Vol. 9 No. 5
October 2025

Opportunity Identification & Problem Solving

INTERNATIONAL JOURNAL OF SYSTEMATIC INNOVATION

ISSN (Print): 2077-7973
ISSN (Online): 2077-8767
DOI: 10.6977/IJoSI.202510_9(5)

ACCSCIENCE PUBLISHING

ISSN (Print): 2077-7973
ISSN (Online): 2077-8767
DOI: 10.6977/IJoSI.202510_9(5)

The International Journal of Systematic Innovation

Publisher:

The Society of Systematic Innovation

Editorial Office:

- *International Journal of Systematic Innovation* Editorial Office
- 9 Raffles Place, Republic Plaza 1 #06-00 Singapore 048619
- Email: ijosi.office@accscience.sg
- Website: https://ijosi.org/journal/IJOSI
- Tel: +65 8182 1586

ISSN (Print): 2077-7973
ISSN (Online): 2077-8767
DOI: 10.6977/IJoSI.202510_9(5)

INTERNATIONAL JOURNAL OF SYSTEMATIC INNOVATION

CONTENTS

OCTOBER 2025 VOLUME 9 ISSUE 5

FULL PAPERS

DOI: 10.6977/IJoSI.202510_9(5).0001
S. Upadhyay & H. K. Soni/Int. J. Systematic Innovation, 9(5), 1-13 (2025)

An ensemble learning framework for text summarization based on an improved multilayer extreme learning machine autoencoder

Sunil Upadhyay*, Hemant Kumar Soni

Department of Computer Science and Engineering, Amity School of Engineering and Technology, Amity University Madhya Pradesh, Gwalior, Madhya Pradesh, India

*Corresponding author E-mail: supadhyay@gwa.amity.edu

(Received 05 October 2024; Final version received 07 July 2025; Accepted 05 August 2025)

Abstract

The massive growth of electronic data has created a demand for efficient tools to manage information and support fast decision-making. Automatic text summarization (ATS) addresses this by condensing large texts into concise, relevant summaries rapidly. ATS methods are categorized as extractive, abstractive, or hybrid. Extractive techniques select key sentences from input documents, whereas abstractive techniques generate new sentences to capture meaning. Hybrid methods combine both strategies. However, despite numerous suggested techniques, machine-generated summaries often fail to match the accuracy and coherence of human-written summaries. This study reviewed existing ATS techniques and highlighted their limitations, particularly high computational costs and low training efficiency. To address these problems, this study proposed an improved multilayer extreme learning machine autoencoder (MLELM–AE) and an ensemble learning framework that integrates four machine learning models: Sentence-bidirectional encoder representations from transformers, autoencoder, variational autoencoder, and the improved MLELM–AE. The proposed framework generates summaries through cosine similarity evaluation, followed by voting-based fusion, re-ranking, and optimal sentence selection. Experimental results showed that the proposed improved MLELM–AE model achieved strong performance, with an execution time of 50,015 ms and a recall-oriented understudy for gisting evaluation 1 score of 0.865145. These findings demonstrate that the proposed ensemble framework delivers more accurate and efficient summaries, offering a promising advancement in ATS.

Keywords: Automatic Text Summarization, Bidirectional Encoder Representations from Transformers, Deep Neural Networks, Multilayer Extreme Learning Machine Autoencoder, Word Embedding, Word2vec

1. Introduction

In today's era, the Internet has huge amounts of data due to the rapid expansion of web-based electronic documents. The proliferation of this vast volume of data makes it complicated to collect pertinent information efficiently. In view of the huge amount of text documents, gathering and processing primary data from various resources is a complex and exhaustive task, often exceeding human capacity. This challenge has motivated researchers to develop techniques for automatic text summarization (ATS), which aim to condense large volumes of text into concise summaries while preserving meaning and context. Over the past several decades, several information retrieval techniques have been explored to address this problem.

Text summarization is a rapidly growing and challenging task in natural language processing (NLP). It aims to produce a condensed version of a document that retains the key ideas of the original text, facilitating the comprehension of these ideas (Mitra et al., 2000). ATS is particularly valuable because manual text summarization is tedious and time-consuming. In the NLP domain, summarization also serves as an intermediary step to reduce text size and complexity. Key application areas of text summarization include text classification, question and answer, legal document summarization, social media text summarization, and headline/title creation.

Text summarization can be categorized by output type into two main approaches (Gambhir & Gupta,

2017): Extractive text summarization and abstractive text summarization. Extractive text summarization is the most widespread approach to text summarization. It extracts important textual units, such as phrases, words, and sentences, based on linguistic and mathematical features to form a summary. On the other hand, abstractive text summarization generates summaries closer to human-written summaries by creating semantic representations and producing new sentences, rather than merely reordering existing ones. Summaries generated by these methods are generally grammatically correct. Therefore, these techniques are not limited to simply picking and reordering sentences from the original text.

Summarization can also be classified by input type (single vs. multi-document) or purpose (generic, domain-specific, or query-based) (Zajic et al., 2008). Generic summarization captures broad themes and addresses a wide community of users; domain-specific summarization incorporates knowledge of specialized fields, such as law or biomedicine, while query-based summarization tailors the output to user needs.

Despite progress, ATS still faces significant challenges. Key issues include encoding high-level semantic structures, handling large input dimensionality, managing out-of-vocabulary words, and ensuring accurate part-of-speech tagging. Conventional machine learning approaches often struggle with these challenges due to their shallow architecture and restricted capability for hierarchical feature learning. Neural network-based methods have improved semantic modeling, but they still face limitations, including computational inefficiency, noisy training data, and the omission of important sentences due to score-based selection.

To overcome these limitations, this study proposed an improved novel ensemble learning-based ATS, called improved multilayer extreme learning machine–autoencoder (MLELM–AE). The multilayer architecture enhances the ability to learn deep and abstract features, improving the identification of salient information. In addition, the proposed algorithm incorporates end-to-end training using backpropagation, allowing iterative refinement of hidden layers and better generalization compared to conventional extreme learning machine (ELM)-based approaches. The AE framework ensures efficient reconstruction, enabling efficient dimensionality reduction while retaining important information for producing high-quality summaries.

The proposed ensemble framework integrates multiple models, including the improved MLELM–AE, AE, variational AE (VAE), and sentence-bidirectional encoder representations from transformers (SBERT). It employs data transformation steps, such as clustering, topic modeling, term frequency–inverse document frequency (TF–IDF) analysis, and frequent term selection to enhance text representation. Entity-focused sentences are captured through topic modeling, while a re-ranking mechanism ensures optimal sentence selection for the final summary. Overall, the proposed ensemble approach significantly advances ATS by combining semantic entity extraction, robust feature learning, and effective sentence re-evaluation.

The remainder of this paper is structured as follows: Section 2 reviews existing works, Section 3 details the proposed methodology, Section 4 presents results and discussion, and Section 5 concludes with future scopes.

2. Related Works

The work by Toprak & Turan (2025) demonstrated an automatic abstractive document summarization framework based on transformers and sentence grouping. The collected dataset was pre-processed and then utilized to train the transformer model. Then, the transformer model proficiently summarized the text. This approach obtained a SimHash text similarity of 93.2%, indicating a high effectiveness and low complexity. However, this model suffered from considerable information loss.

Khan et al. (2025) implemented a hybrid deep learning-based next-generation text summarization for psychological data. Text-to-text transfer transformer (T5) and long short-term memory (LSTM) were employed to perform advanced text summarization. This approach achieved an accuracy, precision, and recall of 74%, 72%, and 72%, respectively, indicating its supremacy. However, the framework had high computational complexity owing to the hybrid scheme.

Alotaibi & Nadeem (2025) introduced an Arabic aspect-based sentiment analysis and abstractive text summarization of traffic services using an unsupervised-centric approach. A fine-tuned AraBART algorithm was employed to perform abstractive text summarization. This algorithm achieved 92.13% precision and 92.07% recall, indicating its high efficacy. However, the model struggled to handle the text from various domains.

Onan & Alhumyani (2024a) propounded an extractive text summarization framework using fuzzy topic modeling and bidirectional encoder representations from transformers (BERT). Here, fuzzy logic was used to improve topic modeling, thereby capturing a nuanced representation of word-topic relationships. This algorithm obtained recall-oriented understudy for gisting evaluation 1 (ROUGE-1) and ROUGE-2 scores of 45.3774 and 24.1808, respectively. It significantly provided high-quality text summaries. However, the framework was ineffective due to the lack of interpretability.

In the work by Onan & Alhumyani (2024b), they implemented a multi-element contextual hypergraph extractive summarizer (MCHES) to perform extractive text summarization. MCHES effectively constructed a contextual hypergraph, showing semantic and discourse hyperedges. The approach achieved an ROUGE-1 score of 44.321 and an ROUGE-2 score of 19.129, indicating its impressive performance in extractive summarization. However, the framework had a maximum risk of bias amplification.

Hassan et al. (2024) demonstrated an approach of extractive text summarization using NLP with an optimal deep learning (ETS-NLPODL) model. The research analysis of various parameters indicated that the ETS-NLPODL approach achieved excellent performance compared to other models regarding diverse performance measures.

Hernández-Castañeda et al. (2023) designed a fitness function based on genetic programming to generate ATS. The experimental outcomes clearly showed that the grouping of lexical and semantic information (LDA+Doc2Vec+TF–IDF) achieved exceptional outcomes in identifying key ideas to form a summary.

Dilawari et al. (2023) proposed a model for both extractive and abstractive summarizations, named as automatic feature-rich model architecture comprises a hierarchical bidirectional LSTM. The results demonstrated that the model outperformed existing techniques, with a ROUGE score of 37.76%, high generality, and high sapiential.

An improved English text summary algorithm based on association semantic rules was proposed in a previous study (Wan, 2018). The method mined relative features among English sentences and phrases, implemented keyword extraction in English abstracts, and applied semantic relevance analysis with association rules distinction, grounded in knowledge theory. Semantic rules were further mined from English teaching texts. The outcome of the replication showed that the technique could accurately extract summaries with improved convergence and output accuracy. This demonstrates strong application value for efficiently reading English texts and gathering important information.

Zenkert et al. (2018) proposed the multidimensional knowledge representation structure. The fallouts of analytics using individual methods for text mining, such as named person recognition, sentiment analysis, and topic detection, were integrated into a knowledge base as dimensions to support knowledge exploration, vision, and computer-aided written tasks. This framework supports cross-dimensional exploration and provides a novel approach for summarization and knowledge discovery.

Similarly, Prameswari et al. (2018) combined sentiment analysis and summary generation, applying their method to hotel reviews in Bali and Labuan Bajo. Their model achieved a rating accuracy of 78% with a Davies–Bouldin index of 0.071, demonstrating potential benefits for the Indonesian tourism industry.

Jain et al. (2017) proposed a neural network-based extractive summarization function, testing on the Document Understanding Conferences (DUC) 2002 dataset. Their approach outperformed four online summarizers in ROUGE evaluations, indicating the importance of robust feature extraction for summary generation. The scale and complexity of training datasets and additional exact methods to convert abstract summaries into extractive summaries will further improve the model.

In clustering-based approaches, Pradip & Patil (2016) developed a hierarchical sentence clustering algorithm to address instability, complexity, and sensitivity issues in traditional methods. Any type of relational clustering algorithm may work with an implemented hierarchical clustering algorithm. The general text mining algorithm can also be used. Experimental results demonstrate that hierarchical clustering was useful and yielded improved results for text documents.

Akter et al. (2017) presented a text summarization method that extracts significant phrases from single or multiple Bengali documents, which were prepared by processes, such as tokenization or interrupt operations. The word score was then determined using the TF–IDF weighting, and the sentence value was calculated with location. For sentence score calculation, the term skeleton and cue were also considered. K-means clustering was used to summarize many or a single document in a final form. Their method reduced redundancy and improved run-time complexity compared to existing extractive approaches.

