DOI: 10.15849/IJASCA.250730.03

Int. J. Advance Soft Compu. Appl, Vol. 17, No. 2, July 2025
Print ISSN: 2710-1274, Online ISSN: 2074-8523

Fractional Snow Ablation Optimization-based Random Multimodel Deep Learning for Web Document Summarization with Large Language Model

Ahmed Bahaaulddin A. Alwahhab[1], and Vian Sabeeh[2]

[1]Information Technologies Management Department, Technical College of Management, Middle Technical University, Baghdad, Iraq
ahmedbahaaulddin@mtu.edu.iq
[2] Information Technologies Management Department, Technical College of Management, Middle Technical University, Baghdad, Iraq
viantalal@mtu.edu.iq

Abstract

The substantial amount of information available on the web has made it challenging for users to search and process relevant content efficiently. Notably, web documents, such as product descriptions, blog posts, research papers, news stories, and articles, often contain lengthy and vast amounts of irrelevant or redundant information. Various methods have been introduced to summarize the web. However, they have faced challenges in mining key points from the documents. Hence, a new model termed Fractional Snow Ablation Optimization with Random Multimodel Deep Learning (FSAO-RMDL) is devised for web document summarization. Initially, the input web document undergoes tokenization, where the Bidirectional Encoder Representations from Transformers (BERT) is employed to tokenize the document. Following this, feature extraction is executed, and extractive summarization is performed using Random Multimodel Deep Learning (RMDL), trained by Fractional Snow Ablation Optimization (FSAO). The FSAO approach incorporates the Fractional Calculus (FC) and Snow Ablation optimization (SAO). Lastly, abstractive summarization is performed by exploiting the GPT-NeoX Large Language Model (LLM). Overall, the experimental outcomes of the FSAO_RMDL approach demonstrate that it obtained a maximum recall, F-measure, and precision of 93.766%, 92.750%, and 91.755%, respectively.

Keywords: *abstractive summarization, deep learning, extractive summarization, large language model, web document summarization.*

1 Introduction

In today’s digital era, cloud resources, including blogs, news, user-generated content, social media platforms, and webpages, have generated vast volumes of textual data, and their numbers are growing exponentially. Furthermore, extensive textual content exists in various books, novels, legal documents, scientific papers, biomedical documents, articles,

Received 26 March 2025; Accepted 6 June 2025

and other archives [1-2]. Text content is a predominant medium for conveying important information across multiple domains. Nevertheless, the volume of explanations and clarifications frequently conceals the actual perspective in the content, making it more challenging to acquire crucial details promptly [3]. Additionally, it takes users longer to locate the information they need. In line with this, the textual content of search results is overly complex to read and understand. Consequently, summarizing and condensing the text resources becomes imperative and significantly more critical [4].

Textual information summarization is challenging for humans due to the rapid growth of information in the vast data era, rendering manual summarization of most information impossible. As a result, this phenomenon significantly impacts long-form text documents since processing and summarizing them require exponentially more human labor and expertise. This, in turn, leads to a crucial bottleneck in advancing social and economic development, resulting in substantial vital information and knowledge going unobserved [5].

Text summarization is a vital area and a key research task in Natural Language Processing (NLP). It is defined as a term that involves producing summaries of various sizes based on user preferences and the original document's content [6]. It automatically alters a text or a group of texts on a similar subject into a summary with crucial, semantically valuable information for various downstream applications. This includes report generation, news digest creation, and search engine optimization. It also minimizes lengthy texts to a briefer abstract while maintaining their actual meaning [7-8]. A manual text summarization procedure that efficiently preserves the text's meaning is time-consuming for humans [9-11]. As a result, automatic text summarization is a more effective alternative to summarizing lengthy texts in seconds.

Text summarization has faced various difficulties in recent eras, prompting the development of solutions dating back to the 1950s [6]. The most common techniques in the field of automatic text summarization are Machine Learning (ML), Deep Learning (DL), graph sorting, and statistics. However, simple and intuitive, statistically based text summarization techniques often overlook understanding word-sense relationships in favor of considering only word-surface features. Moreover, although graph sorting models suit texts with loose structure, they do not account for contextual information [3]. Accordingly, the summarization issue is altered into a sentence-level supervised classification issue in ML. Based on assessing a training set of documents and their corresponding summaries, the system decides whether a sentence in a test document is part of a summary. Subsequently, ML-based summarization algorithms initiate by recognizing features from sentence length and preprocessed documents, such as proper nouns and sentence location. It then feeds those attributes into classifier techniques that employ ML to produce a single score . Building on this, text summarization models based on DL have become increasingly popular recently [12]. As such, DL-based frameworks significantly enhance the ease of engineering by minimizing reliance on manual feature extraction and linguistic pre-knowledge [7]. In particular, developing DL techniques has led to remarkable progress in NLP. Text summarization has significantly benefited from DL methods, as have other tasks, including text translation and sentiment analysis. For example, these modern summarization methods often employ a sequence-to-sequence model, typically an encoder-decoder model of neural networks trained on both the outcome and input [13]. Another DL technique is multimodel DL, which combines various neural network architectures to analyze data efficiently, leveraging the power of multiple neural architectures to oversee complex tasks. These models are particularly valuable for text classification and summarization, where combining different DL approaches yields better results than a single model. Meanwhile, Convolutional Neural Networks (CNNs) are used for feature extraction in text classification. In comparison, Recurrent Neural Networks (RNNs) or transformer models are employed for processing sequential data. In essence, this hybridization approach enhances the efficiency of capturing local and global features, particularly in non-linear text data. This improves accuracy in various tasks, such as sentiment analysis and topic categorization [14].

Bio-inspired techniques like swarm algorithms have already been applied to training DL models. This algorithm mimics the behavior of social organisms, such as flocks of birds or colonies of ants, to solve complex optimization problems. Correspondingly, swarm algorithms are utilized in DL training for significant effects [15]. They were used to improve profound learning results and search for optimal weights. At the same time, other techniques were employed efficiently to fine-tune hyperparameters, such as learning rates, batch size, and network architecture layers, all of which influence the model's performance. Although they lead to faster convergence and more robust models, manually fine-tuning them is time-consuming [16]. In other words, this flexibility is valuable in DL tasks requiring trade-offs between performance aspects [17].

Snow Ablation Optimizer (SAO) is one of the swarm algorithms inspired by nature that mimics the snow ablation phenomena, including snow melting related to environmental conditions. The crucial mechanism behind SAO involves gradually reducing snow layers, thereby simulating search space reduction in various optimization problems. In this algorithm, the objective function represents the environment's influence on the snow, which melts iteratively throughout the optimization process. The possible solutions are defined as snow layers, and the interaction process between these layers and a control system is based on temperature. Correspondingly, the SAO algorithm begins with an initial population of solutions and then updates them through the melting process, systematically eliminating lower-quality solutions. Melting is controlled by a cooling function where the better solutions are preserved while less promising solutions are ignored. This process is repeated until the algorithm converges to an optimal solution. The SAO has effectively solved complex optimization problems, particularly in continuous search spaces, by striking a balance between exploration and exploitation during the search process. Accordingly, SAO is a new melting-inspired mechanism that allows it to avoid premature convergence and overcome local optima [18]. Simultaneously, DL has exploited SAO in many tasks in previous studies. For instance, Manikandan K. (2024) has utilized SAO for hyperparameter tuning to train the Long Short-Term Memory (LSTM) auto-encoder model for sentiment classification. In noisy, ambiguous data, such as social media text, SAO helps improve the model's ability to learn from these data, leading to better accuracy values. The suggested model achieved 94.28% and 97% on sentiment 140 and the airline's dataset, respectively [19]. Wu *et al.* (2024) addressed issues in the Bidirectional Long Short-Term Memory (Bi-LSTM) model for photovoltaic power prediction tasks, such as carefully tuning various hyperparameters in Bi-LSTM models. Similarly, Bi-LSTM models can be prone to overfitting, especially with limited datasets. In addition, Bi-LSTM is less accurate in dealing with nonlinear data. Notably, Wu's model avoided overfitting by balancing exploration and exploitation through SAO behaviors during hyperparameter tuning of the Bi-LSTM model. This also enables effective oversight of non-linear power generation data fluctuations [20].

As a result, recently, users have frequently been overwhelmed by the vast amounts of information available on web pages due to the exponential development of online content. Thus, efficiently summarizing these web documents is crucial for providing users with clear, relevant, and concise information. In particular, the problem of automatically generating a concise and coherent synopsis of a web document while preserving its context and meaning is known as web document summarization. Thus, a robust web document

summarization system must produce concise, educational, and contextually relevant summaries from unstructured web content. Moreover, the prevailing summarization techniques failed to attain a high accuracy rate. Therefore, this paper proposes an efficient approach to Fractional Snow Ablation Optimization with Random Multimodal Deep Learning (FSAO-RMDL) for summarizing web documents. Initially, the web document is considered input and then subjected to the tokenization stage, where it is tokenized using BERT. Subsequently, feature extraction is performed, followed by extractive summarization using RMDL, which FSAO trains. Concurrently, FSAO is obtained by integrating two methods: Fractional Calculus (FC) and SAO. Finally, the abstractive summarization is accomplished using the GPT-NeoX Large Language Model (LLM).
The contribution of this research is as follows:

- ***Proposed FSAO_RMDL for web document summarization:*** A new method called FSAO_RMDL is presented for web document summarization in this research. As such, the RMDL method is employed for extractive summarization and is trained using FSAO, an optimizer derived by combining FC and SAO. Then, abstractive summarization is performed using the GPT-NeoX LLM technique.

The remaining work sections are arranged as follows: Section 2 exemplifies traditional methods' merits, demerits, and challenges. Section 3 elucidates FSAO_RMDL. Section 4 presents the outcomes of FSAO_RMDL, summarizing the findings from the web document, and Section 5 presents the conclusion.

