall	Precision	F-measure	G-means
One-Hot encoding	504×10^{-3}	529×10^{-3}	875×10^{-3}	659×10^{-3}	367×10^{-3}
CBOW	749×10^{-3}	768×10^{-3}	946×10^{-3}	848×10^{-3}	659×10^{-3}
Skip-gram	758×10^{-3}	773×10^{-3}	951×10^{-3}	853×10^{-3}	686×10^{-3}
GloVe	697×10^{-3}	715×10^{-3}	936×10^{-3}	811×10^{-3}	606×10^{-3}
FastText	812×10^{-3}	826×10^{-3}	961×10^{-3}	888×10^{-3}	742×10^{-3}
BERT	842×10^{-3}	862×10^{-3}	970×10^{-3}	901×10^{-3}	763×10^{-3}

TABLE X. COMPARATIVE PERFORMANCE OF LOSS FUNCTION ON THE SNLI DATASET. THE PERFORMANCE METRICS ARE REPRESENTED IN FRACTIONS, MULTIPLIED BY 10^(-3), FOR CONCISE PRESENTATION

Loss function	Accuracy	Recall	Precision	F-measure	G-means
WCE	871×10^{-3}	856×10^{-3}	788×10^{-3}	821×10^{-3}	868×10^{-3}
BCE	895×10^{-3}	874×10^{-3}	828×10^{-3}	850×10^{-3}	890×10^{-3}
DL	915×10^{-3}	885×10^{-3}	870×10^{-3}	877×10^{-3}	908×10^{-3}
TL	905×10^{-3}	880×10^{-3}	848×10^{-3}	864×10^{-3}	899×10^{-3}

TABLE XI. COMPARATIVE PERFORMANCE OF LOSS FUNCTION ON THE MSRP DATASET. THE PERFORMANCE METRICS ARE REPRESENTED IN FRACTIONS, MULTIPLIED BY 10^(-3), FOR CONCISE PRESENTATION

Loss function	Accuracy	Recall	Precision	F-measure	G-means
WCE	861×10^{-3}	906×10^{-3}	887×10^{-3}	896×10^{-3}	836×10^{-3}
BCE	883×10^{-3}	923×10^{-3}	903×10^{-3}	913×10^{-3}	861×10^{-3}
DL	915×10^{-3}	943×10^{-3}	930×10^{-3}	936×10^{-3}	899×10^{-3}
TL	899×10^{-3}	933×10^{-3}	917×10^{-3}	925×10^{-3}	881×10^{-3}

TABLE XII. COMPARATIVE PERFORMANCE OF LOSS FUNCTION ON THE SEMEVAL2014 DATASET. THE PERFORMANCE METRICS ARE REPRESENTED IN FRACTIONS, MULTIPLIED BY 10^(-3), FOR CONCISE PRESENTATION

Loss function	Accuracy	Recall	Precision	F-measure	G-means
WCE	876×10^{-3}	904×10^{-3}	957×10^{-3}	930×10^{-3}	736×10^{-3}
BCE	875×10^{-3}	899×10^{-3}	960×10^{-3}	928×10^{-3}	754×10^{-3}
DL	877×10^{-3}	890×10^{-3}	972×10^{-3}	929×10^{-3}	815×10^{-3}
TL	876×10^{-3}	895×10^{-3}	966×10^{-3}	929×10^{-3}	784×10^{-3}

rank	Proposed+random weights+RL	Proposed without RL	Proposed
1	**Two people are** wading through the water	**Two people are** riding a motorcycle	**Two people are** fighting **and spectators are watching**
2	A few men **are watching** cricket	**Two people are** fighting **and spectators are watching**	Two **spectators are kickboxing and** some people **are watching**
3	Two girls are laughing breathlessly and other girls **are watching** them	Two adults are sitting in the chairs and **are watching** the ocean	Two young women are sparring in a **kickboxing** fight
4	**Two people** wearing snowsuits are on the ground making snow angels	Two **spectators are kickboxing and** some people **are watching**	Two women are **sparring** in a **kickboxing** match
5	Two **spectators are kickboxing and** some people **are watching**	A few men **are watching** cricket	**Two people are** wading through the water

Fig. 3. Top-ranked suspicious sentences for source sentence "Two people are kickboxing, and spectators are watching." Words that appear in the source sentence are bolded.