Jadhav et al. (2019) designed a bidirectional recurrent neural network (RNN)-based encoder-decoder model that identifies key phrases and generates coherent summaries. Initially, key phrases were listed and arranged in a consolidated report. Given the measurable and semantic highlights of sentences, the sense of the sentence was chosen. This shorter representation was then passed through an encoder-decoder template to produce a description of the entire document. The projected model efficiently created a concise and linguistically accurate synthesis by recognizing the content and disclosing it in its terms. The proposed methodology only selected related terms and passed them to a bidirectional RNN to define the central ideas of the article and to represent them.

The ATS problem consists of two main tasks: Single-document and multi-document summarization.

In the case of a single document, input and summarized details are extracted from a specific document, whereas for multiple documents, summaries are generated based on a shared theme. A recent statistical approach was proposed by Madhuri and Kumar (2019) to perform extractive text summarization on single documents. The method of extracting sentences was presented, providing a brief overview of the input text. Phrases were categorized by weight assignment. Highly ranked phrases were then selected to form the final summary, which can also be converted into audio output.

Document review aims to condense the source text into a short and succinct form while preserving accuracy and general significance. Dave & Jaswal (2015) proposed an abstractive summary approach that generates compact and human-readable summaries using WordNet ontology derived from extractive summaries. The generated summaries were grammatically correct and more coherent for human readers.

Elbarougy et al. (2020) introduced an Arabic text summarization method, a graphical system with text expressed on its vertices. An improved PageRank algorithm was applied with initial node scores and multiple iterations to generate optimal summaries while eliminating redundancies. Using the Essex Arabic Summaries Corpus for evaluation, this method outperformed TextRank and LexRank, achieving a final F-measure of 67.98, which surpassed earlier approaches.

Collecting textual information is a challenging activity in biomedical text synthesis. Moradi et al. (2020) proposed a method leveraging BERT-based contextual embeddings to capture the semantic information of biomedical texts. Their deep learning model clustered sentences using BERT and selected the most relevant ones for summary generation. Evaluation with the ROUGE toolkit demonstrated significant improvements in biomedical text synthesis, outperforming other domain-independent approaches.

A multi-target optimization method has contributed to ATS over the years. Sanchez-Gomez et al. (2019) applied a multi-objective artificial bee colony (MOABC) algorithm, incorporating parallelization strategies. Comparative experiments on DUC datasets showed that their asynchronous structure significantly enhanced performance, achieving over 55 times quicker with 64 threads and an efficiency of 86.72%, outperforming traditional synchronous methods.

Qaroush et al. (2021) proposed automated and extractive general Arabic single-document summarizing techniques to construct comprehensive summary details. The proposed extractive methods used statistical and semantic features to evaluate sentence value, diversity, and exposure. Two summarizing techniques were also used to construct a description and then exploited built characteristics, such as score and machine learning supervision. Performance of the proposed technique was tested using the ROUGE metrics, yielding superior results in terms of accuracy, retrieval, and F-score compared to related works.Present graph-based extractive summarization methods represent corpus sentences as nodes, with edges depicting lexical similarity between sentences (Van Lierde & Chow, 2019). However, such approaches cannot adequately capture semantic similarities, since sentences may convey related information using different words. To address this, Van Lierde & Chow (2019) proposed extracting semantical similarities based on topical representations. They introduced a topic model to infer the distribution of hierarchical, context-influenced sentences. Since each concept establishes semantic relationships across sentences by assigning degrees of membership, the authors further proposed a fluid hypergraph model, where nodes represent sentences and fuzzy hyperedges. Sentence collections were then extracted to produce comprehensive summaries while simultaneously optimizing user-defined query relevance, centrality within the hypergraph, and topic coverage. To solve this optimization problem, they developed an algorithm based on submodular function theory. A thorough comparison with other graphic summarizers demonstrated the superiority of their strategy in the coverage of summaries.

Extractive multifocal approaches aim to synthesize key material while reducing redundancy. One promising avenue is multi-objective optimization, which naturally fits the summarization problem (Sanchez-Gomez et al., 2019). This method produces a set of non-dominated solutions or Pareto sequences, though ultimately only one summary is selected. To address this, post-Pareto analyses were performed using various methods, including hypervolume maximization, minimum distance from all points, minimum distance from an ideal point, and a consensus solution. Experiments conducted on DUC datasets and evaluated using ROUGE metrics revealed that the consensus approach outperformed others, improving ROUGE scores by 10.68–27.32%.

In another study, Alami et al. (2019) enhanced ATS efficiency using unregulated deep neural networks combined with a word embedding approach. First, they built a word definition on word integration and demonstrated that the representation of Word2Vec was better than that of traditional bag-of-words (BOW). Second, by combining Word2Vec and unmonitored functional learning approaches, they offered alternative models for incorporating information from various sources. They revealed that uncontrolled neural network models trained on the representation

of Word2Vec were enhanced compared to those trained on BOW models. Third, they described three ensembles: (i) Majority voting between Word2Vec and BOW, (ii) aggregation of BOW with unsupervised neural network outputs, and (iii) a combined ensemble of Word2Vec and unattended neural networks. Results showed that ensemble techniques enhanced ATS performance, with Word2Vec-based ensembles consistently outperforming BOW-based models. Comparative evaluations across two publicly accessible datasets confirmed that Word2Vec ensemble methods yielded the best results, surpassing all studied models in effectiveness.

Abstractive text summarization is a more challenging task than extractive summarization, as it requires generating paraphrased text that conveys the entire meaning of the source. Nonetheless, it typically yields more natural summaries with improved cohesion between sentences. Adelia et al. (2019) demonstrated that RNNs can effectively produce abstractive summaries in both English and Chinese. In their study, a bidirectional gated recurrent unit RNN architecture was used to capture the effect of surrounding words on generated summaries. Applying a similar method to Bahasa Indonesia, they showed that the model could generate summaries closely resembling human-written abstracts, outperforming purely extractive approaches. Their findings suggest that RNN-based abstractive models can achieve strong comprehension of source texts to support high-quality summary generation.

Building on this line of work, Yao et al. (2018) proposed a dual-encoder sequence-to-sequence attentional model for abstractive summarization. Unlike previous research that relied on a single encoder, their model incorporated both a primary encoder, which performed coarse-grained encoding, and a secondary encoder, which provided fine-grained encoding based on raw input and previously generated outputs. By combining both levels, the model reduced redundancy and improved handling of long sequences. The test outcomes of two complicated datasets (DUC 2004 and CNN Daily Mail) revealed that their hybrid model of encoding outperformed existing methods.

Du & Huo (2020) focused on fuzzy logic rules, multi-feature analysis, and genetic algorithms to develop a new automated synthesis paradigm for news texts. Since news articles often contain distinctive elements, such as time, place, and characters, word features were first extracted, and those surpassing a threshold score were identified as keywords. A linear combination of these characteristics revealed the meaning of each sentence, and each feature evaluated the genetic algorithms. Using fuzzy logic, the system generated automated summaries. The simulation results on the DUC 2002 dataset, evaluated with the ROUGE tool, demonstrated that the proposed method outperformed several baseline approaches, including Microsoft Word, System19, System2, System30, single-document summarization–neural network with a genetic algorithm, general context decoder, self-organizing map, and support vector machine ranking.

Alzuhair & Al-Dhelaan (2019) proposed combining multiple graph-based methods to enhance the quality of extractive summary outcomes. Given the widespread use of graph-based techniques in NLP, they developed a hybrid approach that integrates two graph-based techniques (four different weighting methods and two graph methods). To merge the results, both the arithmetic mean and harmonic mean were tested. Experiments conducted on the DUC 2003 and DUC 2004 datasets, evaluated using the ROUGE toolkit, and revealed that the harmonic mean outperformed the arithmetic mean. Furthermore, the hybrid method demonstrated significant improvements over baseline models and several state-of-the-art approaches when combined with weighting schemes.

Building on sequence-to-sequence frameworks, Ding et al. (2020) sought to optimize traditional sequence mapping and semantic representation for abstractive summarization. Their proposed method enhanced the model's semantic comprehension of source texts and improved the coherence of generated summaries. The method was validated on two benchmark datasets, large-scale Chinese short text summarization (LCSTS) and SOGOU datasets, where experimental results showed ROUGE score improvements of 10–13% compared to existing algorithms. These findings demonstrate that optimizing semantic representation can substantially enhance both the accuracy and readability of generated summaries.

Similarly, Liang et al. (2020) introduced an abstractive summarization model tailored for social media texts using a selective sequence-to-sequence (i.e., Seq2Seq) framework. To improve content filtering, a discerning gate was added after the encoder to eliminate irrelevant or redundant information. In addition, they combined inter-entropy with enhancement learning to directly optimize ROUGE scores. Evaluations on the LCSTS dataset demonstrated that their model achieved F1-score gains of 2.6% for ROUGE-1, 2.1% for ROUGE-2, and 2.0% for ROUGE-L compared with the baseline Seq2seq model.

El-Kassas et al. (2020) introduced EdgeSumm, a novel extractive graph-based architecture designed to optimize ATS for single documents. The framework relies on four proposed algorithms, with the first constructing a novel text graph model (NTGM) from the input document. The second and third algorithms identify candidate sentences from the constructed text graph, while the fourth finalizes the summary selection. Unlike many existing methods, EdgeSumm

is domain-independent and unsupervised, requiring no training data. The model was evaluated on the standard DUC 2001 and DUC 2002 datasets using the ROUGE evaluation toolkit. Results showed that EdgeSumm achieved the highest ROUGE scores on DUC 2001, and on DUC 2002, it outperformed several state-of-the-art ATS frameworks by margins of 1.2–4.7% in ROUGE-1 and ROUGE-L. The proposed framework also delivered highly competitive performance on ROUGE-2 and ROUGE-SU4, confirming its robustness and efficiency.

Automatic review summarization has emerged as an effective approach to improving information processing for travelers. However, many review texts contain vague or non-sentimental content, limiting the effectiveness of sentiment-based methods. To address this, Tsai et al. (2020) proposed a systematic framework that first identifies useful reviews through a classifier and then categorizes sentences into six hotel-related features. Subsequently, the polarity of each sentence is evaluated for analytical summaries. Experimental results demonstrated that the proposed method outperformed other methods, producing more accurate and informative summaries of hotel reviews.

Joshi et al. (2019) proposed SummCoder, a novel extractive method for single-document summarization. This framework is based on three sentence-level analysis techniques: Sentence position, content relevance, and sentence novelty. Content relevance is computed using a deep AE network, while novelty is measured through semantic similarity between sentence embeddings in distributed space. Sentence position is modeled using a hand-designed weighting function that assigns higher significance to earlier sentences, with adjustments based on document length. Final summaries are generated by ranking sentences according to a fused score from these three metrics. To support evaluation, the authors introduced the Tor Illegal Documents Summarization (TIDSumm) dataset, specifically built to assist law enforcement agencies in analyzing web documents from the Tor network. Empirical outcomes showed that SummCoder performed on par with or better than, several state-of-the-art approaches across various ROUGE metrics on DUC 2002, blog summarization datasets, and TIDSumm.

Jindal & Kaur (2020) developed an unsupervised approach to summarizing bug reports, aiming to capture both overall content and specific software-related details. Their method begins with automated keyword extraction using TF–IDF, followed by ranking of key sentences. To reduce redundancy, fluid C-means clustering is applied with thresholding, and a rule motor informed by domain knowledge selects the most relevant sentences. Additional hierarchical clustering is employed for re-ranking and improving coherence. The proposed approach was evaluated on the Apache bug report corpus (APBRC) and bug report corpus (BRC) using metrics, such as precision, recall, pyramid precision, and F-score. Experimental results showed substantial improvements over baseline methods, including BRC and logistic regression with crowdsourcing attributes, as well as existing unsupervised methods, such as Hurried and Centroid. The APBRC evaluation reported 78.22% precision, 82.18% recall, 80.10% F-score, and 81.66 pyramid precision, highlighting the method's strong performance in generating cohesive and comprehensive summaries.