2 Related Work

Jalil, Z. *et al.* [21] developed a graph-based summarizer, Grapharizer, to summarize extractive multi-document texts. This technique successfully addressed data redundancy, provided comprehensive coverage of entire topics, and resolved issues such as poor grammar, missing important information, and data repetition. Nevertheless, the Grapharizer did not consider DL algorithms to enhance document summarization efficiency or utilize multiple databases from different domains to improve usability. Meanwhile, Gurusamy, B.M. *et al.* [15] established an Optimized Auto Encoded Long Short-Term Memory Network (OAELSTM) for enhanced automatic text summarization. This model minimized overfitting problems during word embedding training and more effectively processed complicated text to produce attainable and content-rich summaries. This model effectively maintained the coherence and sentence-level structure of the generated summaries. Nevertheless, this method did not incorporate advanced neural network structures to maximize the accuracy and context of text summarization. At the same time, Bano S. *et al.* [22] devised Bidirectional Encoder Representations from Transformers (BERT) and Bidirectional Gated Recurrent Units (BiGRU) for extractive summarization. These reliable methods prevented overfitting problems and produced useful and relevant summaries, striking a balance between computational efficacy and model accuracy. However, these techniques did not investigate the usage of multi-language corpora and alternative assessment measures to provide helpful information about the effectiveness of cross-lingual and cross-metric measures. Additionally, this technique cannot enhance the accuracy and adaptability of automatic text summarization models. Similarly, Fan J. *et al.* [3] introduced the Multi-Features Maximal Marginal Relevance BERT (MFMMR-BertSum) model for extractive social media text summarization. This model minimized the time required to complete social media text summarization tasks, eradicated redundant information, and enhanced summary outcomes. Despite that, this

model did not account for the benefits of abstractive and extractive summarization techniques in improving performance.
Dai, W., and He, Q. [23] proposed a K-means clustering algorithm for automatic summarization. This model efficiently minimizes sentence semantic redundancy to enhance the quality of the extracted summary, achieving high accuracy and reducing sentence repetition in semantic summaries. Nonetheless, the K-means clustering algorithm did not consider the model's influence on sentence selection during text summarization. In addition, Moro, G. *et al.* [24] established an Efficient Memory-Enhanced Transformer-based Architecture (EMMA) for long-document summarization. The EMMA technique requires substantially less Graphics Processing Unit (GPU) memory when summarizing text documents. However, this method neither addressed the backpropagation problems encountered during text summarization nor considered memory writing and reading operations that utilize structured information extracted from the text to complete the task. Furthermore, Divya S. *et al.* [6] developed a hybrid summarization algorithm for extractive-abstractive summarization. This method rapidly converged in producing text summaries after learning the document's context from start to end. This allowed it to eliminate redundancies and extract informative sentences. Nevertheless, this method did not verify the summary's quality since there was no dataset with ground truth summaries. Additionally, Wazery, Y.M. *et al.* [12] established an Optimized Convolutional Neural Network and a Feed-Forward Neural Network (Opt-CNN-FFNN) for extractive summarization. This technique effectually captured a document's semantic and statistical information while requiring low computing time for text generation. Consistent with the others, this model could not contemplate an attention mechanism for the text summarization task. The difficulties encountered in the previous studies are outlined as follows:

- The OAELSTM model, introduced in [13], effectively distills complex textual content into succinct, cohesive summaries for web document summarization while maintaining the essential meaning of the original text. Nevertheless, this method did not capture the internal structure of words, making it challenging for the model to comprehend word variations and derivations.

- The BERT+BiGRU method established in [22] efficiently resolved the redundancy problems by combining various features to perform complex extractive summarization tasks. Nevertheless, this method did not investigate other features to manage highly complicated databases during text summarization.

- The Opt-CNN-FFNN method devised in [12] effectively avoided redundancy while capturing the primary concepts of the text. However, this method did not change other DL language models or transformers for CNN to perform the text summarization task.

- The MFMMR-BertSum model was devised in [3] for extractive social media text summarization. This technique required less computational resources and was relatively simple and faster. Despite that, this method did not enhance the accuracy and efficiency of extractive summarization.

- Abstractive summarization requires additional computational power and resources compared to extractive models, which can challenge real-time applications. This is particularly true with limited hardware or high-volume environments. Moreover, attaining high summary accuracy is highly challenging.

Therefore, the proposed model addresses the limitations of previous studies by developing an adaptable and learnable model that can be applied to various datasets. It also combined abstractive and extractive summarization with cutting-edge transformers (e.g., BERT), offering a richer semantic perspective. Additionally, SAO enhances the model by reducing overfitting and allowing it to focus on more relevant features. The SAO-optimized hybrid model generalizes more effectively and captures a more profound semantic context, enhancing overall summarization quality across various datasets.

3 Methodology

The document is summarized by extracting relevant information from a specified document while preserving key details and condensing the content. However, the prevailing techniques utilized for summarization often faced challenges in automatically generating summaries of web documents with high relevance and accuracy. Therefore, a new model referred to as FSAO_RMDL is introduced to summarize the web document. Initially, the input web document is extracted from the provided dataset, which is then subjected to the tokenization stage, where the BERT [25] tokenizer is used to tokenize the input data. In addition, various features, including sentence-to-sentence similarity [26], Bag-of-Words (BOW) [27], word2vec [28], Term Frequency-Inverse Document Frequency (TF-IDF) [29], and sentence length [30], are extracted. Following this, extractive summarization is performed based on the mined features using RMDL [31], which is tuned by FSAO [32], an optimizer formulated by combining FC [32] and SAO [33]. At the same time, the abstractive summarization uses the GPT-NeoX LLM approach, considering the extractive summary formed by the RMDL. Accordingly, Fig. 1 exhibits the schematic view of FSAO_RMDL for web document summarization. It also addresses BERT tokenization on the input web document. The feature extraction processes extract BOW, TF-IDF, word2vec, sentence length, and sentence-to-sentence similarity. Subsequently, the extracted features are fed into the RMDL model for extractive summarization, which the proposed FSAO optimizes. Notably, the proposed FSAO was initially based on an SAO optimizer enhanced by FC to reach a convergence level efficiently. Correspondingly, extractive summarization produces key idioms, after which abstractive summarization uses the keywords to generate a comprehensive, informative summary paragraph for the readers.

3.1 Data Acquisition

Assume the input data for web document summarization is obtained from a web document database. U with t is the amount of web data, which is modeled as:

$$U = \{U_1, U_2, \ldots, U_z, \ldots, U_t\}. \tag{1}$$

Here, the overall quantity of documents is postulated as t, and U_z exemplifies the z^{th} web document.

3.2 BERT Tokenization

Tokenization was applied to the input web document. U_z is subjected to a tokenization process using BERT tokenization [25]. BERT is employed to change sentences or paragraphs into tokens or individual words. Correspondingly, a single sentence can be compressed into a single token sequence to represent the input, allowing

BERT to manage multiple downstream tasks. Moreover, the specified sentence is regarded as a random span of contiguous text instead of an actual linguistic sentence. As such, the input token is designated as a sequence consisting of one or two sentences concatenated together. Notably, the primary token in all the sequences is a unique classification token, which is typified as ($[CLS]$). In the classification task, this token corresponds to the last hidden state, which is employed as the aggregate sequence representation. The sentence is differentiated based on two methods. First, the given sentence is separated into unique tokens ($[SEP]$). In the second process, the learning embedding is added to each token to demonstrate the sentence with which it is associated. $\beta_o \in \mathbb{R}_x$ exemplifies the last hidden vector for o^{th} the input token, $h \in \mathbb{R}^x$designates the last hidden vector of a unique token, and h indicates the input embedding. Furthermore, the corresponding position token, segment, and embedding are added to generate the input representation. Additionally, the tokenized web document is represented as ϖ_x.

3.3 Feature Extraction

In web document summarization, feature extraction is used to identify attributes within the document, which can be leveraged to create a concise summary. Moreover, this process is also beneficial in determining the most appropriate content and ensuring that the summary characterizes essential information. Concurrently, features such as BOW [27], TF-IDF [29], sentence-to-sentence similarity [26], word2vec [28], and sentence length [30] are extracted by considering ϖ_x as an input. This is in addition to these features, which are described below.

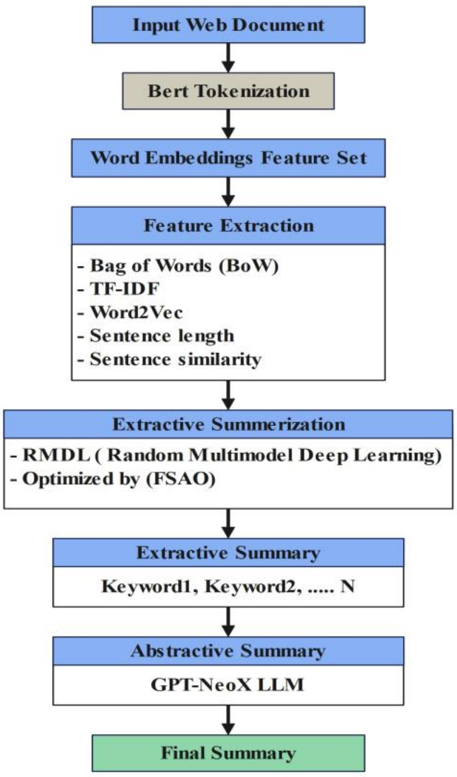


Fig. 1. Schematic view of FSAO_RMDL for web document summarization

3.3.1 Bag of Words

The BOW model [27] encompasses word tokens transformed into a binary representation of bits, allowing the machine to maintain or learn its vocabulary. Here, the outcome of this approach is considered a binary feature vector, which is represented as C_1.

3.3.2 TF-IDF

TF-IDF [29] is exploited to characterize the text by determining the frequency range of each term observed in the document. TF-IDF identifies the word's frequency in a specific document correlated to the inverse proportionality of that word over the whole document corpus. It helps determine the relevant work in the given document. Furthermore, the word frequency score is generated by TF-IDF, and it is expressed as:

$$C_2 = TF_{M,y} \times log\left(\frac{\hbar}{Z_M}\right), \tag{2}$$

wherein $\hbar$ indicates the document count at M^{th} document. The word in the document is denoted as $TF_{M,y}$, a quantity of documents is represented as Z_M, and C_2 Specifies the TF-IDF feature.

3.3.3 Word2Vec

The word2vec [28] signifies the semantic relationship between words within a document. This feature provided maximum performance for the semantic task in establishing the word's association with other similar words. Moreover, cosine similarity is utilized for identifying the similarities between the words, which are written as:

$$Similarity = cos\,\phi = \frac{\bar{b}\cdot\bar{j}}{||\bar{b}||||\bar{j}||}, \tag{3}$$

where b^{th} and j^{th} long vector is denoted as $||b||$and $||j||$, vector dot product from $b$and j is represented as $\bar{b}\cdot\bar{j}$, and C_3 designates the word2vec feature.

3.3.4 Sentence Length

Sentence length [30] is the central element in the summary, which must be of adequate length. Here, the sentence length is assessed by dividing the length by the longest sentence's length, where the number of words in the sentence is designated as length. The sentence length is modeled as follows:

$$C_4 = \frac{L_w}{L_a}, \tag{4}$$

where the length of the long sentence is denoted as L_a, the length of the sentence is represented as L_w, and C_4 signifies the sentence length feature.