G. Discussion

The article presented an innovative approach to plagiarism detection by using an attention mechanism-based LSTM and the DistilBERT model. The utilization of the DistilBERT model is particularly notable as it reduces the size of the original BERT model by 40% while maintaining 97% of its language comprehension capabilities and increasing the speed by 60%. Two novel approaches were introduced to enhance the overall performance of the system. First, a focal loss function was used to address the issue of imbalanced classification, which often occurs when negative pairs significantly outnumber positive pairs. Second, an enriched DE algorithm was introduced to address the limitations associated with traditional gradient-based learning methods, such as initialization sensitivity. The approach was evaluated on three benchmark datasets: SNLI, MSRP, and SemEval2014, and it outperformed other deep models and conventional and population-based models. The effectiveness of the DE algorithm and the focal loss function was further validated through ablation studies.

While the study exhibits considerable promise, there are a few potential limitations:

- The study indeed leverages benchmark datasets that provide reliable standards for performance evaluation. They serve as a crucial starting point for developing and refining the model, and their use enables the results to be compared directly with other models that have also been evaluated using these datasets. However, these datasets, although comprehensive and widely used, may not fully encapsulate the full diversity and complexity of real-world plagiarism scenarios. Real-world plagiarism can be exceptionally intricate, involving subtle paraphrasing, strategic insertion of synonyms, reordering of sentences, or blending of original and copied material, among other tactics. These practices can often deceive conventional plagiarism detection tools, requiring models that can comprehend and identify such complex forms of plagiarism. Furthermore, these benchmark datasets might lack certain forms of plagiarism seen in specific fields or cultures. Plagiarism, after all, can differ greatly across different academic disciplines, professional fields, and cultural contexts. For instance, the plagiarism practices in a literature research paper could be entirely different from those in a technical report in engineering. Lastly, the real-world plagiarism scenarios are continually evolving, influenced by the advancement of technology and changes in writing and copying techniques. The dynamic and constantly changing nature of real-world plagiarism can present a challenge that these static, fixed datasets may not be fully equipped to address.
- The incorporation of DistilBERT is indeed a step forward in reducing the model's size and enhancing its operational speed, owing to its design that maintains substantial language comprehension capabilities while being considerably smaller and faster than the original BERT model. This makes the model more feasible for applications that demand quicker processing times and limited memory capacities. However, even with these benefits, the combined system that also includes the attention mechanism-based LSTM and the enriched DE algorithm may still have significant computational demands. The LSTM component, known for its ability to remember long-term dependencies in sequence data, can be computationally intensive, particularly for longer sequences or larger datasets. The recurrent nature of LSTMs, where outputs from one step are fed as inputs to the next, makes parallelization of computations difficult, potentially slowing down the training process. On the other hand, the DE algorithm, while providing an innovative solution to the limitation of sensitivity to initialization inherent in gradient-based training methods, adds another layer of complexity to the system. The operations involved in differential evolution, such as mutation, recombination, and selection, while aiding the optimization process, also contribute to the computational burden. Moreover, the system must be trained on multiple iterations to effectively learn from the data, and each iteration involves processing the entire dataset. The computational demands can, therefore, escalate with the volume of data, length of sequences, and complexity of the tasks at hand. In the real world, these requirements could translate to higher memory and processing power requirements, extended training times, and increased energy consumption. They could also limit the system's deployability on devices with limited computational capabilities, such as mobile devices or low-end personal computers. Therefore, while the use of DistilBERT, LSTM, and the DE algorithm offers various advantages, further work could be directed towards optimizing the system to make it more efficient and less resource-intensive.

Finally, future works that can be considered are as follows:

- Investigating the model's performance on other languages beyond those in the current datasets could be beneficial, potentially leading to the development of a more universally applicable plagiarism detection tool.
- It would be valuable to test the model in real-world scenarios, such as academic papers or professional reports, to further assess its effectiveness and robustness.
- There may be scope to optimize the DE algorithm further for this specific use case. Tuning the parameters of the mutation operator based on the characteristics of the plagiarism detection task could potentially enhance the system's performance.
- Exploring the use of other pre-trained language models, including GPT-3 and T5, and compare their performance with DistilBERT. Comparing the capabilities of multiple pre-trained language models such as GPT-3 and T5 for plagiarism detection tasks could provide valuable insights into the suitability of these models for this application. GPT-3 is a transformer-based language model trained on a massive corpus of diverse texts and has shown impressive

results in a variety of natural language processing tasks. T5, on the other hand, is a text-to-text transformer that can be fine-tuned for different tasks, including text classification and sequence labeling.