3. Methodology

3.1. Improved MLELM–AE

The improved MLELM–AE is a hybrid neural network model that integrates the fast training ability of ELMs with the deep feature learning capability of AEs. Conventional ELMs typically employ only a single hidden layer and compute output weights analytically, which enables extremely fast training but restricts their ability to capture complex patterns. To address these issues, the proposed improved MLELM–AE introduces a multilayer architecture structure as a deep AE. This design enables the network to learn hierarchical and abstract depictions of input data.

This approach is particularly designed for tasks, such as ATS and dimensionality reduction, where capturing deep semantic features is important. The algorithm begins by defining the network architecture, including the input layer size, output layer size (typically matching the input in AEs), and the configuration of hidden layers. Bias vectors and weight matrices are set randomly for every hidden layer. During the forward pass, input data are propagated through each hidden layer using a non-linear activation function, enabling the model to capture complex patterns and relationships within the data. The output layer then attempts to reconstruct the original input, consistent with the fundamental nature of an AE.

In contrast to traditional ELMs that depend exclusively on closed-form solutions to compute output weights, the proposed model adopts an iterative optimization approach. For a pre-defined number of iterations, the model computes the reconstruction error (the difference between the input and the reconstructed output) and updates the weights using a specified learning rate. This hybrid approach preserves the computational efficiency of ELMs in the hidden layers while enabling the model to adaptively fine-tune the output layer weights. Compared to traditional ELM or single-layer AEs, the proposed model demonstrates improved convergence.

The innovation of the improved MLELM–AE stems from its integration of the fast learning capacity of ELMs with the deep feature extraction strength of multilayer AE. This design leverages fixed random weights in the hidden layers while allowing adaptive updates in the output layer, thereby enabling deep feature extraction at a minimal computational cost. By employing reconstruction loss as the training objective, the model is particularly well-suited for unsupervised learning applications. Compared with shallow architectures, it demonstrates superior ability to capture complex data representations, providing an efficient balance among performance, training speed, and architectural simplicity. The flow of data between hidden layers is mathematically formulated in Eq. (1):

$$H_i \quad g(\ _i)\ H_i \tag{1}$$

where β_i is the output weights, T is equivalent to the input data X at the first layer of MLELM, β_{i+1} is the output weight matrix of the i^{th} hidden layer, and $i+1^{th}$ layer weights are the outputs of MLELM. Regularized least squares were used for output layer weight calculation of MLELM.

The proposed improved MLELM–AE algorithm introduces numerous key novelties over conventional models, such as ELMs and AE. The main contributions are outlined in Table 1.

3.2. Algorithm of the Improved MLELM–AE

Input:
- Training data: TRx
- Number of iterations: niterations
- Learning rate: lrate

Output:
- Improved MLELM–AE trained model: A

a. Initialization

1. Describe input dimensions:
   - input_size, hsizes, osize (sizes of input, hidden layers, and output, respectively)
2. Initialize weights and biases for each layer:
   - For each layer k:
     - G[k] = random matrix of size (hsizes[k], input_size if k == 0 else hsizes[k-1])
     - h[k] = random matrix of size (hsizes[k], 1)
3. Initialize output weights and biases:
   - G_out = random matrix of size (osize, hsizes[-1])
   - h_out = random matrix of size (osize, 1)

b. Train the network

For each iteration in range niterations:

1. Forward pass:
   - Initialize activations = [input_data]
   - For each layer k:
     - Compute:
       Y = activation_function(G[k] * Y + h[k])
     - Append Y to activations
   - Compute final output:
     output = G_out * Y + h_out
2. Calculate loss:
   - Compute loss:
     loss = mean((output - activations[0])^2)
3. Backward pass:

i. Compute output error and delta:
- oerror = output - activations[0]
- odelta = oerror

ii. Update output weights and biases:
- G_out -= lrate * (odelta * activations[-1].T)
- h_out -= lrate * mean(odelta, axis=1, keepdims=True)

iii. Compute hidden layer errors:
- Initialize herrors = [odelta]
- For each layer k in reverse:
  - hidden_error = G[k+1].T * herrors[-1]
  - hdelta = hidden_error * activations[k+1] * (1 - activations[k+1])
  - Append hdelta to hdeltas and hidden_error to herrors

iv. Update weights and biases for hidden layers:
- For each layer k:
  - G[k] -= lrate * (hdeltas[k] * activations[k].T)
  - h[k] -= lrate * mean(hdeltas[k], axis=1, keepdims=True)

c. Return the trained model
- Return the trained model A (Improved MLELM–AE)

3.3. Ensemble Learning Framework for Text Summarization

In the proposed ensemble learning framework, the enhancement of sentence representations and the improvement of output summaries' quality are achieved using an ensemble of deep learning models: The improved MLELM–AE, SBERT, AE, and VAE (Fig. 1). From the output of these models, cosine similarity scores are computed, followed by a voting-based fusion strategy, re-ranking, and optimal sentence selection.

In this approach, Word2Vec and SBERT semantic embedding models are first used to convert the input document into dense vector representations, effectively capturing the contextual relationships within the text. These embeddings are then passed through four parallel encoding modules: SBERT, AE, VAE, and the improved MLELM–AE. Each encoder

Table 1. Novelty of the proposed improved MLELM–AE algorithm

Feature	Traditional ELM	AE	Improved MLELM–AE
Hidden layers	Single	Multiple	Multiple
Training	Non-iterative (closed form)	Backpropagation	ELM with backpropagation
Speed	Fast	Moderate	Fast and adaptive
Output update	Only output layer	All layers	Output and hidden layers
Loss function	Classification loss	Reconstruction loss	Reconstruction loss (MSE)
Adaptability	Low	High	High
Learning	Randomized and no tuning	Gradient-based	Hybrid: Random initialization and gradient tuning

Abbreviations: AE: Autoencoder; ELM: Extreme learning machine; MLELM: Multilayer ELM; MSE: Mean squared error.

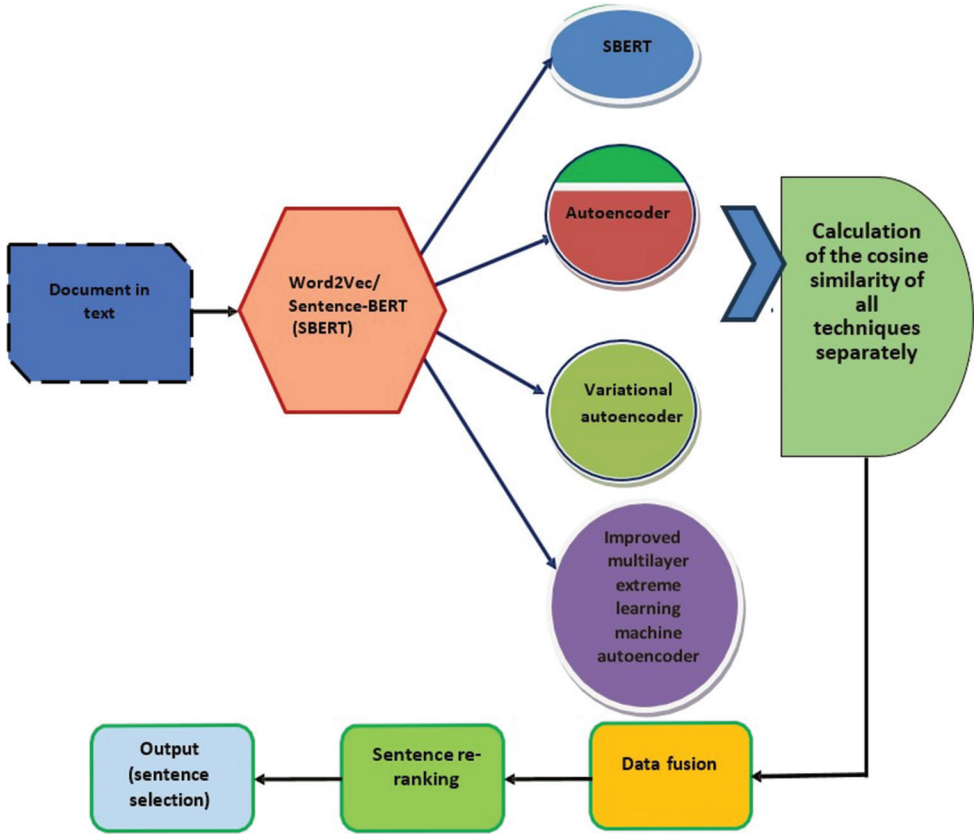

Fig. 1. Ensemble learning framework for text summarization

extracts sentence-level features independently, focusing on different aspects of sentence semantics and information compression. SBERT retains rich contextual information and deep contextual features, while AE and VAE reduce dimensionality and gather latent semantic structures. The improved MLELM–AE leverages the computational efficiency of extreme learning alongside the representational strength of deep learning models to enhance feature abstraction. Once sentence-level embeddings are formed, cosine similarity is computed separately for each model to assess sentence significance. The similarity scores are then integrated using a data fusion method, allowing the integration of diverse model perspectives. Based on the fused scores, sentences are re-ranked to prioritize informative and non-redundant content. Finally, the highest-ranked sentences are selected to form the extractive summary. This ensemble-based framework improves summarization effectiveness, semantic quality, and robustness by integrating the diverse capabilities of various encoding techniques.

4. Results and Discussion

4.1. Software Requirements

The proposed framework was implemented in PYTHON, a widely used general-purpose and high-level programming language that is primarily developed for emphasizing code readability. The

syntax of PYTHON permits developers to define concepts in fewer lines of code. Similarly, PYTHON effectively incorporates the system and works faster. PYTHON is used in numerous applications, including artificial intelligence, scientific computing, and automation. In addition, for many common tasks, the comprehensive standard library of PYTHON provides modules and functions.

4.2. Hardware Requirements

The hardware necessities for the proposed model and framework are as follows:

- Processor: Intel Core i5/i7
- Central processing unit speed: 3.20 GHz
- Operating system: Windows 10
- System type: 64-bit
- RAM: 4 GB

4.3. Dataset Description

The proposed improved MLELM–AE model was evaluated using the DUC 2002 dataset, which is publicly available (https://ieee-dataport.org/documents/sentence-embeddings-document-sets-duc-2002-summarization-task). The DUC 2002 dataset consists of 1,358 text documents. For experimentation, the dataset was divided into training, validation, and testing subsets. Specifically, 70% of the documents (950) were used for training, 10% (135) for validation, and the remaining 20% (271) for testing. The detailed hyperparameters employed in the proposed framework are presented in Table 2.

Table 2. Detailed hyperparameters of the models

Specifications	Proposed improved MELM-AE	AE	VAE	SBERT
Epoch	500	500	500	500
Activation function	ReLU	ReLU	ReLU	ReLU
Weight initialization	Hyperfan-In	Xavier	Xavier	Xavier
Learning rate	0.0001	0.008	0.017	0.124
Batch size	100	80	60	20
Optimizer	Adam	Adam	Adam	Adam
Loss function	MSE	MSE	MSE	MSE
Dropout rate%	0.2	0.5	0.4	0.3

Abbreviations: AE: Autoencoder; MLELM: Multilayer extreme learning machine; MSE: Mean squared error; ReLU: Rectified linear unit; SBERT: Sentence bidirectional encoder representations from transformers; VAE: Variational autoencoder.