3.3.5 Sentence-to-Sentence Similarity

This feature [26] measures the similarity among the sentences based on their structure, content, and meaning. Notably, this similarity is usually measured using several models, namely cosine similarity, which intends to evaluate the close relation. This includes conveying the same information or idea between two sentences, which is articulated as:

$$C_5 = \frac{F_d}{Max(F_d)}, \tag{5}$$

where this feature is denoted as C_5, F_d denotes the number of sentence similarities, and $Max(F_d)$ represents the maximum number of sentence similarities.

Here, the features that are explained above are unified to generate a feature vector, C, and it is designated as:

$$C = \{C_1, C_2, C_3, C_4, C_5\}. \tag{6}$$

3.4 Extractive summarization

Extractive summarization extracts a document by directly selecting and mining the original text's most significant sections, phrases, or sentences. Accordingly, the original sentence in the summary is retained by selecting the most applicable ones. Furthermore, this process creates a summarized document, extracting the most relevant and significant information by eliminating less relevant or redundant content. It is primarily beneficial to deal with large corpora or long documents, where rapid selection of critical points is necessary. In addition, the extractive summarization is performed by employing RMDL [31].In particular, the proposed approach combines three DL models instead of relying on one type. First, CNN is preferred in detecting patterns since it focuses on essential phrases within the text. Second, RNN understands sequences or sentence structures. Third, deep neural networks (DNNs) process complex features in many layers. Furthermore, RMDL targeted extractive text summarization by analyzing text features such as word importance and sentence semantic similarity. The multimodel then votes on the best sentence to include. Following this, the FSAO enhances the efficiency of RMDL as a coach who trains athletes to reach their best performance. As a result, FSAO fine-tunes RMDL for better accuracy.
SAO is inspired by the natural snow melting phenomenon, which occurs when snow layers gradually melt under environmental conditions. In conjunction with RMDL fine-tuning, each candidate group of hyperparameters is represented as a layer of snow or solution. As such, SAO works iteratively by melting the less promising hyperparameter adjustments and focusing on preserving and exploring better solutions. It also avoids trapping in poor solutions by a selective melting strategy to ensure better solution space exploration and determine optimal adjustments for the multimodel DL system. At the same time, FC enhances the fine-tuning process by preparing a mathematical framework that preserves the history of the search path, offering a more precise update.
Contrary to traditional optimization paradigms that depend on simple step-by-step changes, FC conjuncts memory of past movements in the solution space. This results in mode-balanced decisions that avoid arbitrary jumps or slow convergence. Hence, by combining these two ways, FSAO successfully balances between exploration and exploitation. This combination yielded faster convergence to optimal, stable, and robust hyperparameters, leading to a more accurate and reliable RMDL model. Correspondingly, the fine-tuned RMDL refines key sentences from documents by exploiting the strength of various DL models like CNNs, RNNs, and DNNs. The teamwork between SAO's naturally inspired paradigm and FC's mathematical strategy produces a smooth acceleration. It strengthens the multimodel system's training, enhancing its ability to generate high-quality, pithy summaries. Briefly, SAO mimics snow melting to eliminate inferior solutions progressively. Simultaneously, FC controls this melting with a memory-based mechanism that renders the optimization more efficient and stable. This is in tandem with enabling the fine-tuning of RMDL for enhanced performance in web document summarization. The description of FSAO_RMDL is provided below.

3.4.1 Architecture of RMDL

RMDL [31] is a robust approach that leverages the strengths of multiple techniques to enhance the performance of DL models. This method incorporates various DL methods tuned with randomness into the training process, allowing these combined approaches to make robust decisions and predictions. As such, the feature vector C is fed as an input to RMDL, where DL techniques, including CNN, DNN, and RNN, are combined to obtain this model. Note that integrating multiple techniques minimizes overfitting issues, thereby enhancing the model's overall capability. In addition, RNN is utilized for text classification, while CNN is employed for image or document classification. At the same time, DNN is utilized for classification and includes many hidden layers. Two RNNs RMDL uses: LSTM and Gated Recurrent Unit (GRU). Furthermore, the number of layers of all these DL multi-approaches is created, and the RMDL's output is represented as:

$$N(g_{q1}, g_{q2}, \dots \dots, g_{q3}) \; = \left[\frac{1}{2} + \frac{(\sum_{H=1}^{s} V_{nH})}{s} - \frac{1}{2}\right], \tag{7}$$

where V_{nH} symbolizes the prediction result for the H^{th} model at n^{th}. The data point, the number of random techniques, is denoted as s. Meanwhile, the majority voting, which is used for assessing the final result g_n, is represented as:

$$g_n = [g_n \dots \dots g_H \; \dots \dots g_{ns}]^T, \tag{8}$$

where g_{nH} postulates the n^{th} data point of document label, and is formulated as:

$$Y_p = arg\,\max_{\vartheta}\,[Soft\,max(C)], \tag{9}$$

where ϑ exemplifies the number of classes. The output attained in RMDL is stipulated as Y_p, and C stipulates the mined features fed to the RMDL model. Fig. 2 displays the RMDL architecture. Notably, the RMDL in Fig. 2 is a hybrid multimodel system that combines three DL models: CNN, DNN, and RNN. Each extracts a specific set of features to minimize overfitting by incorporating a voting classification mechanism.

3.4.2 Proposed Fractional Snow Ablation Optimization for Tuning Random Multimodel Deep Learning

FSAO, which is engineered by combining FC [32] and SAO [33], is utilized to fine-tune the RMDL network, thereby enhancing convergence efficiency and overall performance. Specifically, SAO [33] is a novel nature-inspired metaheuristic approach for engineering design and numerical optimization. Accordingly, the snow's melting and sublimation behavior is emulated primarily to achieve a trade-off between exploration and exploitation, discouraging premature convergence and expanding the solution space. Moreover, SAO exhibits high scalability, even as the dimensionality of the optimization issue increases. The simulation results demonstrate that the SAO model was more robust and could yield high performance. In line with this, FC [32] optimizes the algorithm's computational performance. Furthermore, FC is enhanced by incorporating fine-tuning and flexibility into the optimization models to enhance the search process. Overall, the integration of FC and SAO is more effective in improving performance, and the mathematical process of FSAO is presented below.

(i) Initialization: The iteration process in SAO begins with an arbitrarily formed swarm, where the swarm is represented in a matrix form with dim columns and R Rows, expressed as:

$$A = f + \varphi \times (q - f), \tag{10}$$

$$A = \begin{bmatrix} l_{1,1} & l_{1,2} & \cdots & l_{1,dim-1} & l_{1,dim} \\ l_{2,1} & l_{2,2} & \cdots & l_{2,dim-1} & l_{1,dim} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ l_{R-1,1} & l_{R-1,2} & \cdots & l_{R-1,dim-1} & l_{R-1,dim} \\ l_{R,1} & l_{R,2} & \cdots & l_{R,dim-1} & l_{R,dim} \end{bmatrix}, \tag{11}$$

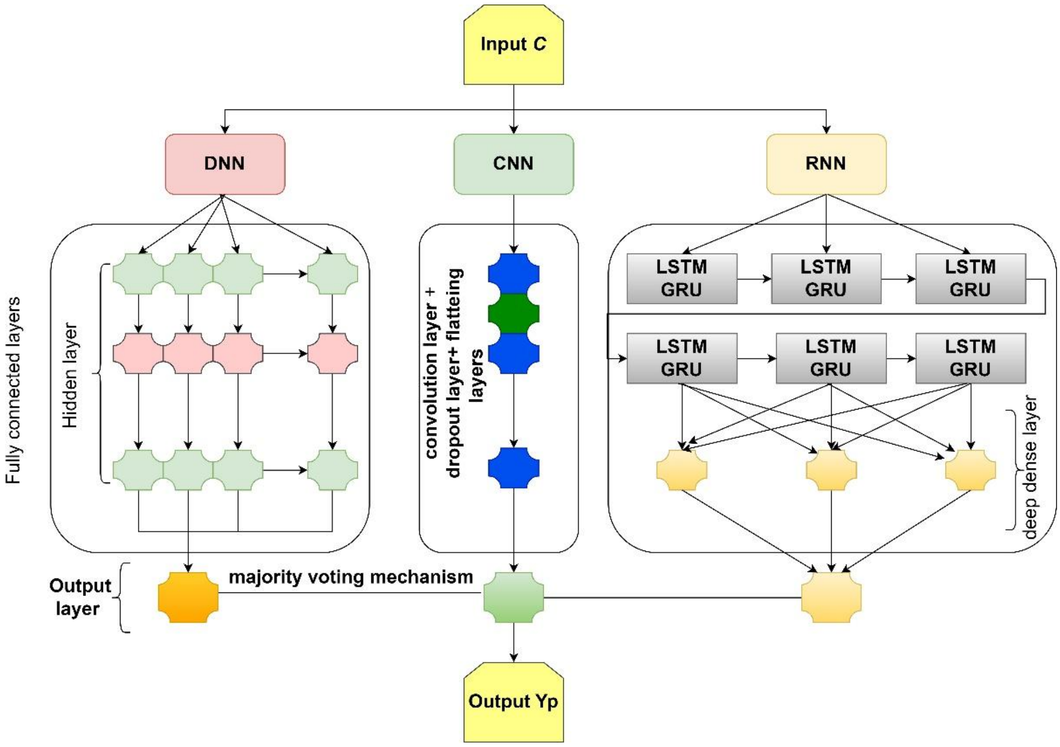


Fig. 2. RMDL architecture

where the swarm is denoted as A, l specifies the search agent's position, and the solution space's dimension is implied as dim, random number among [0,1] is characterized as φ, size of the swarm is denoted as R, solution space's upper, and lower bounds are represented as q and f.

(ii) Fitness function: Mean Square Error (MSE) is employed as a fitness function to evaluate the prediction quality during the tuning process. Moreover, the MSE is designated as:

$$MSE = \frac{1}{m} \sum_{p=1}^{m} (Y_p - Y_p^*)^2, \tag{12}$$

wherein, Y_p^* specifies the predicted value of RMDL, Y_p symbolizes the actual values, and m represents the number of samples.