VI. Conclusion

Plagiarism is the unacknowledged use of another individual's language, information, or writing without crediting the source. An innovative model was introduced to detect plagiarism based on DistilBERT word embeddings, an LSTM approach with an attention mechanism, and an enhanced DE algorithm used for pre-training the networks. To address the issue of inherent class imbalance, focal loss was employed. The enhanced DE algorithm groups the present population to pinpoint a potential area within the search space and integrates a novel update mechanism. DistilBERT can improve the performance of BERT by 40% and 97% in terms of size, language comprehension abilities, and speed, respectively. Extensive experiments on three datasets confirm the approach to yield excellent performance, outperforming various plagiarism detection approaches. The DE algorithm is superior to several other meta-heuristic methods. In forthcoming studies, the plan is to utilize the technique on several deep models, and an investigation of a version of the algorithm that can handle multiple objectives is underway.

Acknowledgment

This work was supported by the Science and Technology Research Program of the Chongqing Municipal Education Commission of China. (Grant No. KJQN202004002) , and the Science and Technology Research Program of Chongqing Municipal Education Commission of China. (Grant No. KJQN202104008).

References

[1] F. Khaled, M.S.H. Al-Tamimi, Plagiarism detection methods and tools: An overview, Iraqi Journal of Science. (2021), pp.2771–2783.

[2] R. Zhu, X. Tu, J.X. Huang, Deep learning on information retrieval and its applications, in: Deep Learning for Data Analytics, Elsevier, 2020, pp. 125–153.

[3] J. Wang, Y. Dong, Measurement of text similarity: a survey, Information. 11 ,p.421, (2020).

[4] A. Mahmoud, M. Zrigui, Semantic similarity analysis for paraphrase identification in Arabic texts, in: Proceedings of the 31st Pacific Asia Conference on Language, Information and Computation, 2017, pp. 274–281.

[5] E. Deza, M.M. Deza, M.M. Deza, E. Deza, Encyclopedia of distances, Springer, 2009.

[6] M.L. Menéndez, J.A. Pardo, L. Pardo, M.C. Pardo, The jensen-shannon divergence, J Franklin Inst. 334 (1997) , pp.307–318.

[7] T. Van Erven, P. Harremos, Rényi divergence and Kullback-Leibler divergence, IEEE Trans Inf Theory. 60, (2014),pp. 3797–3820.

[8] S.V. Moravvej, A. Mirzaei, M. Safayani, Biomedical text summarization using conditional generative adversarial network (CGAN), ArXiv Preprint ArXiv:2110.11870. (2021).

[9] A. Sanborn, J. Skryzalin, Deep learning for semantic similarity, CS224d: Deep Learning for Natural Language Processing Stanford, CA, USA: Stanford University. (2015).

[10] J. Pennington, R. Socher, C.D. Manning, Glove: Global vectors for word representation, in: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2014, pp. 1532–1543.

[11] F. Rahutomo, T. Kitasuka, M. Aritsugi, Semantic cosine similarity, in: The 7th International Student Conference on Advanced Science and Technology ICAST, 2012, p. 1.

[12] Q. Chen, Q. Hu, J.X. Huang, L. He, CA-RNN: using context-aligned recurrent neural networks for modeling sentence similarity, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2018.

[13] W. Bao, W. Bao, J. Du, Y. Yang, X. Zhao, Attentive Siamese LSTM network for semantic textual similarity measure, in: 2018 International Conference on Asian Language Processing (IALP), IEEE, 2018, pp. 312–317.

[14] Z. Chi, B. Zhang, A sentence similarity estimation method based on improved siamese network, Journal of Intelligent Learning Systems and Applications. 10, (2018), pp.121–134.

[15] M.T.R. Laskar, X. Huang, E. Hoque, Contextualized embeddings based transformer encoder for sentence similarity modeling in answer selection task, in: Proceedings of the Twelfth Language Resources and Evaluation Conference, 2020, pp. 5505–5514.

[16] J. Sarzynska-Wawer, A. Wawer, A. Pawlak, J. Szymanowska, I. Stefaniak, M. Jarkiewicz, L. Okruszek, Detecting formal thought disorder by deep contextualized word representations, Psychiatry Res. 304, p.114135, (2021).

[17] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, ArXiv Preprint ArXiv:1810.04805. (2018).

[18] S.V. Moravvej, M.J.M. Kahaki, M.S. Sartakhti, A. Mirzaei, A method based on attention mechanism using bidirectional long-short term memory (BLSTM) for question answering, in: 2021 29th Iranian Conference on Electrical Engineering (ICEE), IEEE, 2021, pp. 460–464.