4.4. Performance Evaluation of the Proposed Improved MLELM–AE model

The performance of the proposed improved MLELM–AE model was compared with existing techniques, including AE, SBERT, and VAE, to demonstrate its reliability. The evaluation was conducted using standard metrics, such as accuracy, precision, recall, F-measure, sensitivity, and ROUGE-1 score. The proposed improved MLELM–AE achieved superior results, with accuracy, precision, recall, F-measure, sensitivity, and ROUGE-1 scores of 96.32%, 97.16%, 96.01%, 97.24%, 97.01%, and 0.865145, respectively. In contrast, the existing techniques attained comparatively lower average performance across these metrics, as summarized in Table 3. These results confirm that the proposed improved MLELM–AE model significantly outperforms the baseline models in extractive text summarization.

Specifically, the highest accuracy of 96.32% was achieved by the proposed improved MLELM–AE model, significantly outperforming AE (91.41%), SBERT (90.87%), and VAE (91.48%), thereby confirming its robust ability to correctly identify relevant instances. In terms of precision (97.16%) and F-measure (97.24%), the proposed model exhibited exceptional performance, indicating its ability to generate highly accurate summaries or predictions with minimal false positives and a robust balance between precision and recall. Similarly, the high recall score (96.01%) highlights its effectiveness in capturing the majority of relevant outputs, ensuring comprehensive coverage of the target content. In contrast, AE, SBERT, and VAE recorded lower recall values of 91.03%, 91.11%, and 92.49%, respectively, highlighting their limitations in capturing all relevant elements.

The proposed improved MLELM–AE model also attained an outstanding ROUGE-1 score of 0.865145, a significant measure in text summarization that assesses unigram overlap between system-generated and reference summaries. This outperformed AE (0.819125), SBERT (0.805981), and VAE (0.816013), confirming that the summaries produced by the proposed model are more semantically and lexically aligned with human-authored summaries.

Moreover, the execution time of the improved MLELM–AE (50,015 ms) was shorter than that of AE (56,236 ms), SBERT (61,008 ms), and VAE (63,018 ms), demonstrating efficiency without compromising performance (Fig. 2). Finally, the model achieved a low error rate (0.010766), reflecting its accuracy in fitting training data; nonetheless, further assessment on unseen datasets is required to meticulously validate its generalization capability.

Overall, the experimental results indicate that the proposed improved MLELM–AE model not only

Table 3. Comparative assessment of the models

Model	Accuracy %	Precision %	Recall %	F-Measure %	ROUGE-1	Time (ms)	Error
Proposed improved MLELM–AE	96.32	97.16	96.01	97.24	0.905145	50,015	0.010766
AE	91.41	93.21	91.03	93.12	0.819125	56,236	0.031064
SBERT	90.87	92.47	91.11	93.52	0.805981	61,008	0.066596
VAE	91.48	93.52	92.49	94.01	0.816013	63,018	0.092872

Abbreviations: AE: Autoencoder; MLELM: Multilayer extreme learning machine; ROUGE-1: Recall-oriented understudy for gisting evaluation 1; SBERT: Sentence bidirectional encoder representations from transformers; VAE: Variational autoencoder.

Table 4. Comparative analysis with previously described frameworks

References	Techniques	ROUGE-1 score
Proposed ensemble framework in the present study	AE, SBERT, VAE, and improved MLELM–AE	0.865145
Hernández-Castañeda et al. (2022)	GA	0.414000
Hernández-Castañeda et al. (2020)	GA, LDA, and TF–IDF	0.486810

Abbreviations: AE: Autoencoder; GA: Genetic algorithm; LDA: Latent Dirichlet allocation; MLELM: Multilayer extreme learning machine; ROUGE-1: Recall-Oriented Understudy for Gisting Evaluation 1; TF–IDF: Term frequency–inverse document frequency.

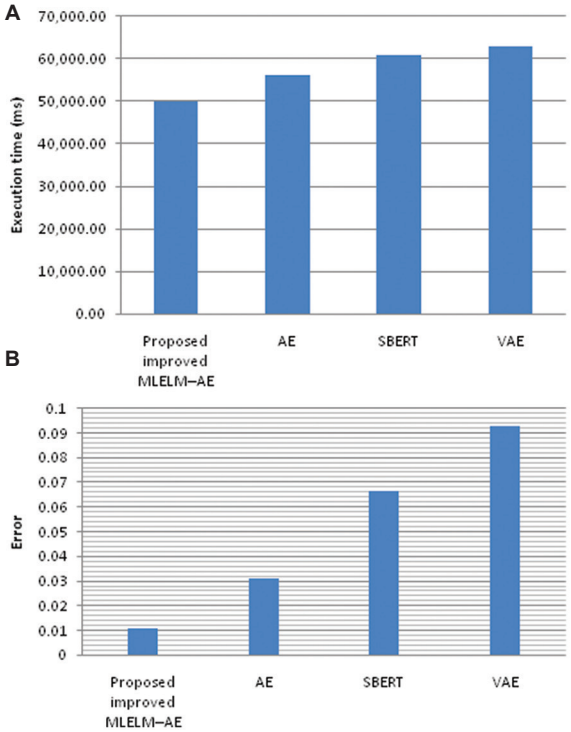

Fig. 2. Execution time (A) and error (B) of the models Abbreviations: AE: Autoencoder; MLELM: Multilayer extreme learning machine; SBERT: Sentence bidirectional encoder representations from transformers; VAE: Variational autoencoder

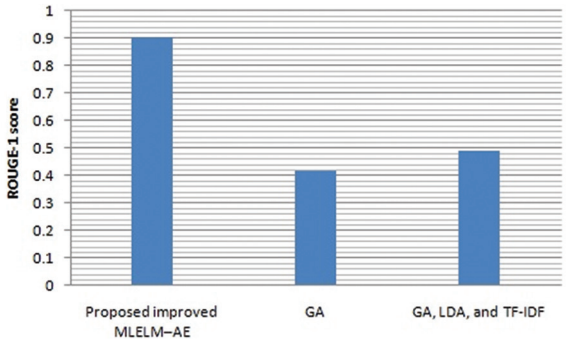

Fig. 3. Comparative analysis of the proposed ensemble framework and previously described models Abbreviations: AE: Autoencoder; GA: Genetic algorithm; LDA: Latent Dirichlet allocation; MLELM: Multilayer extreme learning machine; ROUGE-1: Recall-Oriented Understudy for Gisting Evaluation 1; TF–IDF: Term frequency–inverse document frequency

attains state-of-the-art accuracy and performance metrics but also offers computational proficiency, making it a promising approach for real-world applications in text summarization and related NLP tasks.

4.5. Comparative Analysis of the Proposed Ensemble Framework

A comparative analysis of the proposed ensemble framework and previously described frameworks (Hernández-Castañeda et al., 2020; Hernández-Castañeda et al., 2022) was conducted to further validate the model's reliability. The results are summarized in Table 4 and Fig. 3. The proposed ensemble framework achieved a notably high ROUGE-1 score of 0.865145, primarily due to the incorporation of the improved MLELM–AE model. In contrast, the existing genetic algorithm approach achieved a considerably lower ROUGE-1 score of 0.414 on the same DUC 2002 dataset. Similarly, the

model based on a genetic algorithm, latent Dirichlet allocation, and TF–IDF techniques attained a lower ROUGE-1 score of 0.48681, which can be attributed to their computational complexity. These findings clearly demonstrate that the proposed ensemble framework outperforms traditional approaches in performing ATS.

5. Conclusion

ATS is a widely explored research area in the NLP community, as it enables the generation of concise and informative summaries from large volumes of text. This paper presents an improved ensemble learning-based ATS framework that incorporates the AE, SBERT, VAE, and improved MLELM–AE. The DUC 2002 dataset was employed for training and evaluation. The research methodology involves several steps, including pre-processing, slang identification and filtering, part-of-speech tagging, entity extraction, vectorization, ensemble modeling, similarity evaluation, re-ranking, and optimal sentence selection. Experimental results demonstrate that the proposed improved MLELM–AE achieved high accuracy (96.32%), precision (97.16%), and recall (96.01%). On the other hand, the proposed ensemble framework achieved a high ROUGE-1 score of 0.865145, significantly outperforming existing models. These findings clearly validate the effectiveness of the proposed approaches in delivering improved ATS performance.

Acknowledgments

None.

Funding

None.

Conflict of Interest

The authors declare that they have no competing interests.

Author Contributions

Conceptualization: Sunil Upadhyay
Investigation: Sunil Upadhyay
Writing–original draft: Sunil Upadhyay
Writing–review and editing: All authors

Availability of Data

Data sharing is not applicable to this article as no datasets were generated or analyzed during the present study.

References

Adelia, R., Suyanto, S., & Wisesty, U.N. (2019). Indonesian abstractive text summarization using bidirectional gated recurrent unit. *Procedia Computer Science*, 157, 581–588. https://doi.org/10.1016/j.procs.2019.09.017

Akter, S., Asa, A.S., Uddin, M.P., Hossain, M.D., Roy, S.K., & Afjal, M.I. (2017). An Extractive Text Summarization Technique for Bengali Document(s) using K-Means Clustering Algorithm. In: *Proceedings of the 2017 IEEE International Conference on Imaging, Vision and Pattern Recognition (icIVPR)*. IEEE, p1–6. https://doi.org/10.1109/ICIVPR.2017.7890883

Alami, N., Meknassi, M., & En-nahnahi, N. (2019). Enhancing unsupervised neural networks based text summarization with word embedding and ensemble learning. *Expert Systems with Applications*, 123, 195–211. https://doi.org/10.1016/j.eswa.2019.01.037

Alotaibi, A., & Nadeem, F. (2025). An unsupervised integrated framework for Arabic aspect-based sentiment analysis and abstractive text summarization of traffic services using transformer models. *Smart Cities*, 8(2), 62. https://doi.org/10.3390/smartcities8020062

Alzuhair, A., & Al-Dhelaan, M. (2019). An approach for combining multiple weighting schemes and ranking methods in graph-based multi-document summarization. *IEEE Access*, 7, 120375–120386. https://doi.org/10.1109/access.2019.2936832

Dave, H., & Jaswal, S. (2015). Multiple Text document summarization system using hybrid summarization technique. In: *Proceedings of the 2015 1st International Conference on Next Generation Computing Technologies (NGCT)*, IEEE, p804–808. https://doi.org/10.1109/ngct.2015.7375231

Dilawari, A., Khan, M.U.G., Saleem, S., Zahoor-Ur-Rehman, & Shaikh, F.S. (2023). Neural attention model for abstractive text summarization using linguistic feature space. *IEEE Access*, 11, 23557–23564. https://doi.org/10.1109/access.2023.3249783

Ding, J., Li, Y., Ni, H., & Yang, Z. (2020). Generative text summary based on enhanced semantic attention and gain-benefit gate. *IEEE Access*, 8, 92659–92668. https://doi.org/10.1109/access.2020.2994092

Du, Y., & Huo, H. (2020). News text summarization based on multi-feature and fuzzy logic. *IEEE Access*, 8, 140261–140272. https://doi.org/10.1109/access.2020.3007763

Elbarougy, R., Behery, G., & El Khatib, A. (2020). Extractive Arabic text summarization using modified PageRank algorithm. *Egyptian*

Informatics Journal, 21(2), 73–81. https://doi.org/10.1016/j.eij.2019.11.001

El-Kassas, W.S., Salama, C.R., Rafea, A.A., & Mohamed, H.K. (2020). EdgeSumm: Graph-based framework for automatic text summarization. *Information Processing and Management*, 57(6), 102264. https://doi.org/10.1016/j.ipm.2020.102264

Gambhir, M., & Gupta, V. (2017). Recent automatic text summarization techniques: A survey. *Artificial Intelligence Review*, 47(1), 1–66. https://doi.org/10.1007/s10462-016-9475-9