(iii) Exploration stage: A high-decentralized feature is presented by the search agent due to the irregular movement during the process when the liquid water or snow converted from snow transforms into water vapor. Subsequently, a Brownian motion is employed to model this scenario. Brownian motion, as a stochastic process, is widely applied to emulate the changing behavior of stock prices, the irregular and endless movement of particles, and the foraging behavior of animals, among other phenomena. Additionally, the step size is obtained using the probability density function for standard Brownian motion based on a normal distribution with variance one and mean 0, which is modeled as:

$$G_{BM}(v;0,1) = \frac{1}{\sqrt{2\pi}} \times exp\left(-\frac{v^2}{2}\right). \tag{13}$$

Furthermore, in the search space, some potential regions are explored, which is enabled by Brownian motion through the use of uniform and dynamic step lengths. Correspondingly, the steam spreading out in the search space is reflected well. Moreover, during the exploration process, the position updated is formulated as follows:

$$A(u+1) = Elite(u) + BM(u) \otimes \Big(\varphi_1 * (S(u) - A(u)) + (1-\varphi_1) * (\bar{A}(u) - A(u))\Big), \tag{14}$$

$$A(u+1) = Elite(u) + BM(u) \otimes (\varphi_1 * S(u) - \varphi_1 * A(u) + (1-\varphi_1) * \bar{A}(u) - (1-\varphi_1) * A(u)), \tag{15}$$

$$A(u+1) = Elite(u) + BM(u) \otimes (\varphi_1 * S(u) - A(u)(\varphi_1 + (1-\varphi_1)) + (1-\varphi_1) * \bar{A}(u)), \tag{16}$$

$$A(u+1) = Elite(u) + BM(u) \otimes (\varphi_1 * S(u) - A(u)(\varphi_1 + 1 - \varphi_1)) + (1-\varphi_1) * \bar{A}(u)), \tag{17}$$

$$A(u+1) = Elite(u) + BM(u) \otimes (\varphi_1 * S(u) - A(u) + (1-\varphi_1) * \bar{A}(u)). \tag{18}$$

Moreover, by subtracting $A(u)$ on both sides to apply FC, we obtain:

$$A(u+1) - A(u) = Elite(u) + BM(u) \otimes (\varphi_1 * S(u) - A(u) + (1-\varphi_1) * \bar{A}(u)) - A(u). \tag{19}$$

FC [32] is integrated with SAO to enhance convergence, making the process faster and more robust against local optima. Thus, by applying FC [32]:

$$D^\eta(A(u+1) - A(u)) = Elite(u) + BM(u) \otimes (\varphi_1 * S(u) - A(u) + (1-\varphi_1) * \bar{A}(u)) - A(u), \tag{20}$$

$$A(u+1) - \eta * A(u) - \frac{1}{2}\eta * A(u-1) - \frac{1}{6}(1-\eta) * A(u-2) - \frac{1}{24}\eta * (1-\eta)(2-\eta)A(u-3) = Elite(u) + BM(u) \otimes (\varphi_1 * S(u) - A(u) + (1-\varphi_1) * \bar{A}(u)) - A(u), \tag{21}$$

$$A(u+1) = Elite(u) + BM(u) \otimes (\varphi_1 * S(u) - A(u) + (1-\varphi_1) * \bar{A}(u)) - A(u) + \eta * A(u) + \frac{1}{2}\eta * A(u-1) + \frac{1}{6}(1-\eta) * A(u-2) + \frac{1}{24}\eta * (1-\eta)(2-\eta)A(u-3), \tag{22}$$

$$A(u+1) = Elite(u) + BM(u) \otimes (\varphi\_1 * S(u) - A(u) + (1-\varphi\_1) * \bar{A}(u)) - A(u)(1-\eta) + \frac{1}{2}\eta * A(u-1) + \frac{1}{6}(1-\eta) * A(u-2) + \frac{1}{24}\eta * (1-\eta)(2-\eta)A(u-3) \tag{23}$$

Here, the random number between [0,1] is represented as φ_1, a vector with a random number regarding Gaussian distribution symbolizing the Brownian motion is indicated as $BM_\omega(u)$. Meanwhile, ω^{th} an individual at the u^{th} iteration is denoted as $A_\omega(u)$, current finest solution is signified as $S(u)$, entry-wise multiplication is expressed as $\otimes$. At the same time, $\bar{A}_u$ embodies the whole swarm's centroid position, while an arbitrary individual chosen from a group of several elites in the swarm is represented as $Elite(u)$. Furthermore, the corresponding mathematical equations are presented below.

$$\bar{A}(u) = \frac{1}{R}\sum_{\omega=1}^{R} A_\omega(u), \tag{24}$$

$$Elite(u) \in \left[S(u), Athi_{r_{sec}}[u]\right], \tag{25}$$

wherein an individual's centroid position with fitness value ranked in the top 50%, termed as leader, is symbolized as $A_r(u)$. In comparison, the third-best and second-best individuals in the present population are characterized as $A_{thi}(u)$ and A_{sec}. In addition, the $A_r(u)$is evaluated based on the equation below:

$$A_r(u) = \frac{1}{R_1}\sum_{\omega=1}^{R_1} A_\omega(u). \tag{26}$$

Here, ω^{th} the best leader is signified as $A_\omega(u)$. The number of leaders is stipulated as R_1, where $R_1$is half the size of the entire swarm. Following this, at every iteration, $Elite(u)$ is selected arbitrarily from a set that involves the leader's centroid position, third-best individual, and second-best individual and presents an optimum solution. To control the movement to the centroid location and the movement toward the present optimum individual, the parameter $\varphi_1$is used. The combination of the $\varphi_1 \times (S(u) - A_\omega(u))$ and $(1-\varphi_1) \times (\bar{A}(u) - A_\omega(u))$ cross terms are also exploited to reflect the incorporation among individuals.

(iv) Exploitation stage: The exploitative characteristics of this algorithm are formulated in this stage. During the process of snow conversion into liquid water through melting behavior, the search agents are stimulated to exploit the high-quality solutions around the present optimum solution rather than increasing with high-decentralized features in the solution space. Accordingly, the degree-day approach is used for reflecting the snow melting process, and it is formulated as follows:

$$J = \gamma \times (\ell - \ell_1), \tag{27}$$

wherein a base temperature, which is typically set to 0, is implied as ℓ_1, and the average daily temperature is implied as ℓ. Meanwhile, the snowmelt rate is denoted as J, the key parameter used for emulating the melting behavior, and is expressed as:

$$J = \gamma \times \ell. \tag{28}$$

where the degree-day factor, which varies from 0.35 to 0.6, is represented as γ. Moreover, the expression to update the γ^{th} value in every iteration is articulated as:

$$\gamma = 0.35 + 0.25 \times \frac{e^{\frac{u}{u_{max}}}}{e-1}, \tag{29}$$

wherein the termination criterion is expressed as u_{max}. Following this, the snowmelt rate is determined using the expression presented below:

$$J = \left[0.35 + 0.25 \times \frac{e^{\frac{u}{u_{max}}}}{e-1} []^{\frac{-u}{u_{max}}}\right]. \tag{30}$$

Additionally, the location updating expression in the exploitation stage is designated as:

$$A_\omega(u+1) = J \times S(u) + BM_\omega(u) \otimes \left[\varphi_2 \times \left(S(u) - A_\omega(u)\right) + (1-\varphi_2) \times \left(\bar{A}(u) - A_\omega(u)\right)\right], \tag{31}$$

where the snowmelt rate is characterized as J, a random number ranging from $[-1,1]$ is represented as φ_2, which is used to facilitate communication among individuals. Moreover, the individuals are expected to employ capable regions based on the swarm's centroid position and present the finest search agent's knowledge using cross-terms $\varphi_2 \times (S(u) - A_\omega(u))$ and $(1-\varphi_2) \times (\bar{A}(u) - A_\omega(u))$.

(vi) Re-evaluation: After every iteration, the present solution is re-evaluated to determine the enhancements and is assessed using the expression (12).

(vii) Termination: The process is continued repetitively until the finest solution is accomplished. In essence, the overall performance of RMDL is effectively improved by employing FSAO. This enhances the robustness and effectiveness of determining optimal solutions with faster convergence. The extractive summarization is postulated as Y_p. The pseudocode of FSAO is portrayed as follows:

```
Start FSAO
Initialization: Swarm A_ω(ω = 1,2,...,R), u = 0, u_max, R/2
Fitness evaluation
Record the present optimum individual S(u)
While (u < u_max do
    Calculate the snowmelt rate J
     for every individual do
             Update every individual's location using equations (23) and (31)
     end for
     Fitness valuation
      Update S(u)
     u = u + 1
End while
Return S(u)
```

3.5. LLM Model

The outcome Y_p generated by RMDL, the input is used to obtain an abstractive summarized output and is then subjected to the LLM. In NLP, LLMs like GPT-NeoX are regarded as an essential phase that enables document summarization. At this point, abstractive summarization is enabled by a transformer-based model that generates relevant text in response to prompts and produces contextually accurate summaries. The model can also condense and rephrase the document, making the process highly efficient in generating summaries of various text type.

Due to its powerful language generation capabilities, this approach utilizes a large-scale model to tune multiple datasets. Furthermore, it is also employed to oversee various tasks, such as text generation, summarization, translation, and question generation, among others. In particular, this technique exhibits long-range dependency in generating summaries from documents and can effectively manage large datasets. Thus, the GPT-NeoX removes irrelevant information from web documents, making them applicable to large-scale practices. Moreover, the outcome attained from GPT-NeoX LLM is formulated as U_h, which is the abstractive summary.

4 Results, Analysis and Discussions

This section presents an assessment of the FSAO_RMDL for web document summarization, providing an exhaustive description of the dataset and performance measures.

4.1 Experimental setup

The implementation of the FSAO_RMDL is performed utilizing the Python tool with the Data Understanding Conference (DUC) 2002 [34] and the DUC 2004 database [35].

4.2 Dataset description

The text document used for summarization is taken from the DUC, namely the DUC 2002 [34] and DUC 2004 [35] datasets.

a) DUC 2002 dataset: This database comprises approximately 600 documents [34], classified into 60 collections. Each collection was differentiated using various criteria, including event sets and biographical sets, to name a few, as well as single-document abstracts, comprised documents, and multi-document abstracts or extracts. In addition, 200- and 400-word summaries are available.

b) DUC 2004 dataset: This database contains 500 new articles and 50 sets of Text Retrieval Conference (TREC) documents [35]. Each collection comprises approximately ten documents, and four handcrafted summaries accompany each article.