[19] H. Han, W.-Y. Wang, B.-H. Mao, Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning, in: Advances in Intelligent Computing: International Conference on Intelligent Computing, ICIC 2005, Hefei, China, August 23-26, 2005, Proceedings, Part I 1, Springer, 2005, pp. 878–887.

[20] I. Mani, I. Zhang, kNN approach to unbalanced data distributions: a case study involving information extraction, in: Proceedings of Workshop on Learning from Imbalanced Datasets, ICML, 2003, pp. 1–7.

[21] T.G. Dietterich, Ensemble learning, The Handbook of Brain Theory and Neural Networks. 2 ,(2002), pp.110–125.

[22] S.H. Khan, M. Hayat, M. Bennamoun, F.A. Sohel, R. Togneri, Cost-sensitive learning of deep feature representations from imbalanced data, IEEE Trans Neural Netw Learn Syst. 29, (2017), pp.3573–3587.

[23] J.J. Chen, C.-A. Tsai, H. Moon, H. Ahn, J.J. Young, C.-H. Chen, Decision threshold adjustment in class prediction, SAR QSAR Environ Res. 17, (2006), pp.337–352.

[24] [24] S. Wang, W. Liu, J. Wu, L. Cao, Q. Meng, P.J. Kennedy, Training deep neural networks on imbalanced data sets, in: 2016 International Joint Conference on Neural Networks (IJCNN), IEEE, 2016, pp. 4368–4374.

[25] C. Huang, Y. Li, C.C. Loy, X. Tang, Learning deep representation for imbalanced classification, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 5375–5384.

[26] Y. Yan, M. Chen, M.-L. Shyu, S.-C. Chen, Deep learning for imbalanced multimedia data classification, in: 2015 IEEE International Symposium on Multimedia (ISM), IEEE, 2015, pp. 483–488.

[27] C.A.R. de Sousa, An overview on weight initialization methods for feedforward neural networks, in: 2016 International Joint Conference on Neural Networks (IJCNN), IEEE, 2016, pp. 52–59.

[28] R. Storn, K. Price, Differential evolution-a simple and efficient heuristic for global optimization over continuous spaces, Journal of Global Optimization. 11, p.341, (1997).

[29] S.V. Moravvej, S.J. Mousavirad, M.H. Moghadam, M. Saadatmand, An lstm-based plagiarism detection via attention mechanism and a population-based approach for pre-training parameters with imbalanced classes, in: Neural Information Processing: 28th International Conference, ICONIP 2021, Sanur, Bali, Indonesia, December 8–12, 2021, Proceedings, Part III 28, Springer, 2021, pp. 690–701.

[30] S.J. Mousavirad, G. Schaefer, I. Korovin, D. Oliva, RDE-OP: A region-based differential evolution algorithm incorporation opposition-based

learning for optimising the learning process of multi-layer neural networks, in: Applications of Evolutionary Computation: 24th International Conference, EvoApplications 2021, Held as Part of EvoStar 2021, Virtual Event, April 7–9, 2021, Proceedings 24, Springer, 2021, pp. 407–420.

[31] W. Deng, S. Shang, X. Cai, H. Zhao, Y. Song, J. Xu, An improved differential evolution algorithm and its application in optimization problem, Soft Comput. 25, (2021) ,pp.5277–5298.

[32] S.J. Mousavirad, S. Rahnamayan, Evolving feedforward neural networks using a quasi-opposition-based differential evolution for data classification, in: 2020 IEEE Symposium Series on Computational Intelligence (SSCI), IEEE, 2020, pp. 2320–2326.

[33] D. Bajer, Adaptive k-tournament mutation scheme for differential evolution, Appl Soft Comput. 85, p.105776, (2019).

[34] D. Sarkar, A. Narang, S. Rai, Fed-focal loss for imbalanced data classification in federated learning, ArXiv Preprint ArXiv, 2011.06283. (2020).

[35] S. Das, A. Konar, Automatic image pixel clustering with an improved differential evolution, Appl Soft Comput. 9, (2009), pp.226–236.

[36] I. Fister, D. Fister, S. Deb, U. Mlakar, J. Brest, I. Fister, Post hoc analysis of sport performance with differential evolution, Neural Comput Appl. 32, (2020), pp.10799–10808.