Hassan, A.Q.A., Al-Onazi, B.B., Maashi, M., Darem, A.A., Abunadi, I., & Mahmud, A. (2024). Enhancing extractive text summarization using natural language processing with an optimal deep learning model. *AIMS Mathematics*, 9(5), 12588–12609. https://doi.org/10.3934/math.2024616

Hernández-Castañeda, Á., García-Hernández, R.A., & Ledeneva, Y. (2023). Toward the automatic generation of an objective function for extractive text summarization. *IEEE Access*, 11, 51455–51464. https://doi.org/10.1109/access.2023.3279101

Hernández-Castañeda, Á., García-Hernández, R.A., Ledeneva, Y., & Millán-Hernández, C.E. (2020). Extractive automatic text summarization based on lexical-semantic keywords. *IEEE Access*, 8, 49896–49907. https://doi.org/10.1109/access.2020.2980226

Hernández-Castañeda, Á., García-Hernández, R.A., Ledeneva, Y., & Millán-Hernández, C.E. (2022). Language-independent extractive automatic text summarization based on automatic keyword extraction. *Computer Speech and Language*, 71, 101256. https://doi.org/10.1016/j.csl.2021.101267

Jadhav, A., Jain, R., Fernandes, S., & Shaikh, S. (2019). Text Summarization using Neural Networks. In *2019 6th IEEE International Conference on Advances in Computing, Communication and Control (ICAC3)*. https://doi.org/10.1109/icac347590.2019.9036739

Jain, A., Bhatia, D., & Thakur, M.K. (2017). Extractive text summarization using word vector embedding. In: *2017 International Conference on Machine Learning and Data Science (MLDS)*, p51–55. https://doi.org/10.1109/mlds.2017.12

Jindal, S.G., & Kaur, A. (2020). Automatic keyword and sentence-based text summarization for software bug reports. *IEEE Access,* 8, 65352–65370. https://doi.org/10.1109/access.2020.2985222

Joshi, A., Fidalgo, E., Alegre, E., & Fernández-Robles, L. (2019). SummCoder: An unsupervised framework for extractive text summarization based on deep auto-encoders. *Expert Systems with Applications*, 129, 200–215. https://doi.org/10.1016/j.eswa.2019.03.045

Khan, B., Usman, M., Khan, I., Khan, J., Hussain, D., & Gu, Y.H. (2025). Next-generation text summarization: A T5-LSTM FusionNet hybrid approach for psychological data. *IEEE Access*, 13, 1–15.

Liang, Z., Du, J., & Li, C. (2020). Abstractive social media text summarization using selective reinforced Seq2Seq attention model. *Neurocomputing*, 410, 432–440. https://doi.org/10.1016/j.neucom.2020.04.137

Madhuri, J.N., & Kumar, R.G. (2019). Extractive Text Summarization using Sentence Ranking. In: *Proceedings of the 2019 International Conference on Data Science and Communication (IconDSC)*, p1–3. https://doi.org/10.1109/icondsc.2019.8817040

Mitra, M., Buckley, C., & Research, S. (2000). *Automatic Text Summarization by Paragraph Extraction*. Available from: https://www.researchgate.net/publication/2457789 [Last accessed on 2025 Sep 09].

Moradi, M., Dorffner, G., & Samwald, M. (2020). Deep contextualized embeddings for quantifying the informative content in biomedical text summarization. *Computer Methods and Programs in Biomedicine*, 184, 105117. https://doi.org/10.1016/j.cmpb.2019.105117

Onan, A., & Alhumyani, H. (2024b). Contextual hypergraph networks for enhanced extractive summarization: Introducing multi-element contextual hypergraph extractive summarizer (MCHES). *Applied Sciences*, 14(11), 4671. https://doi.org/10.3390/app14114671

Onan, A., & Alhumyani, H.A. (2024a). FuzzyTP-BERT: Enhancing extractive text summarization with fuzzy topic modeling and transformer networks. *Journal of King Saud University - Computer and Information Sciences*, 36(6), 102080. https://doi.org/10.1016/j.jksuci.2024.102080

Pradip, K.G., & Patil, D.R. (2016). Summarization of Sentences using Fuzzy and Hierarchical Clustering Approach. In: *2016 Symposium on Colossal Data Analysis and Networking (CDAN)*. IEEE. p1–7. https://doi.org/10.1109/cdan.2016.7570907

Prameswari, P., Zulkarnain, Z, Surjandari, I., & Laoh, E. (2018). *Mining Online Reviews in Indonesia's Priority Tourist Destinations using Sentiment Analysis and Text Summarization Approach*. In: 2017 IEEE 8th International Conference on Awareness Science and Technology (iCAST). IEEE, p36–41.

https://doi.org/10.1109/icawst.2017.8256540

Qaroush, A., Abu Farha, I., Ghanem, W., Washaha, M., & Maali, E. (2021). An efficient single document Arabic text summarization using a combination of statistical and semantic features. *Journal of King Saud University - Computer and Information Sciences*, 33(6), 677–692. https://doi.org/10.1016/j.jksuci.2019.03.010

Sanchez-Gomez, J.M., Vega-Rodríguez, M.A., & Pérez, C.J. (2019). Parallelizing a multi-objective optimization approach for extractive multi-document text summarization. *Journal of Parallel and Distributed Computing*, 134, 166–179. https://doi.org/10.1016/j.jpdc.2019.09.001

Toprak, A., & Turan, M. (2025). Enhanced automatic abstractive document summarization using transformers and sentence grouping. *The Journal of Supercomputing*, 81(4), 1–30. https://doi.org/10.1007/s11227-025-07048-6

Tsai, C.F., Chen, K., Hu, Y.H., & Chen, W.K. (2020). Improving text summarization of online hotel reviews with review helpfulness and sentiment. *Tourism Management*, 80, 104122. https://doi.org/10.1016/j.tourman.2020.104122

Van Lierde, H., & Chow, T.W.S. (2019). Learning with fuzzy hypergraphs: A topical approach to query-oriented text summarization. *Information Sciences*, 496, 212–224. https://doi.org/10.1016/j.ins.2019.05.020

Wan, L. (2018). Extraction Algorithm of English Text Summarization for English Teaching. In: *2018 3rd International Conference on Intelligent Transportation, Big Data and Smart City (ICITBS)*. https://doi.org/10.1109/icitbs.2018.00085

Yao, K., Zhang, L., Du, D., Luo, T., Tao, L., & Wu, Y. (2018). Dual encoding for abstractive text summarization. *IEEE Transactions on Cybernetics*. IEEE, New York. https://doi.org/10.1109/tcyb.2018.2876317

Zajic, D.M., Dorr, B.J., & Lin, J. (2008). Single-document and multi-document summarization techniques for email threads using sentence compression. *Information Processing and Management*, 44(4), 1600–1610. https://doi.org/10.1016/j.ipm.2007.09.007

Zenkert, J., Klahold, A., & Fathi, M. (2018). Towards Extractive Text Summarization using Multidimensional Knowledge Representation. In: *2018 IEEE International Conference on Electro/ Information Technology (EIT)*, p826–831. https://doi.org/10.1109/EIT.2018.8500186

AUTHOR BIOGRAPHIES

Mr. Sunil Upadhyay received his M.TECH. degree from the Department of Computer Science and Engineering at RGEC Meerut, Gautam Buddha Technical University, formerly known as UPTU, Lucknow, U.P., India, in 2012. He is pursuing his Ph.D. degree in Computer Science and Engineering from Amity University, Madhya Pradesh, Gwalior, India. His presentsf research interest includes NLP, data analytics, artificial intelligence, and machine learning.

Prof. (Dr.) Hemant Kumar Soni completed his M. Tech. (IT) at Bundelkhand University, Jhansi, Uttar Pradesh, India, and a Doctoral degree in Computer Science and Engineering from Amity University, Madhya Pradesh, Gwalior, India. He has 26 years of teaching experience for undergraduate and post-graduate courses in Computer Science, and is presently working as a Professor in the Department of Computer Science and Engineering at Amity University Madhya Pradesh, Gwalior, India. His research interests are in natural language processing, data science, machine learning, and data mining. He published many research papers in Web of Science, SCI, and Scopus-indexed journals. His research articles received high levels of citations in Scopus and Google Scholar. He is a Reviewer of many referred journals, including *IEEE Transactions*.

DOI: 10.6977/IJoSI.202510_9(5).0002
Z.R. Zhang & Y.W. Chan/Int. J. Systematic Innovation, 9(5), 14-22 (2025)

Exploring satisfaction with military catering services using the service quality model and importance-performance analysis

Zu-Rong Zhang[1], Ya-Wen Chan[2]*

[1]Logistics Division, Air Force Meteorological Squadron, Taipei, Taiwan, Republic of China

[2]Department of Health and Creative Plant-Based Food Industries, Fo Guang University, Yilan, Taiwan, Republic of China

*Corresponding author E-mail: ywchan@mail.fgu.edu.tw

(Received 27 March, 2025; Final version received 28 July, 2025; Accepted 15 August, 2025)

Abstract

The importance of military catering in military organizations cannot be overlooked, as it not only impacts the health and physical fitness of service members but also directly affects combat readiness and morale. This study focuses on a northern air force base, using the Parasuraman-Zeithaml-Berry service quality (SERVQUAL) model's Gap 1 and Gap 5 as its framework. The aim is to investigate the perception gaps in catering service quality between food service providers and customer. An importance-performance analysis matrix is employed to further analyze the findings. The analysis reveals that, regarding "catering service quality," food service providers who are actively serving without formal food service certification, and those with high school or college education, tend to place more emphasis on tangibility, reliability, empathy, and responsiveness. For service quality expectations, customers who possess a college education and have obtained a food service certification show higher expectations in tangibility and reliability dimensions. Younger customer, aged 18–25, who are uncertified and less experienced, report greater satisfaction with the catering service's reliability, responsiveness, and assurance dimensions after their experience with the base's services. Regarding the perception difference in Gap 1 of the SERVQUAL model, the study suggests that services should prioritize user experience and ensure transparency by publicizing findings from meal review meetings. Feedback can be gathered through a satisfaction mailbox to address and efficiently amend any service deficiencies. For Gap 5 in terms of experience, customers show particular concern for food safety measures and overall service quality, indicating that these areas should be maintained or enhanced. Regular training is recommended to improve the knowledge and effectiveness of food service providers in these critical aspects.

Keywords: Group Catering, Importance-Performance Analysis Matrix, Service Quality Model, User Satisfaction

1. Introduction

Military catering services not only fulfill basic nutritional needs but also play a critical role in supporting military operations and assurance readiness. Conducting academic research on the service quality of military catering can facilitate management and operational optimization, thus enhancing overall combat effectiveness and the well-being of military personnel. This study is based on the service quality model (SERVQUAL) and its scale proposed by Parasuraman et al. (1988). It targets "food service providers" and "customer" at an Air Force base in northern Taiwan, distributing questionnaires to investigate perceived differences in service quality during meal times and aiming to minimize latent risks in military catering services.

The objectives of this research are threefold: to examine the perception gap in service quality between "food service providers" and the "customer" (Gap 1); to explore the perception gap between "customer' expectations" and their "actual experiences" with service quality in Air Force catering services (Gap 5); and to propose actionable improvement strategies for both gaps. The findings of this study are intended to serve as a strategic reference for military units in enhancing catering service quality in the future.

2. Literature Review

2.1. Group Catering

Morgan (2004) defines group catering as a systematic approach to meal management that enables coordinated food service operations to produce meals that maximize customer satisfaction while ensuring reasonable profitability for the catering organization. Examples include self-service buffet arrangements, which minimize labor requirements and provide large quantities of dishes within a short time to satisfy the dining needs of many people.