4.3 Evaluation metrics

The evaluation measures, including F-measure, precision, and recall, used to assess the efficiency of the FSAO_RMDL approach are explained below.

i) Recall: The recall is assessed by comparing the relevant information in the generated summary to the relevant sentence in the original document. Recall is designated as:

$$D = \frac{T}{\lambda}. \tag{32}$$

Here, D denotes recall, T signifies the number of relevant sentences in the summary, and λ exemplifies the number of appropriate sentences in the original document.

ii) Precision: This measure is exploited for measuring how much content in the produced summary is suitable to the original content and is signified as:

$$F = \frac{T}{Q}, \tag{33}$$

wherein F indicates the precision and Q specifies the overall number of sentences in the summary.

iii) F-measure: F-measure is attained by combining recall and precision into a single score, which provides a balanced evaluation. The expression of this measure is formulated as:

$$\beta = 2\frac{D*F}{D+F}, \tag{34}$$

where β expresses the F-measure.

4.4. Experimental results

Fig. 3 depicts the experimental outcomes of FSAO_RMDL for abstractive summarization. Notably, the input document obtained from the DUC 2002 dataset and the generated abstractive and extractive summarizations are portrayed in Fig. 3(a). Meanwhile, the input document obtained from the DUC 2004 database and the generated abstractive and extractive summarizations are portrayed in Fig. 3(b).

4.5. Algorithmic Methods

Multiple algorithms, such as the Tunicate Swarm Algorithm+RMDL (TSA+RMDL) [31] [36], Border Collie Optimization+RMDL (BCO+RMDL) [31] [37], Gannet Optimization Algorithm+RMDL (GOA+RMDL) [31] [38], and SAO+RMDL [31] [33], are compared with FSAO+RMDL to evaluate the effectiveness of the model. Correspondingly, an algorithmic assessment of the FSAO+RMDL is performed by varying the swarm sizes between 5 and 20.

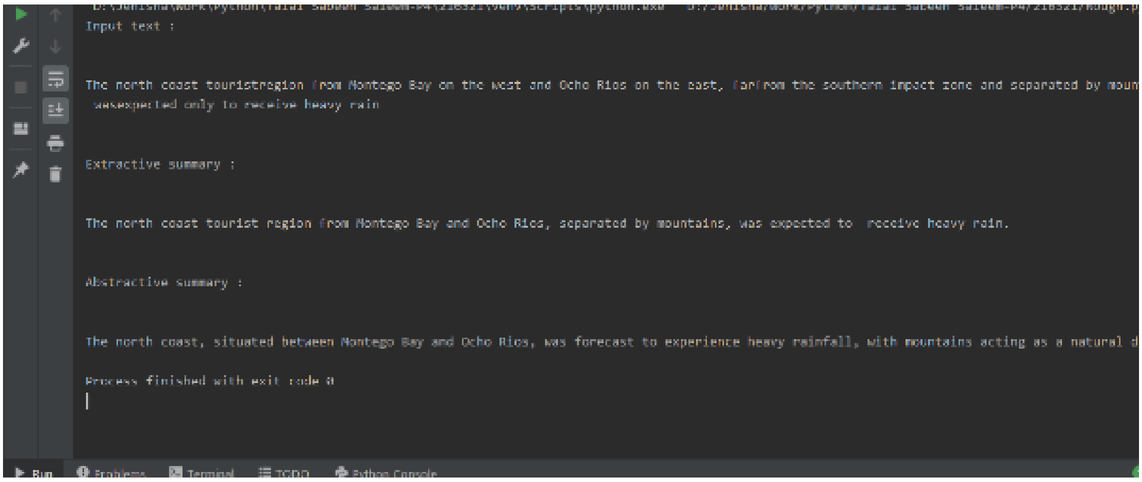
4.6. Algorithmic Assessment

This article devises a hybrid FSAO algorithmic scheme to update the trainability of the RMDL employed for abstractive summarization. Moreover, the effectiveness of this approach is evaluated using documents from the DUC 2004 and DUC 2002 databases based on changes in swarm sizes.

4.6.1 DUC 2002 database

The estimation of the FSAO+RMDL for the DUC 2002 dataset regarding swarm size is illustrated in Fig. 4. Fig. 4(a) presents the examination of the FSAO+RMDL with the F-measure. At a swarm size of ten, the F-measure quantified by existing models, such as TSA+RMDL, BCO+RMDL, GOA+RMDL, SAO+RMDL, and FSAO+RMDL, is 82.375%, 83.865%, 85.211%, 86.319%, and 88.217%, respectively. This indicates that the

FSAO+RMDL achieved an F-measure improvement of 6.62%, 4.93%, 3.41%, and 2.15% compared to the traditional models.
The evaluation of the FSAO+RMDL regarding recall is illustrated in Fig. 4(b). Remarkably, the FSAO+RMDL achieved a recall of 90.876%, which is higher by 7.70%, 5.50%, 4.41%, and 3.30% compared to the recalls recorded by TSA+RMDL (83.877%), BCO+RMDL (85.876%), GOA+RMDL (86.868%), and SAO+RMDL (87.877%), with a swarm size of 15. The precision-based examination of the FSAO+RMDL is indicated in Fig. 4(c). Conversely, by considering a swarm size of five, the precision achieved by TSA+RMDL is 80.870%, BCO+RMDL is 81.766%, GOA+RMDL is 83.877%, SAO+RMDL is 84.877%, and FSAO+RMDL is 86.876%. This indicates that the FSAO+RMDL achieved enhanced performance of 6.91%, 5.88%, 3.45%, and 2.30% compared to the existing models.



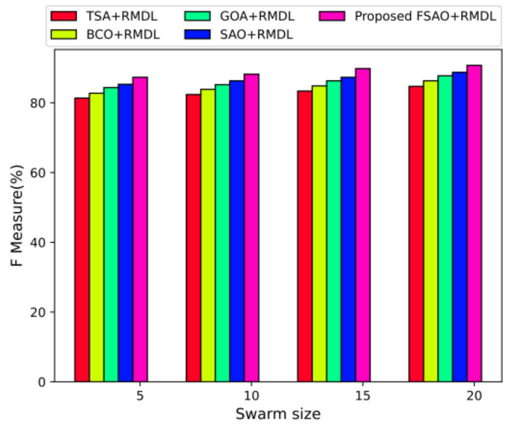
(a)

(b)

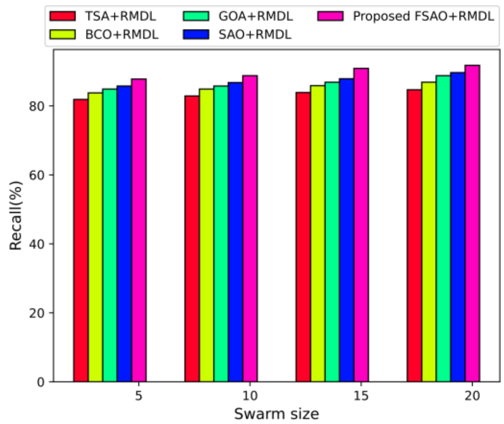
Fig. 3. Experimental results of FSAO_RMDL for (a) DUC 2002 and (b) DUC 2004 datasets

4.6.2 Assessment using the DUC 2004 dataset

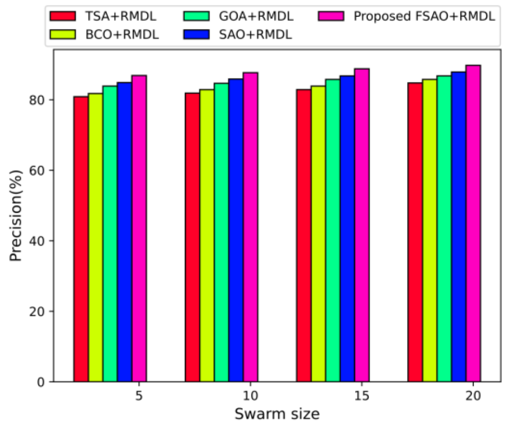
Fig. 5 specifies the evaluation of the FSAO+RMDL for the DUC 2004 dataset based on swarm size. In Fig. 5(a), the examination of the FSAO+RMDL for the F-measure is displayed. When the swarm size is assumed as 20, the FSAO+RMDL gained a higher F-measure of 92.750%, which is better than the F-measure values calculated by TSA+RMDL, BCO+RMDL, GOA+RMDL, and SAO+RMDL at 86.318%, 87.876%, 89.318%, and 90.809% by 6.93%, 5.25%, 3.70%, and 2.09%.



(a)



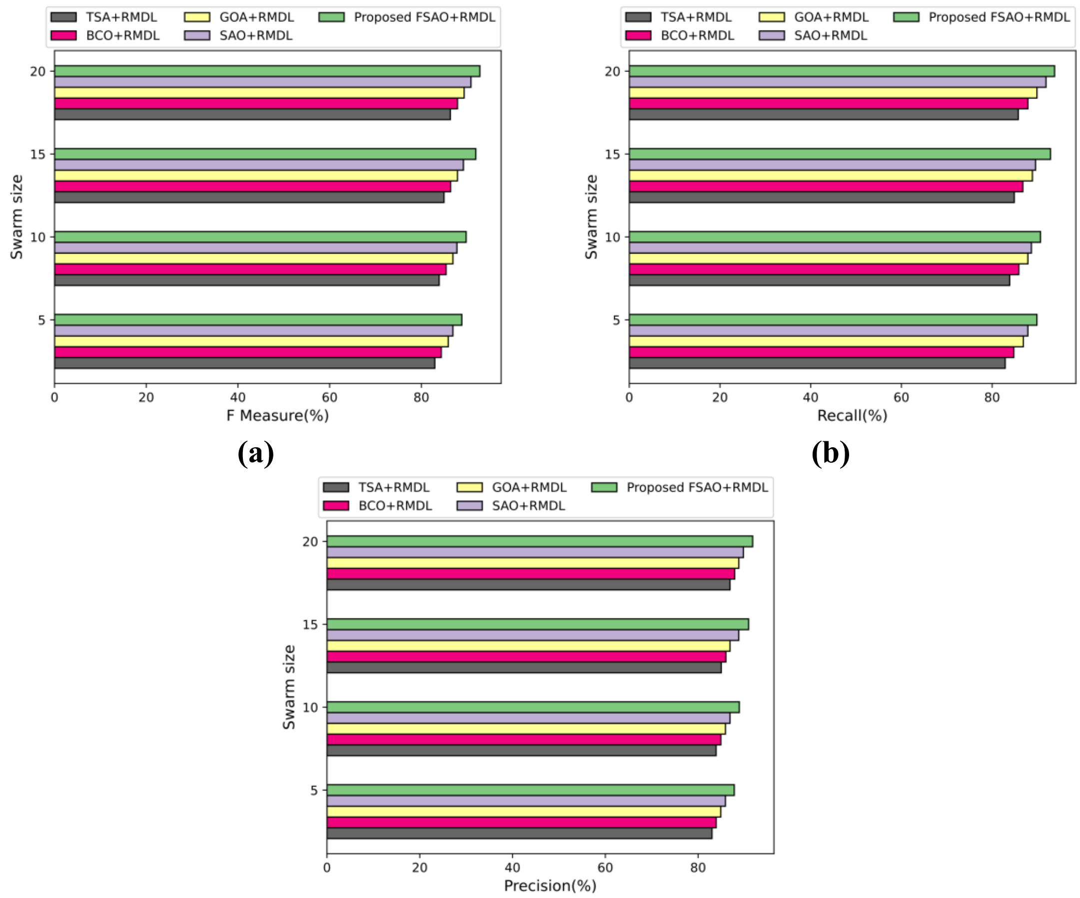
(b)