[37] S. Vijayarani, M.J. Ilamathi, M. Nithya, Preprocessing techniques for text mining-an overview, International Journal of Computer Science & Communication Networks. 5, (2015), pp.7–16.

[38] [38] M.F. Porter, An algorithm for suffix stripping, Program. 14, (1980), pp.130–137.

[39] S.J. Mousavirad, H. Ebrahimpour-Komleh, Human mental search: a new population-based metaheuristic optimization algorithm, Applied Intelligence. 47, (2017),pp. 850–887.

[40] S.K. Prabhakar, H. Rajaguru, D.-O. Won, Performance Analysis of Hybrid Deep Learning Models with Attention Mechanism Positioning and Focal Loss for Text Classification, Sci Program. 2021, (2021), pp.1–12.

[41] S.R. Bowman, G. Angeli, C. Potts, C.D. Manning, A large annotated corpus for learning natural language inference, ArXiv Preprint ArXiv:1508.05326. (2015).

[42] J. Chen, Q. Chen, X. Liu, H. Yang, D. Lu, B. Tang, The bq corpus: A large-scale domain-specific chinese corpus for sentence semantic equivalence identification, in: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, 2018, pp. 4946–4951.

[43] M. Marelli, S. Menini, M. Baroni, L. Bentivogli, R. Bernardi, R. Zamparelli, A SICK cure for the evaluation of compositional distributional semantic models., in: Lrec, Reykjavik, 2014: pp. 216–223.

[44] E.L. Pontes, S. Huet, A.C. Linhares, J.-M. Torres-Moreno, Predicting the semantic textual similarity with siamese CNN and LSTM, ArXiv Preprint ArXiv:1810.10641. (2018).

[45] K. V Price, Differential evolution, Handbook of Optimization: From Classical to Modern Approach. (2013), pp.187–214.

[46] X.-S. Yang, Firefly algorithm, stochastic test functions and design optimisation, International Journal of Bio-Inspired Computation. 2 ,(2010) pp.78–84.

[47] X.-S. Yang, A new metaheuristic bat-inspired algorithm, Nature Inspired Cooperative Strategies for Optimization (NICSO 2010). (2010) 65–74.

[48] [48] X.-S. Yang, S. Deb, Cuckoo search via Lévy flights, in: 2009 World Congress on Nature & Biologically Inspired Computing (NaBIC), Ieee, 2009, pp. 210–214.

[49] D. Karaboga, B. Basturk, A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm, Journal of Global Optimization. 39, (2007), pp.459–471.

[50] S. Mirjalili, S.M. Mirjalili, A. Lewis, Grey wolf optimizer, Advances in Engineering Software. 69, (2014), pp.46–61.

[51] S. Mirjalili, A. Lewis, The whale optimization algorithm, Advances in Engineering Software. 95, (2016), pp.51–67.

[52] [52] D. Bairathi, D. Gopalani, Salp swarm algorithm (SSA) for training feed-forward neural networks, in: Soft Computing for Problem Solving: SocProS 2017, Volume 1, Springer, 2019, pp. 521–534.

[53] G. Hackeling, Mastering Machine Learning with scikit-learn, Packt Publishing Ltd, 2017.

[54] S. Sonkar, A.E. Waters, R.G. Baraniuk, Attention word embedding, ArXiv Preprint ArXiv:2006.00988. (2020).

[55] S. Thavareesan, S. Mahesan, Sentiment lexicon expansion using Word2vec and fastText for sentiment prediction in Tamil texts, in: 2020 Moratuwa Engineering Research Conference (MERCon), IEEE, 2020, pp. 272–276.

[56] V. Pihur, S. Datta, S. Datta, Weighted rank aggregation of cluster validation measures: a monte carlo cross-entropy approach, Bioinformatics. 23, (2007), pp.1607–1615.

[57] S. Xie, Z. Tu, Holistically-nested edge detection, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 1395–1403.

[58] C.H. Sudre, W. Li, T. Vercauteren, S. Ourselin, M.J. Cardoso, Generalised Dice Overlap as a Deep Learning Loss Function for Highly Unbalanced Segmentations, DEEP LEARNING IN MEDICAL IMAGE ANALYSIS AND MULTIMODAL LEARNING FOR CLINICAL DECISION SUPPORT. 10553, (2017), pp.240–248.

[59] S. Sadegh Mohseni Salehi, D. Erdogmus, A. Gholipour, Tversky loss function for image segmentation using 3D fully convolutional deep networks, ArXiv E-Prints. (2017) arXiv-1706.