2.2. User Satisfaction

Customer satisfaction, also known as "CS," refers to the alignment of a customer's expectations with their perception of having those needs met. Cardozo (1965) suggests that customer satisfaction increases the likelihood of repeat purchases and can further influence the willingness to buy other products. Scholars Czepiel et al. (1974) argue that the degree of customer satisfaction can be seen as an overall evaluative response within the service process, representing a composite of subjective reactions to various product attributes (Oliver, 1981). Furthermore, Rosenzweig and Singh (1991) emphasize that "customer satisfaction" should be measured individually across the performance of each attribute of a product, with these individual scores aggregated to produce an overall satisfaction measure. In summary, both customer satisfaction and overall satisfaction vary depending on the industry and the specific research subjects.

2.3. Service Quality Model and Service Quality

The SERVQUAL defines service quality based on the customer's experience throughout the service process. Wyckoff (1984) suggests that service quality is achieved by meeting the immediate needs of the customer, a perspective closely tied to the existing brand image (Sasser, Olsen, & Wyckoff, 1978). In contrast, Gronroos (1982) posits that service quality is determined by comparing the consumer's "expectations" with their "actual experiences." Lehtinen and Lehtinen (1982) further conceptualize service quality across three dimensions, interaction, tangibility, and communality, arguing that service quality should be evaluated from the customer's perspective. According to their view, the quality valued by customers is derived from both the service process and the outcome.

The SERVQUAL utilized in this study is based on the SERVQUAL scale, developed by the scholars in 1988, for measuring service quality. A brief overview is provided below:

(i) Tangibility
In the service process, tangible aspects emphasize the actual service experience, encompassing all physical elements or sensations encountered during dining. This includes the environment, equipment, facilities, staff, decor, scent, hygiene, and even the attitude and demeanor of personnel in delivering service to customers (Kazarian, 1983).

(ii) Reliability
Reliability reflects the customer's expectation beyond simply satisfying hunger; it includes the desire for dependable food, service, facilities, environment, safety, hygiene, and everything pertinent to the customer's dining experience.

(iii) Assurance
Assurance complements tangibility, signifying the politeness and respect service staff demonstrate toward customers while providing food or services. It builds trust and confidence in the service staff's overall performance, thus contributing to customer satisfaction.

(iv) Responsiveness
Unforeseen incidents and even disasters are unpredictable. Through training, service staff can enhance their responsiveness and learn to appropriately assist customers when problems or mishandlings arise. Effective remediation can even encourage customer loyalty and increase the likelihood of repeat visits.

(v) Empathy
According to Maslow's hierarchy (Maslow, 1943) of needs, the need for esteem is reflected here, where customers seek respectful treatment from service staff. Empathy focuses on delivering personalized attention and the most suitable service, ensuring a satisfying dining experience for customers (Maslow, 1943).

In 1985, Parasuraman et al. at Cambridge University developed the SERVQUAL. This model emphasizes the core idea that "the customer is the determinant of service quality." Within this service quality framework, there are five gaps, each highlighting critical areas that must be addressed to ensure customer satisfaction with the service. The model suggests that bridging these five service quality gaps is essential to achieving customer satisfaction (Fig. 1).

3. Research Methodology

3.1. Data Collection and Analysis Method

This study adopts the SERVQUAL as its research methodology and utilizes the SERVQUAL scale to develop a service quality satisfaction questionnaire.

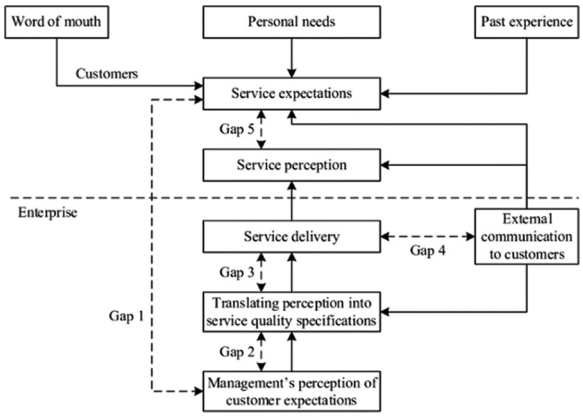

Fig. 1. Service quality model (SERVQUAL)
Source: Blackett (1988); Parasuraman et al. (1985)

The questionnaire targets "food service providers" and "customer" at an Air Force base in northern Taiwan. The study focuses on Gaps 1 and 5 of the SERVQUAL service quality model as the basis for questionnaire items, and the design incorporates the five dimensions from the revised SERVQUAL scale.

"Food service providers" refers to those responsible for menu design, calculating the number of diners, procuring ingredients, and organizing and preparing meals within the base. These personnel may include externally hired chefs or in-house mess staff. "Customer" includes both military and civilian personnel at the base who utilize group catering services. In this study, the term refers specifically to catering service managers and operators, including those with responsibilities for planning, oversight, and execution.

A single structured questionnaire was employed in this study, comprising three sections: the first section collected respondents' demographic information; the second section assessed the service quality of institutional catering services; and the third section evaluated overall user satisfaction with the group meals. All three sections adopted consistent item designs and utilized a five-point Likert scale for measurement, thereby ensuring comparability across constructs. This design allowed the researchers to derive both Gap 1 and Gap 5 using a single questionnaire instrument.

3.2. Measurement Tools

The research framework is structured as follows:

- Gap 1: The difference between "catering managers' perception of customer' expectations" and "customer' expectations of catering service quality"
- Gap 5: The difference between "customer' expectations of catering service quality" and "customer' experience with catering service quality" (Fig. 2).

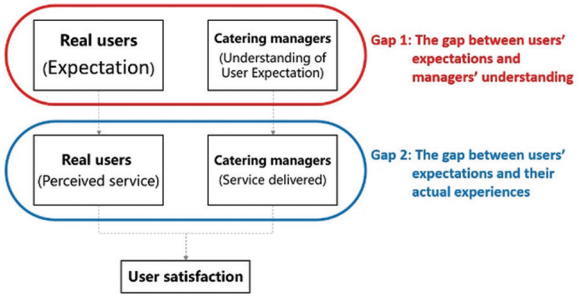

Fig. 2. Research framework diagram

4. Data Analysis and Results

A total of 460 valid questionnaires were collected in this study, distributed among "food service providers" and "customer." The detailed analysis is as follows:

For the food service provider's dimension, 170 valid questionnaires were collected. Among the respondents, 52% were male and 48% female. Most respondents were non-military staff (39%), followed by volunteer service members (30%), with active duty and reserve duty each accounting for 11%, and conscripts at 9%. In addition, 61% were military personnel, while 39% were in-house contracted staff.

In the customer dimension, 290 valid questionnaires were obtained. Demographic analysis showed a majority of male respondents (65%) compared to female respondents (35%). The majority were reserve duty members (68%), followed by conscripts (16%), volunteer service members (10%), active duty (4%), and non-military staff (2%).

4.1. Reliability and Validity Analysis

For the formal questionnaire, the Cronbach's alpha values were as follows: 0.94 for "food service providers," 0.95 for "customer' expectations," and 0.96 for "customer' actual experiences," indicating a high level of reliability. Regarding validity, the Kaiser-Meyer-Olkin (KMO) and Bartlett's test of sphericity for the six dimensions—tangibility, reliability, assurance, responsiveness, empathy, and overall satisfaction—were 0.83, 0.81, 0.74, 0.80, 0.64, and 0.84, respectively. Although the KMO for the empathy dimension was 0.64 (slightly below the 0.7 threshold), it was within the acceptable range and therefore retained. All other dimensions had KMO values above 0.7, indicating good validity of the questionnaire. All Bartlett's tests of sphericity were statistically significant at $p<0.001$, confirming the suitability of the data for factor analysis.

4.2. Reliability and Validity Analysis

4.2.1. Reliability Analysis

After pilot testing and item screening, the internal consistency of each questionnaire was examined.

The overall Cronbach's alpha coefficients were as follows: 0.94 for the "institutional catering staff" scale, 0.95 for the "customer' expectations" scale, and 0.96 for the "customer' perceived experience" scale.

(i) Reliability of the institutional catering staff scale:
Tangibles (6 items): Cronbach's $\alpha = 0.82$
Reliability (6 items): Cronbach's $\alpha = 0.82$
Assurance (4 items): Cronbach's $\alpha = 0.77$
Responsiveness (5 items): Cronbach's $\alpha = 0.85$
Empathy (3 items): Cronbach's $\alpha = 0.65$
Overall satisfaction (5 items): Cronbach's $\alpha = 0.85$.

Although the alpha coefficient for the "Empathy" dimension was slightly below the commonly accepted threshold of 0.70, it was retained as it remains within the marginally acceptable range. All other dimensions showed acceptable reliability, indicating that the questionnaire demonstrates strong internal consistency.

(ii) Reliability of the customer' expectations scale:
Tangibles (6 items): Cronbach's $\alpha = 0.89$
Reliability (6 items): Cronbach's $\alpha = 0.93$
Assurance (4 items): Cronbach's $\alpha = 0.90$
Responsiveness (5 items): Cronbach's $\alpha = 0.86$
Empathy (3 items): Cronbach's $\alpha = 0.86$
Overall satisfaction (5 items): Cronbach's $\alpha = 0.89$

All dimensions achieved alpha values exceeding 0.70, indicating a high degree of internal reliability.

(iii) Reliability of the customer' perceived experience scale:
Tangibles (6 items): Cronbach's $\alpha = 0.92$
Reliability (6 items): Cronbach's $\alpha = 0.94$
Assurance (4 items): Cronbach's $\alpha = 0.90$
Responsiveness (5 items): Cronbach's $\alpha = 0.92$
Empathy (3 items): Cronbach's $\alpha = 0.85$
Overall satisfaction (5 items): Cronbach's $\alpha = 0.92$

All dimensions yielded Cronbach's alpha values above the 0.70 threshold, confirming the questionnaire's reliability.

4.2.2. Factor Analysis

This section presents the KMO values and Bartlett's test of sphericity results for each dimension.

(i) Factor analysis of the institutional catering staff scale:
Tangibles (6 items): KMO = 0.83
Reliability (6 items): KMO = 0.81
Assurance (4 items): KMO = 0.74
Responsiveness (5 items): KMO = 0.80
Empathy (3 items): KMO = 0.64
Overall satisfaction (5 items): KMO = 0.84

Although the KMO value for the "Empathy" dimension was slightly below the 0.70 threshold, it was considered marginally acceptable and thus retained. All other dimensions reported KMO values above 0.70, indicating sampling adequacy and supporting the suitability of the data for factor analysis.

4.3. Correlation Analysis

In this study, the *p*-value between customer' expectations and actual experiences was 0.000 for all dimensions, with Pearson correlation coefficients all below 0.01, indicating a moderate positive correlation across the dimensions. This result confirms a correlation between the expectations and experiences of customer. It substantiates the hypothesis that a service gap exists between food service, users' service quality, and customer' expectations, as well as a gap between customer' expectations of catering service quality and their satisfaction after the experience.

4.4. Importance-Performance Analysis (IPA) Matrix

To further understand the differences between the two gaps, this study employs the IPA matrix as an analytical tool (Martilla & James, 1977).

Gap 1: The gap between "catering staff"'s perception of customer' expectations" and "customer' expectations of catering service quality" (Fig. 3).

(i) Quadrant I: Keep up the good work (high expectation and high satisfaction).
- Tangibility:

Item 2: Food service provider's attire is clean and orderly.
Item 3: Dining environment and hygiene quality are good.
Item 4: Provided meals adhere to refrigeration at 7°C and freezing at −18°C, with measures to prevent cross-contamination risks.
Item 5: Hot dishes meet the standard core temperature of above 60°C.
Item 8: Meals are provided on time.