(c)

Fig. 4. Algorithmic validation of the FSAO+RMDL employing the DUC 2002 database with a) F-measure, b) recall, and c) precision

In comparison, Fig. 5(b) specifies the analysis of FSAO+RMDL regarding recall. The recall recorded by the FSAO+RMDL is 90.654%, while the values computed by the traditional models, including TSA+RMDL, BCO+RMDL, GOA+RMDL, and SAO+RMDL, are 83.877%, 85.877%, 87.877%, and 88.654%, respectively, at a swarm size of 10. Therefore, the FSAO+RMDL is demonstrated to generate a superior recall value of 7.48%, 5.27%, 3.06%, and 2.21% more than the conventional schemes. At the same time, the valuation of the FSAO+RMDL regarding precision is indicated in Fig. 5(c). When the swarm size of 15 is considered, the approaches, including TSA+RMDL, BCO+RMDL, GOA+RMDL, SAO+RMDL, and FSAO+RMDL, quantified a precision of 84.987%, 85.987%, 86.877%, 88.757%, and 90.876%. This depicts that the FSAO+RMDL gained superior precision by 6.48%, 5.38%, 4.40%, and 2.33%.



(c)

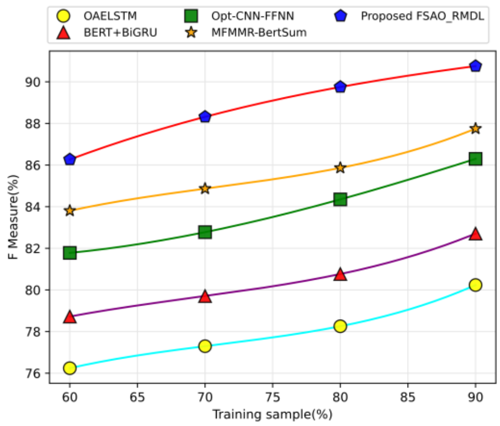
Fig. 5. Algorithmic investigation of the FSAO_RMDL employing the DUC 2004 dataset based upon a) F-measure, b) recall, and c) precision

4.7 Comparative Techniques

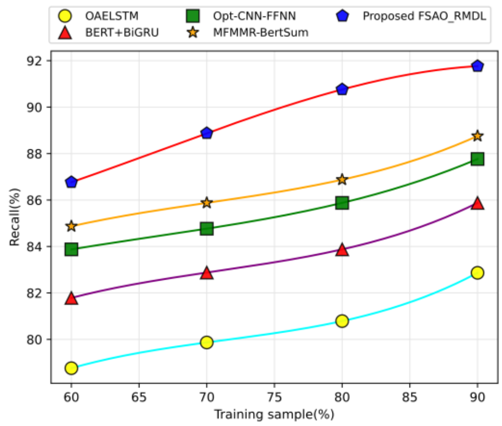
Classical abstractive summarization methods, such as OAELSTM [13], BERT+BiGRU [22], Opt-CNN-FFNN [12], and MFMMR-BertSum [3], are considered for evaluating the superiority of the FSAO_RMDL model in abstractive summarization. This study utilizes documents from the DUC 2002 and DUC 2004 databases with multiple training samples.

4.7.1 DUC 2002 database

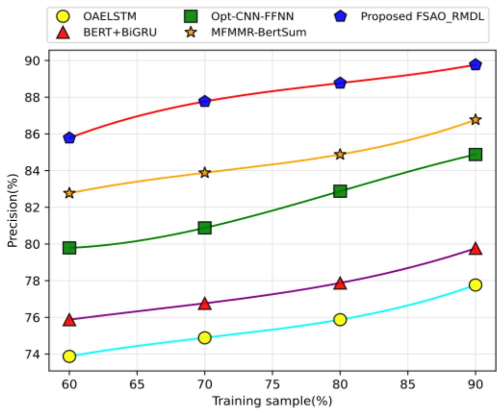
The FSAO_RMDL evaluated the consideration of the DUC 2002 database for several abstractive summarization models, as depicted in Fig. 6. The investigation by FSAO_RMDL regarding the F-measure is illustrated in Fig. 6(a). In particular, the F-measure recorded by OAELSTM, BERT+BiGRU, Opt-CNN-FFNN, and MFMMR-BertSum is 78.255%, 80.765%, 84.350%, and 85.865%, respectively, assuming a training sample size of 80%. Meanwhile, the FSAO_RMDL model achieved a superior F-measure of 89.751%, outperforming the baseline methods by 12.81%, 10.01%, 6.02%, and 4.33%. Additionally, Fig. 6(b) represents the investigation by FSAO_RMDL regarding recall. The recall measured with 60% training sample by OAELSTM is 78.760%, BERT+BiGRU is 81.787%, Opt-CNN-FFNN is 83.876%, MFMMR-BertSum is 84.870%, and the FSAO_RMDL is 86.766%. This illustrates that the FSAO_RMDL recorded a superior performance of 9.23%, 5.74%, 3.33%, and 2.19%. Meanwhile, Fig. 6(c) specifies the investigation of FSAO_RMDL while considering the precision value. The FSAO_RMDL quantified a precision of 87.766%, which is higher by 14.67%, 12.53%, 7.85%, and 4.43% than the precision calculated by the classical approaches, such as OAELSTM, BERT+BiGRU, Opt-CNN-FFNN, and MFMMR-BertSum at 74.887%, 76.767%, 80.877%, and 83.879%, at 70% training sample.



(a)



(b)



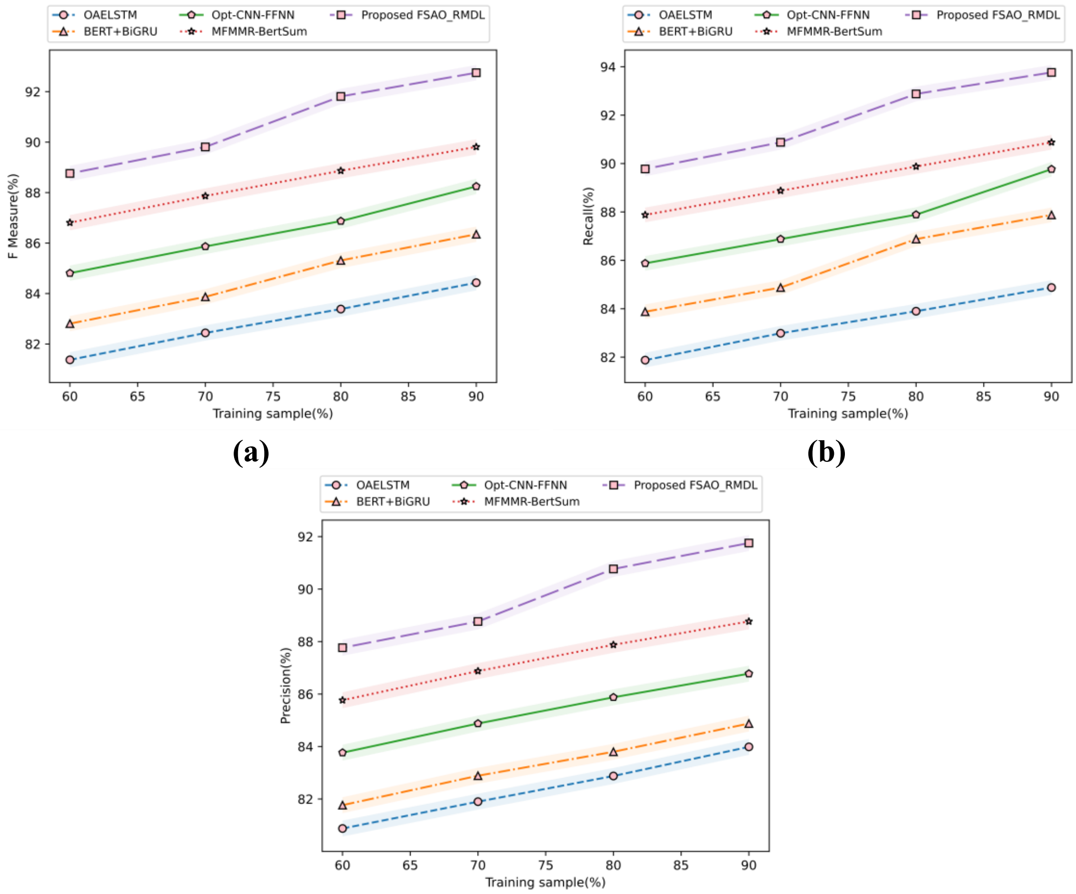
(c)

Fig. 6. Investigation of FSAO_RMDL for DUC 2002 database with a) F-measure, b) recall, and c) precision

4.7.2 For DUC 2004 dataset

The evaluation of the FSAO_RMDL, as represented in Fig. 7, is based on the document acquired from the DUC 2004 database. The analysis of the FSAO_RMDL concerning the F-measure is illustrated in Fig. 7(a). Accordingly, the F-measure computed by FSAO_RMDL is 91.810%, while the F-measures generated by the prevailing schemes, namely OAELSTM, BERT+BiGRU, Opt-CNN-FFNN, and MFMMR-BertSum, are 83.384%, 85.312%, 86.870%, and 88.866%, respectively. The performance of F-measure is enhanced by 9.18%, 7.08%, 5.38%, and 3.21% compared to prevailing techniques when assuming an 80% training sample size.

Fig. 7(b) displays the assessment of FSAO_RMDL regarding recall. The FSAO_RMDL recorded a recall of 90.877%, using a 70% training sample, whereas baseline approaches, including OAELSTM, BERT+BiGRU, Opt-CNN-FFNN, and MFMMR-BertSum, quantified recall at 82.987%, 84.876%, 86.877%, and 88.877%, respectively. This demonstrates that the FSAO_RMDL has achieved performance enhancements of 8.68%, 6.60%, 4.40%, and 2.20%. Moreover, in Fig. 7(c), the examination of FSAO_RMDL regarding the precision value is depicted. The precision computed by OAELSTM, BERT+BiGRU, Opt-CNN-FFNN, MFMMR-BertSum, and FSAO_RMDL is 80.876%, 81.766%, 83.766%, 85.765%, and 87.766%, respectively, with a 60% training sample. Consequently, FSAO_RMDL has achieved performance enhancements of 11.14%, 9.25%, 6.91%, and 4.43%.



(a) (b)

(c)

Fig. 7. Investigation of the FSAO_RMDL exploiting the DUC 2004 database based on (a) F-measure, (b) recall, and (c) precision

4.8 Ablation comparative study

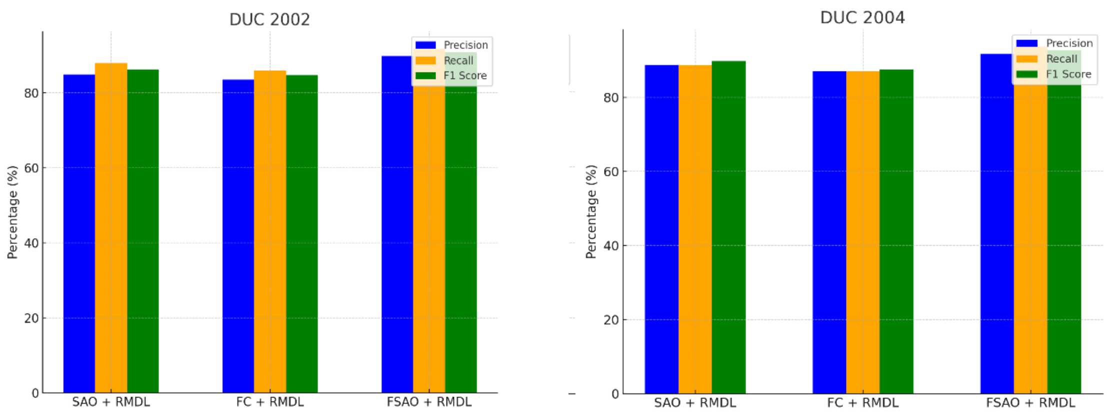
We have conducted an ablation study to compare the influence of each optimization method on the proposed system. Tests on SAO+RMDL and FC+RMDL have been implemented on both DUC2002 and DUC2004 datasets. This comparative study aims to reveal the influence of each optimizer on the proposed system and to gain a deep understanding of the optimizers' behaviors within the system. Table 1 below presents precision, recall, and F1 scores for each SAO+RMDL and FC+RMDL implemented on DUC2002 and DUC 2004 datasets.

SAO+RMDL gain on DUC 2002 scores of precision 84.9%, recall 87.9%. The scores with DUC 2004 were 88.8% and 88.7% for precision and recall, respectively. Notably, the proposed system, FSAO+RMDL, demonstrates a clear improvement of about +4-5% in accuracy, recall, and F1 scores. The fractal mathematics FC+RMDL registered with the DUC 2002 dataset precision of 83.5%, recall 86%, and F1 84.7%. At the same time, the model scored at DUC 2004 precision, recall, and F1 AS 87.0%, 87.4%, and 87.5%, respectively.

Table 1: Comparison between SAO, FC, FSAO influence on the RMDL

Optimization Method	Dataset	Precision (%)	Recall (%)	F1 Score (%)
SAO + RMDL	DUC 2002	84.9	87.9	86.3
FC + RMDL		83.5	86.0	84.7
FSAO + RMDL		89.8	91.8	90.8
SAO + RMDL	DUC 2004	88.8	88.7	89.8
FC + RMDL		87.0	87.0	87.5
FSAO + RMDL		91.8	93.7	92.8

Fig. 8(a) and 8(b) illustrate the three methods on both DUC 2002 and 2004 using precision, recall, and F1 scores. It can be observed the superiority of our proposed model over other paradigms. SAO simulates snow melting stochastically, preventing it from being trapped in local minima. On the other hand, FC integrates past iterative information, reaching convergence smoothly and rapidly. Thus, combining SAO optimizers with FC mathematics makes RMDL avoid overfitting with a smoothing performance on parameter updates, helping generalize the model to hidden data. Nevertheless, SAO alone may face slow convergence in fine-tuning since it could become trapped near optimal positions. Moreover, using merely FC hinders the ability to explore the minimum areas despite the convergence process's speed.



(a) **(b)**

Fig. 8. Investigation of the three methods exploiting the (a) DUC 2002 database and (b) DUC 2002 database

4.9 Comparative Discussion

This section compares the FSAO_RMDL based on several evaluation metrics, utilizing data from DUC 2002 and DUC 2004 databases. The outcomes corresponding to a 90% training sample are summarized in Table 2. With the dataset of DUC 2002, the proposed FSAO_RMDL model scored an F-measure of approximately 90.755%, a recall of 91.765%, and a precision of 89.766%. In comparison, previous methods, such as OAELSTM, BERT+BiGRU, Opt-CNN-FFNN, and MFMMR-BertSum, achieved F-measures of 80.235%, 82.708%, 86.297%, and 87.750%, respectively. Additionally, these techniques registered recall of 82.866%, 85.876%, 87.766%, and 88.756%. Lastly, the precision reported of OAELSTM is 77.766%, BERT+BiGRU is 79.765%, Opt-CNN-FFNN is 84.877%, and MFMMR-BertSum is 89.766%. With the DUC 2004 dataset, the FSAO_RMDL achieved a maximal F-measure, recall, and precision of 92.750%, 93.766%, and 91.755%, respectively. Moreover, the traditional methods, such as OAELSTM, BERT+BiGRU, Opt-CNN-FFNN, and MFMMR-BertSum, achieved F-measures of 84.429%, 86.350%, 88.246%, and 89.810%, respectively. Likewise, these classical techniques recorded recall of 84.877%, 87.877%, 89.766%, and 90.878%. Meanwhile, the precision calculated by OAELSTM is 83.987%, BERT+BiGRU is 84.876%, Opt-CNN-FFNN is 86.766%, and MFMMR-BertSum is 88.768%.

Table 2: Comparative discussion of FSAO_RMDL

Dataset	Metrics	OAELSTM	BERT+ BiGRU	Opt-CNN-FFNN	MFMMR-BertSum	Proposed FSAO_RMDL
DUC 2002	F-Measure (%)	80.235	82.708	86.297	87.750	90.755
	Recall (%)	82.866	85.876	87.766	88.756	91.765
	Precision (%)	77.766	79.765	84.877	86.766	89.766
DUC 2004	F-Measure (%)	84.429	86.350	88.246	89.810	92.750
	Recall (%)	84.877	87.877	89.766	90.878	93.766
	Precision (%)	83.987	84.876	86.776	88.768	91.755

In addition, RMDL offers a more straightforward and accurate technique for summarization by extracting key sentences. LLM is flexible, readable, and cohesive, producing abstract summaries that improve the model's performance. Conversely, FC facilitates better generalization when summarizing complex or lengthy documents. In contrast, SAO optimizes model parameters by mimicking the natural process of snow ablation, enabling more effective training. By integrating these two approaches, the FSAO_RMDL framework attains superior summarization outcomes, generating high-quality abstractive summaries with enhanced precision. Hence, it is crucial to address the aspect of inference times and energy consumption within this study. In particular, LLMs such as GPT-NeoX contain billions of parameters that need massive computational resources, resulting in long inference times and significant energy consumption. This, in turn, can decrease the efficiency of real-time applications, especially those with limited hardware. Conversely, the FSAO_RMDL system relies on extractive summarization through an RMDL approach, which targets preprocessing and refining the input to reduce the load on the abstractive LLM step. Thus, by accommodating the FSAO to tune the parameter, the system enhances efficiency by lowering inference time and computational cost compared to LLM standalone systems. As such, pure LLM-based summarizers are powerful yet less efficient since they take the whole input directly without filtering or feature compression. However, the proposed hybrid system trades off between computation cost and summarization quality by coupling optimized extractive models with LLMs. This makes it more appropriate for more resource-scarce settings or applications with shorter turnaround time demands.

5 Conclusion

Website document summarization generates a meaningful and concise summary of documents, such as news articles and blog posts. However, the conventional techniques employed for document summarization often fail to effectively create a summary that retains the key information and core content. Thus, a new approach called FSAO_RMDL is formulated for web document summarization. Initially, the input web document is passed through tokenization, where BERT is employed to tokenize the document. The features, namely sentence-to-sentence similarity, BOW, word2vec, TF-IDF, and sentence length, are mined. Subsequently, the extractive summarization is executed by RMDL, which is tuned using FSAO, an approach formulated by combining FC and SAO. Lastly, abstractive summarization is conducted by exploiting the GPT-NeoX LLM. Overall, the experimental results of FSAO_RMDL presented maximum recall, F-measure, and precision of 93.766%, 92.750%, and 91.755%, respectively. Nevertheless, future work intends to explore hybrid models, which can be beneficial for rephrasing the document into a readable and coherent summary.

References

[1] Zhang, M., Zhou, G., Yu, W., Huang, N., & Liu, W. (2022). A Comprehensive Survey of Abstractive Text Summarization Based on Deep Learning. *Computational Intelligence and Neuroscience*, 2022, e7132226. doi: https://doi.org/10.1155/2022/7132226.

[2] Al-Taani, A. (2021). Recent Advances in Arabic Automatic Text Summarization. *International Journal of Advances in Soft Computing and Its Applications*, *13*(3), 60–71. doi: https://10.15849/ijasca.211128.05 .

[3] Fan, J., Tian, X., Chengyao Lv, Zhang, S., Wang, Y., & Zhang, J. (2023). Extractive social media text summarization based on MFMMR-BertSum. *Array*, 20, 100322–100322. doi: https://doi.org/10.1016/j.array.2023.100322 .

[4] El-Kassas, W. S., Salama, C. R., Rafea, A. A., & Mohamed, H. K. (2020). Automatic Text Summarization: A Comprehensive Survey. *Expert Systems with Applications*, 165, 113679. doi: https://doi.org/10.1016/j.eswa.2020.113679 .

[5] Nikolaos Giarelis, Charalampos Mastrokostas, & Nikos Karacapilidis. (2023). Abstractive vs. Extractive Summarization: An Experimental Review. *Applied sciences*, 13(13), 7620–7620. doi: https://doi.org/10.3390/app13137620 .