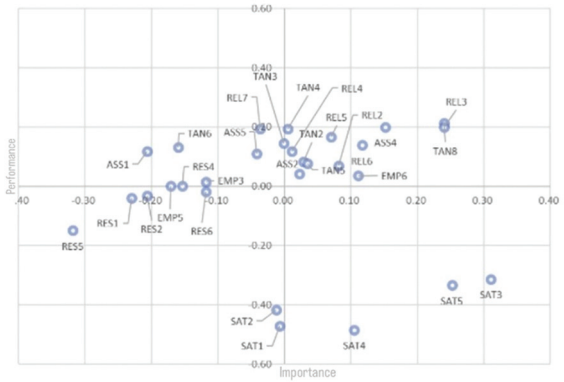

Fig. 3. Analysis matrix for "Gap 1"
Abbreviations: ASS: Assurance; EMP: Empathy; REL: Reliability; RES: Responsiveness; SAT: Overall satisfaction; TAN: Tangibility. Source: Compiled by this study

- Reliability:
  Item 2: food service providers have obtained relevant food service certifications.
  Item 3: Adequate preventive measures are in place under pandemic conditions, such as weekly disinfection and environmental sanitation per meal during outbreaks.
  Item 4: Catering following the nutritional balance in accordance with the base mission requirements.
  Item 5: The food is fresh.
  Item 6: Cleanliness of food containers and ingredients is well-maintained.
- Assurance:
  Item 2: Reliable services are provided.
  Item 4: Food is used within its expiration date.
- Empathy:
  Item 6: Clear and accessible complaint channels for catering services.

(ii) Quadrant II: Overly effortful (low expectation and high satisfaction).

- Overall Satisfaction:
  Item 3: Overall food portion is adequate.
  Item 4: Satisfaction with the overall taste of food.
  Item 5: Good variety in food selection.

(iii) Quadrant III: Low-priority improvement (low expectation and low satisfaction).

- Responsiveness:
  Item 1: Food service providers do not ignore issues due to busyness.
  Item 2: Questions raised by users are answered accurately.
  Item 5: Food delivery personnel are quick, quiet, and precise.
  Item 6: Quality service is provided on the first attempt.
- Overall satisfaction:
  Item 1: Overall food quality is good.
  Item 2: Overall dining environment hygiene is satisfactory.

(iv) Quadrant IV: Concentrate here (high expectation and low satisfaction).

- Tangibility:
  Item 6: Food containers are structurally sound without cracks or damage.
- Reliability:
  Item 7: Effective oversight of daily potential food safety incidents.
- Assurance:
  Item 1: Actual dishes served are consistent with the menu.
  Item 5: Food service providers prioritize users' rights in food service.
- Responsiveness:
  Item 4: Issues raised are actively addressed by the catering unit.
- Empathy:
  Item 3: Routine review of catering errors.
  Item 5: The catering unit shows proactive concern for users.

Gap 5: The difference between "customer' expectations of catering service quality" and "customer' experience with catering service quality" (Fig. 4).

(i) Quadrant I: Keep up the good work (high expectation and high satisfaction).

- Tangibility:
  Item 2: food service provider's attire is clean and orderly.
  Item 4: Meals provided adhere to refrigeration standards of 7°C and freezing standards of −18°C, with measures to prevent cross-contamination.
  Item 5: Hot dishes maintain a core temperature standard of above 60°C.
- Reliability:
  Item 2: food service providers have obtained relevant food service certifications.
  Item 3: Adequate preventive measures, such as weekly disinfection of the dining area and daily sanitation during outbreaks.
  Item 4: Meals are nutritionally balanced according to base mission requirements.
  Item 5: The meal is fresh.
  Item 4: Cleanliness of food containers and ingredients is well-maintained.
- Assurance:
  Item 4: Food is used within its expiration date.
- Empathy:
  Item 3: Routine review of catering errors.

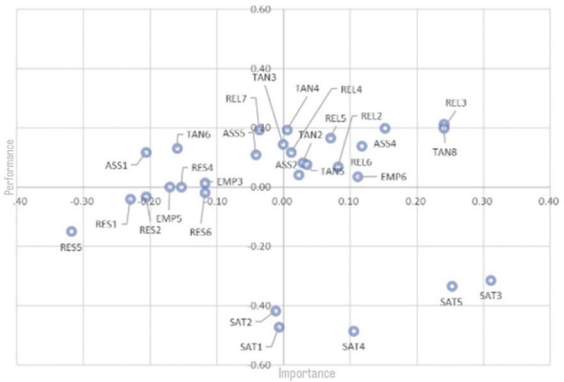

Fig. 4. Gap 5 analysis matrix
Abbreviations: ASS: Assurance; EMP: Empathy; REL: Reliability; RES: Responsiveness; SAT: Overall satisfaction; TAN: Tangibility. Data Source: Compiled by this study

Item 5: The catering unit shows proactive concern for users.

(ii) Quadrant II: Overly effortful (low expectation and high satisfaction).

- Tangibility:
  Item 3: Dining environment and hygiene quality are good.
  Item 6: Food containers are structurally sound without cracks or damage.
- Reliability:
  Item 7: Daily food safety incidents are managed accurately.
- Assurance:
  Item 1: Dishes served are consistent with the menu.
  Item 2: The service provided is reliable.
  Item 5: Food service providers prioritize users' rights in service.
- Responsiveness:
  Item 4: Issues raised by users are promptly addressed.
- Empathy:
  Item 6: Clear and accessible complaint channels for catering services.

(iii) Quadrant III: Low-priority improvement (low expectation and low satisfaction).

- Responsiveness:
  Item 1: Food service providers do not ignore issues due to busyness.
  Item 2: Questions raised by users are accurately answered.
  Item 5: Food delivery personnel are quick, quiet, and precise.
  Item 6: Quality service is provided on the first attempt.
- Overall satisfaction:
  Item 1: Overall food quality is good.
  Item 2: Overall dining environment hygiene is satisfactory.
  Item 4: Satisfaction with the overall taste of food.

(iv) Quadrant IV: Concentrate here (high expectation and low satisfaction).

- Overall satisfaction:
  Item 3: Adequate portion sizes for meals.
  Item 5: Good variety in food selection.

These areas in Quadrant IV should be prioritized for review and improvement to better align with user expectations.

In addition, it is recommended that future improvements incorporate intelligent menu design systems that leverage big data analytics to identify the preferences of customer. Such systems can provide personalized, seasonal, and nutritionally balanced meal options. Furthermore, the application of modern cooking techniques, such as sous vide, and the adoption of energy-efficient smart kitchen equipment may enhance both meal quality and operational efficiency.

From a management perspective, it is advisable to implement a participatory service improvement mechanism, such as regularly organizing user forums or conducting anonymous feedback surveys, to enhance user engagement. Menu planning should incorporate local culinary characteristics and seasonal ingredients to promote dietary diversity and health orientation. Moreover, offering customized options for special dietary needs—such as low-carbohydrate, plant-based, or gluten-free meals—may further improve overall dining satisfaction and user loyalty.

5. Conclusion and Recommendations

This chapter presents the findings in Section 1, followed by practical recommendations for military units in Section 2.

5.1. Research Findings

For the differences in service quality perception by hierarchical level, the study found significant differences in the perceived quality of catering services, specifically in the SERVQUAL dimension of tangibility, based on the hierarchical level of food service providers. Higher-ranking personnel demonstrated a stronger focus on tangible aspects, including food, service, facilities, safety, and hygiene. This suggests that military personnel are more attuned to and value tangible service quality compared to in-house contracted staff within the northern air force base. Specifically, "military personnel > in-house contracted staff" highlights that military personnel are more aware and concerned about the tangible aspects of catering service quality than their civilian counterparts.

In terms of the impact of certification on the perception of service quality reliability, significant differences were observed in the SERVQUAL reliability dimension based on whether the food service providers held food service certifications. Personnel without certification showed a greater concern for reliable, trustworthy services, implying a perception gap between certified and uncertified staff regarding service reliability. Specifically, "uncertified > certified" highlights that uncertified food service providers place more importance on reliability compared to certified personnel at the base.

Regarding the educational background, perceived responsiveness, and empathy, education level also led to significant differences in perceptions

of the SERVQUAL dimensions of responsiveness and empathy. For empathy, "high school > university (and above)" indicates that high school-level personnel are more attentive to empathetic service, while for responsiveness, "university (and above) > high school" and "high school > junior high (and below)" suggest that personnel with high school or higher education levels prioritize responsive and empathetic services.

In summary, significant differences were observed across hierarchical level, certification status, and education level. In addition, other demographic factors such as gender, age, years of service, and military duty type were found to be non-significant in this analysis.

In the analysis of differences in expectations for catering quality in SERVQUAL dimensions among customer based on demographic variables, the analysis revealed significant differences based on food service certification status. Independent sample *t*-tests indicated that users with certifications placed higher importance on tangible aspects of catering service quality—such as food, service, equipment, safety, and hygiene—than those without certifications.

In addition, educational background also significantly affected expectations in the SERVQUAL tangibility dimension. Users with a university-level education or higher placed greater emphasis on tangible aspects of catering quality than those with a high school education or below, indicating a perceptual gap based on educational level. In summary, certification status and education level were significant factors, while gender, age, hierarchical level, military duty, and years of service were not.

For the differences in actual experiences of catering quality in SERVQUAL dimensions among customer, the analysis of demographic factors reported significant differences in the SERVQUAL responsiveness dimension based on food service certification status. Users without certifications reported higher responsiveness satisfaction compared to those with certifications, indicating that certification status influences perceptions of responsiveness in actual service experiences.

Years of service also showed significant differences in the assurance and responsiveness dimensions. Users with 1–5 years of service or 6–10 years reported higher levels of assurance and responsiveness than those with over 16 years of service, suggesting that newer employees place a higher emphasis on trust and responsive service quality than longer-serving staff.

In addition, age significantly influenced perceptions of reliability, with younger users (aged 18–25) reporting a stronger expectation for reliable service compared to older users (aged 46–65). This indicates that younger customer are more likely to expect dependable service post-experience compared to their older counterparts.

In summary, age, years of service, and certification status showed significant effects on the perception of actual experiences in catering quality, while gender, hierarchical level, education level, and military duty did not.

5.1.1. Analysis of Differences in Catering Service Quality Perception Between Food Service Providers and Customer (Gap 1)

Customer generally held lower expectations regarding the overall portion sizes and variety of meals, but reported high satisfaction after experiencing the catering service (Chang, 2024). Customer expected food service providers to maintain professional attire, ensure dining hygiene, conduct routine disinfection, maintain appropriate food temperatures (cold/hot), and serve meals on time. In addition, customer anticipated that personnel would have relevant certifications, provide balanced nutrition, use fresh ingredients within their effective dates, ensure container cleanliness, deliver trustworthy service, and offer accessible complaint channels. These expectations were generally met by the food service providers.

For the unmet expectations in food safety, user-centered service, and proactive oversight, customer expected food service providers to ensure the structural integrity of food containers, rigorously control food safety, reliably manage meal provision, prioritize user rights, show proactive concern, make timely adjustments, and conduct routine service reviews. However, food service providers placed less emphasis on these aspects, leading to unmet expectations in these areas.

In terms of the unmet expectations in empathy, responsiveness, and overall cleanliness, customer also expected attentive, considerate service, prompt and accurate responses, and quiet, efficient service that delivers satisfaction in a single attempt. In addition, they held low expectations for overall food quality and cleanliness of the dining environment, and the performance of food service providers in these areas did not lead to high satisfaction among customer.

5.1.2. Analysis of Differences Between Customer' Expectations and Actual Experiences of Catering Service Quality (Gap 5)

Customer generally had low expectations regarding portion size and meal variety, yet reported high satisfaction after experiencing these aspects of the catering service (Chang, 2024). They expected food service providers to maintain clean attire, ensure container and ingredient cleanliness, hold relevant

certifications, control food temperature (both hot and cold), conduct routine disinfection, provide fresh, balanced meals, monitor expiration dates, and review any service errors. They also expected personnel to show proactive concern for customer' needs and these expectations were met with high satisfaction in their experience.