[6] Divya S, Sripriya N, Andrew, J., & Mazzara, M. (2024). Unified extractive-abstractive summarization: a hybrid approach utilizing BERT and transformer models for enhanced document summarization. *PeerJ Computer Science*,10, e2424–e2424. doi: https://doi.org/10.7717/peerj-cs.2424 .

[7] Ma, C., Zhang, W. E., Guo, M., Wang, H., & Sheng, Q. Z. (2020). Multi-document Summarization via Deep Learning Techniques: A Survey. *arXiv (Cornell University)*. doi: https://doi.org/10.48550/arxiv.2011.04843 .

[8] Abualigah, L., Bashabsheh, M. Q., Alabool, H., & Shehab, M. (2019). Text Summarization: A Brief Review. In *Studies in Computational Intelligence* (pp. 1–15). Springer, Cham. doi: https://doi.org/10.1007/978-3-030-34614-0_1 .

[9] Elsaid, A., Mohammed, A., Ibrahim, L. F., & Sakre, M. M. (2022). A Comprehensive Review of Arabic Text Summarization. *IEEE Access*, 10, 38012–38030. doi: https://doi.org/10.1109/access.2022.3163292 .

[10] Mridha, M. F., Lima, A. A., Nur, K., Das, S. C., Hasan, M., & Kabir, M. M. (2021). A Survey of Automatic Text Summarization: Progress, Process and Challenges. *IEEE Access*, 9, 156043–156070. doi: https://doi.org/10.1109/access.2021.3129786 .

[11] Widyassari, A. P., Rustad, S., Shidik, G. F., Noersasongko, E., Syukur, A., Affandy, A., & Setiadi, D. R. I. M. (2022). Review of automatic text summarization techniques & methods. *Journal of King Saud University-Computer and Information Sciences*, *34*(4), 1029-1046. doi: https://doi.org/10.1016/j.jksuci.2020.05.006 .

[12] Wazery, Y. M., Saleh, M. E., & Ali, A. A. (2023). An optimized hybrid deep learning model based on word embeddings and statistical features for extractive summarization. *Journal of King Saud University - Computer and Information Sciences*, *35*(7), 101614. doi: https://doi.org/10.1016/j.jksuci.2023.101614 .

[13] Bharathi Mohan Gurusamy, Rangarajan, P. K., & Altalbe, A. (2024). Whale-optimized LSTM networks for enhanced automatic text summarization. *Frontiers in Artificial Intelligence*, *7*. doi: https://doi.org/10.3389/frai.2024.1399168 .

[14] Chen, X., Ouyang, C., Liu, Y., Luo, L., & Yang, X. (2018, September). A Hybrid Deep Learning Model for Text Classification. In *2018 14th International Conference on Semantics, Knowledge and Grids (SKG)*, (pp. 46–52). IEEE. doi: https://doi.org/10.1109/skg.2018.00014

[15] Kennedy, J., & Eberhart, R. (1995, November). Particle swarm optimization. In *Proceedings of ICNN'95 - International Conference on Neural Networks* (Vol. *4*, pp. 1942–1948). IEEE. doi: https://doi.org/10.1109/icnn.1995.488968 .

[16] Anam, S., Dwi Lestari, C. A., Tisna Amijaya, F. D., Tarno, H., & Fitriah, Z. (2024). New Tomato Leaf Disease Classification Method Based on DenseNet121 with Bat Algorithm Hyperparameters Optimization. *International Journal of Advances in Soft Computing & Its Applications*, *16*(2). doi: https://doi.org/10.15849/IJASCA.240730.03 .

[17] Yasear, S. A., & Ku-Mahamud, K. R. (2021). REVIEW OF THE MULTI-OBJECTIVE SWARM INTELLIGENCE OPTIMIZATION ALGORITHMS. *Journal of Information and Communication Technology*, *20*(2), 171–211. doi: https://doi.org/10.32890/jict2021.20.2.3 .

[18] Deng, L., & Liu, S. (2023). Snow ablation optimizer: A novel metaheuristic technique for numerical optimization and engineering design. *Expert Systems with Applications*, *225*, 120069–120069. doi: https://doi.org/10.1016/j.eswa.2023.120069.

[19] K, M., & V, G. (2024). Modeling of Snow Ablation Optimization Algorithm with Deep Learning Approach for Sentiment Classification on Social Media Corpus. *International Journal of Electronics and Communication Engineering*, *11*(12), 44–55. doi: https://doi.org/10.14445/23488549/ijece-v11i12p105 .

[20] Wu, Y., Xiang, C., Qian, H., & Zhou, P. (2024). Optimization of Bi-LSTM Photovoltaic Power Prediction Based on Improved Snow Ablation Optimization Algorithm. *Energies*, *17*(17), 4434–4434. doi: https://doi.org/10.3390/en17174434 .

[21] Jalil, Z., Nasir, M., Moutaz Alazab, Nasir, J., Amjad, T., & Abdullah Alqammaz. (2023). *Grapharizer: A Graph-Based Technique for Extractive Multi-Document Summarization*. *12*(8), 1895–1895. doi: https://doi.org/10.3390/electronics12081895 .

[22] Bano, S., Khalid, S., Nasser Mansoor Tairan, Shah, H., & Hasan Ali Khattak. (2023). Summarization of scholarly articles using BERT and BiGRU: Deep learning-based extractive approach. *Journal of King Saud University. Computer and Information Sciences/Maǧalaẗ Ǧam'aẗ Al-Malīk Saud : Ùlm Al-Ḥasib Wa Al-Ma'lumat*, *35*(9), 101739–101739. doi: https://doi.org/10.1016/j.jksuci.2023.101739 .

[23] Dai, W., & He, Q. (2024). Automatic summarization model based on clustering algorithm. *Scientific Reports*, *14*(1), 15302. doi: https://doi.org/10.1038/s41598-024-66306-4 .

[24] Moro, G., Luca Ragazzi, Valgimigli, L., Giacomo Frisoni, Sartori, C., & Marfia, G. (2023). Efficient Memory-Enhanced Transformer for Long-Document Summarization in Low-Resource Regimes. *Sensors*, *23*(7), 3542–3542. doi: https://doi.org/10.3390/s23073542 .

[25] Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. ArXiv.org. https://arxiv.org/abs/1810.04805#

[26] Suanmali, L., Salim, N., & Binwahlan, M. S. (2009, May). Feature-Based Sentence Extraction Using Fuzzy Inference Rules. In *2009 International Conference on Signal Processing Systems* (pp. 511–515). IEEE. doi: https://doi.org/10.1109/icsps.2009.156 .

[27] Fatima, Z., Zardari, S., Fahim, M., Andleeb Siddiqui, M., Ibrahim, A. A. A., Nisar, K., & Naz, L. F. (2022). A novel approach for semantic extractive text summarization. *Applied Sciences*, *12*(9), 4479. doi: https://doi.org/10.3390/app12094479 .

[28] "Word2Vec Model Analysis for Semantic Similarities in English Words," *Procedia Computer Science*, vol. 157, pp. 160–167, Jan. 2019, doi: https://doi.org/10.1016/j.procs.2019.08.153 .

[29] Jalilifard, A., Caridá, V. F., Mansano, A. F., Cristo, R. S., & da Fonseca, F. P. C. (2021, June). Semantic sensitive TF-IDF to determine word relevance in documents. In *Advances in Computing and Network Communications: Proceedings of CoCoNet 2020 (Vol.2* , pp. 327-337). Springer Singapore. doi: http://dx.doi.org/10.48550/arXiv.2001.09896 .

[30] Vidyasri, P., & Suresh, S. (2025). FDN-SA: Fuzzy deep neural-stacked autoencoder-based phishing attack detection in social engineering. *Computers & Security*, *148,* 104188. doi: https://doi.org/10.1016/j.cose.2024.104188 .

[31] Kowsari, K., Heidarysafa, M., Brown, D. E., Meimandi, K. J., & Barnes, L. E. (2018, April). Rmdl: Random multimodel deep learning for classification. In *Proceedings of the 2nd international conference on information system and data mining* (pp. 19-28). ACM. doi: https://doi.org/10.18178/ijmlc.2018.8.4.703 .

[32] Bhaladhare, P. R., & Jinwala, D. C. (2016). A Clustering Approach Using Fractional Calculus-Bacterial Foraging Optimization Algorithm for k-Anonymization in Privacy Preserving Data Mining. *International Journal of Information Security and Privacy*, *10*(1), 45–65. doi: https://doi.org/10.4018/ijisp.2016010103.

[33] Deng, L., & Liu, S. (2023). Snow ablation optimizer: A novel metaheuristic technique for numerical optimization and engineering design. *Expert Systems with Applications*, *225*, 120069–120069. doi: https://doi.org/10.1016/j.eswa.2023.120069.

[34] Over, P., Dang, H., & Harman, D. (2007). DUC in context. *Information Processing & Management*, *43*(6), 1506-1520.
[35] "Document Understanding Conferences - Past Data," *Nist.gov*, 2025. https://www-nlpir.nist.gov/projects/duc/data.html (accessed Mar. 23, 2025).
[36] Kaur, S., Awasthi, L. K., Sangal, A. L., & Dhiman, G. (2020). Tunicate Swarm Algorithm: A new bio-inspired based metaheuristic paradigm for global optimization. *Engineering Applications of Artificial Intelligence*, *90*, 103541. doi: https://doi.org/10.1016/j.engappai.2020.103541.
[37] Dutta, T., Bhattacharyya, S., Dey, S., & Platos, J. (2020). Border collie optimization. *IEEE access*, *8*, 109177-109197. doi: https://doi.org/10.1109/ACCESS.2020.2999540 .
[38] JPan, J. S., Zhang, L. G., Wang, R. B., Snášel, V., & Chu, S. C. (2022). Gannet optimization algorithm: A new metaheuristic algorithm for solving engineering optimization problems. *Mathematics and Computers in Simulation*, *202*, 343-373. doi: https://doi.org/10.1016/j.matcom.2022.06.007 .

Notes on contributors

Ahmed bahaaulddin A. alwahhab is a lecturer at the Technical College of management since 2005. He received his master's degree in computer science from informatics institute for high studies in Iraq, and a Ph.D. degree in computer science from University of Technology in Iraq. He is interested in the field of artificial intelligence, natural language processing, and recommender systems.

Vian T. Sabeeh is a lecturer at the Technical College of management since 2005. She received her master's degree in computer science from informatics institute for high studies in Iraq, and a Ph.D. degree in computer science from Oakland university in USA. She is interested in the field of artificial intelligence, natural language processing, computer vision and semantic web.