For the unmet expectations in hygiene, reliability, and accessible feedback channels, customer had high expectations for a hygienic dining environment, reliable service, a focus on user rights, responsive problem resolution, and accessible complaint channels. However, actual satisfaction post-experience was lower than expected, indicating a service perception gap in these areas.

In terms of low expectations and low satisfaction in responsiveness and overall quality, customer held low expectations for responsiveness in understanding user needs, accurately addressing issues, meal delivery efficiency, overall food quality, dining environment hygiene, and food flavor satisfaction. These aspects were also rated poorly in actual experience, reflecting low satisfaction and confirming that these areas did not meet user expectations.

5.2. Research Recommendations

Based on the research conclusions, the following three recommendations are proposed, covering cognitive service, expected service, and actual experience, to help military units improve group catering user satisfaction in the future.

5.2.1. Focus Areas for Immediate Improvement

From the perspective of customer' experience, users emphasized the need for strict quality control over food containers, oversight of potential food safety incidents, consistency between served dishes and the menu, and prioritizing user rights in food service. In addition, they expect prompt responses to feedback, routine reviews of service errors, and proactive attention from catering units. The portions and variety of meals were also highlighted as areas with lower satisfaction post-experience, suggesting these should be prioritized for improvement. These elements are critical and should be the focus of immediate action, with food service providers responsiveness considered for secondary improvement.

5.2.2. Maintaining High Standards in Expected Service Quality

Customer reported high satisfaction with aspects, such as personnel appearance, professionalism, environmental hygiene, appropriate food temperature and expiration control, timely meal provision, trustworthy service, and accessible complaint channels. It is recommended that military units maintain these standards consistently.

5.2.3. Training and Development for Enhanced Service Quality

The study indicates that customers prioritize not only food safety and reliability but also quality service and responsiveness during the dining process. To address these needs, it is suggested that service quality and management-related courses be incorporated into training programs for food service providers to improve their service quality.

These recommendations aim to provide a reference for military units as they work to enhance the internal quality of group catering services.

Acknowledgments

None.

Funding

None.

Conflict of Interest

The authors declare that they have no competing interests.

Author Contributions

Conceptualization: Ya-Wen Chan
Methodology: Ya-Wen Chan
Data curation: All authors
Formal analysis: Ya-Wen Chan
Visualization: All authors
Writing – original draft: All authors
Writing – review & editing: Ya-Wen Chan

Availability of Data

Data are not publicly available due to confidentiality restrictions.

References

Blackett, T. (1988). Researching brand names. *Marketing Intelligence and Planning*, 6(3), 5–8. https://doi.org/10.1108/eb045757

Cardozo, R.N. (1965). An experimental study of customer effort, expectation, and satisfaction. *Journal of Marketing Research,* 2(3), 244–249.

https://doi.org/10.1177/002224376500200303
Chang, T.R. (2024). *Exploring Service Quality Gaps in Military Catering using the PZB Model: A Case Study of a Northern Air Force Base (Master's Thesis, Fo Guang University, Department of Health and Creative Plant-Based Food Industries). Fo Guang University Institutional Repository*. Available from: https://hdl.handle.net/11296/na5d6q [Last accessed on 2025 Aug 30].
Czepiel, J.A., Rosenberg, L.J., & Akerele, A. (1974). Perspectives on consumer satisfaction. In: *AMA Conference Proceedings*. American Marketing Association, p119–123.
Foster, S.T. (2001). *Managing Quality: An Integrative Approach*. Upper Saddle River, NJ: Prentice Hall.
Gronroos, C. (1982). *Strategic Management and Marketing in the Service Sector.* Helsinki, Finland: Chartwell-Bratt.
Kazarian, E.A. (1983). *Foodservice Facilities Planning*. New York, NY: John Wiley & Sons.
Lehtinen, J.R., & Lehtinen, U. (1982). *Service Quality: A Study of Quality Dimensions.* Helsinki, Finland: Service Management Institute.
Martilla, J.A., & James, J.C. (1977). Importance-performance analysis. *Journal of Marketing*, 41(1), 77–79. https://doi.org/10.1177/002224297704100112
Maslow, A.H. (1943). A theory of human motivation. *Psychological Review*, 50(4), 370–396. https://doi.org/10.1037/h0054346
Maslow, A.H. (1970). *Motivation and Personality*. 2nd ed. New York, NY: Harper & Row.
Morgan, K.J. (2004). School meals and sustainable food chains: The role of creative public procurement. *The Political Quarterly*, 75(1), 36–45. https://doi.org/10.1111/j.1467-923X.2004.00614.x
Oliver, R.L. (1981). Measurement and evaluation of satisfaction processes in retail settings. *Journal of Consumer Research*, 8(2), 25–48. https://doi.org/10.1086/208842
Parasuraman, A., Zeithaml, V.A., & Berry, L.L. (1985). Problems and strategies in services marketing. *Journal of Marketing*, 49(1), 33–46. https://doi.org/10.1177/002224298504900103
Parasuraman, A., Zeithaml, V.A., & Berry, L.L. (1988). SERVQUAL: A multiple-item scale for measuring consumer perceptions of service quality. *Journal of Retailing*, 64(1), 12–40.
Rosenzweig, P.M., & Singh, J.V. (1991). Organizational environments and the multinational enterprise. *Academy of Management Review*, 16(2), 340–361. https://doi.org/10.5465/amr.1991.4278953
Sasser, W.E., Olsen, R.P., & Wyckoff, D.D. (1978). *Management of Service Operation: Text and Cases*. Boston, MA: Allyn and Bacon.
Wyckoff, D.D. (1984). New tools for achieving service quality. *Cornell Hotel and Restaurant Administration Quarterly*, 25(3), 78–92. https://doi.org/10.1177/001088048402500317

AUTHOR BIOGRAPHIES

Ya-Wen Chan is an Assistant Professor in the Department of Health and Creative Plant-Based Food Industries at Fo Guang University, Taiwan. She earned her Ph.D. in Management and has extensive experience in research on service quality, consumer behavior, and sustainable food systems. Her academic work focuses on applying mixed-method approaches to explore consumer perceptions, purchase intentions, and satisfaction in both institutional and community-based food services. She has published studies on smart farming, plant-based diets, and catering service management in peer-reviewed journals. In addition to her academic research, she is actively engaged in university social responsibility (USR) projects, promoting plant-based nutrition, health education, and community service initiatives across Taiwan and abroad.

Zu-Rong Zhang is a graduate of the Master's Program at the College of Lifelong Learning, Fo Guang University, Taiwan. His research interests focus on service quality management, group catering services, and user satisfaction analysis, with particular attention to the application of the SERVQUAL model and Importance–Performance Analysis (IPA) in institutional settings. During his graduate studies, he developed expertise in empirical research design, quantitative analysis, and survey methodology, contributing to academic projects on food safety and catering service quality. In addition to his academic training, he has practical experience in military logistics and catering management, serving as a First Lieutenant and Logistics Staff Officer in the Air Force Meteorological Squadron. His academic background, combined with professional practice, allows him to bridge theory and real-world application in the improvement of catering services.

DOI: 10.6977/IJoSI.202510_9(5).0003
W. Zhu, S-K. Huang, etc./Int. J. Systematic Innovation, 9(5), 23-42 (2025)

A graphics processing unit-based parallel simplified swarm optimization algorithm for enhanced performance and precision

Wenbo Zhu[1], Shang-Ke Huang[2], Wei-Chang Yeh[2,3]*, Zhenyao Liu[4], Chia-Ling Huang[5]

[1]School of Mechatronical Engineering and Automation, Foshan University, Foshan, Guangdong, China

[2]Integration and Collaboration Laboratory, Department of Industrial Engineering and Engineering Management, College of Engineering, National Tsing Hua University, Hsinchu, Taiwan

[3]Department of Industrial and Systems Engineering, College of Electrical Engineering and Computer Science, Chung Yuan Christian University, Taoyuan, Taiwan

[4]School of Economics and Management, Taizhou University, Taizhou, Jiangsu, China

[5]Department of International Logistics and Transportation Management, School of Transportation and Tourism, Kainan University, Taoyuan, Taiwan

*Corresponding author E-mail: wcyeh@ie.nthu.edu.tw

(Received 13 January 2025; Final version received 02 August 2025; Accepted 04 August 2025)

Abstract

Graphics processing units (GPUs) have emerged as powerful platforms for parallel computing, enabling personal computers to solve complex optimization tasks effectively. Although swarm intelligence algorithms naturally lend themselves to parallelization, a GPU-based implementation of the simplified swarm optimization (SSO) algorithm has not been reported in the literature. This paper introduces a compute CUDA-SSO algorithm on the CUDA platform, with a time complexity analysis of O (Ngen × Nsol × Nvar), where Ngen is the number of iterations, Nsol is the population size (i.e., number of fitness function evaluations), and Nvar represents the required pairwise comparisons. By eliminating resource preemption of personal best and global best updates, CUDA-SSO significantly reduces the overall complexity and prevents concurrency conflicts. Numerical experiments demonstrate that the proposed approach achieves an order-of-magnitude improvement in run time with superior solution precision relative to central processing unit-based SSO, making it a compelling methodology for large-scale, data-parallel optimization tasks.

Keywords: Compute Unified Device Architecture, Graphics Processing Unit, Parallelism, Simplified Swarm Optimization, Swarm Intelligence Algorithms

1. Introduction

In recent years, graphics processing units (GPUs) have significantly impacted high-performance computing, particularly for data- and compute-intensive applications. Originally designed to accelerate real-time three-dimensional graphics, GPUs now offer a parallel architecture that can handle massive throughput in general-purpose scientific computing. Thanks to the availability of thousands of arithmetic logic units (ALUs) and large memory bandwidth, personal computers equipped with modern GPUs have become highly effective platforms for performing large-scale computations (AlZubi et al., 2020; Hachaj & Piekarczyk, 2023). This evolution has fueled a surge of interest in GPU-accelerated algorithms across diverse fields, including medical image processing (Corral et al., 2024; Mittal & Vetter, 2014), energy optimization (Mortezazadeh et al., 2022), and geospatial modeling (Hager et al., 2008).

One notable class of algorithms that can benefit significantly from the massive parallelism of GPUs is swarm intelligence (SI). Swarm intelligence algorithms (SIAs), such as particle swarm optimization (PSO), genetic algorithms (GA), and fireworks algorithms,

draw inspiration from natural phenomena (e.g., bird flocking, fish schooling, and evolutionary processes). By orchestrating collective behaviors, these methods iteratively refine candidate solutions within a high-dimensional search space (Abbasi et al., 2020; Navarro et al., 2014; NVIDIA, n.d.). SIAs naturally lend themselves to parallel implementations, since core operations such as fitness evaluation and local solution updating occur at the per-particle or per-agent level, often with minimal dependency among individuals. Prior studies have documented considerable speedups when porting SIAs to GPU architectures (Tan & Ding, 2015; Yeh, 2017; Yeh & Wei, 2012; Yildirim et al., 2015), highlighting the strong synergy between swarm parallelism and GPU hardware concurrency.

Despite the demonstrated success of GPU-based SIAs, one variant, simplified swarm optimization (SSO), has received limited attention on modern parallel platforms. Since its inception in 2009 (Lee et al., 2012), SSO has proven to be an effective population-based search method, praised for its conceptual simplicity and robust performance on real-world optimization tasks (Corley et al., 2006; Luo et al., 2019; Yeh, 2015). However, existing research on SSO has primarily examined serial (central processing unit [CPU]-based) implementations, leaving a conspicuous gap regarding its parallel potential. By focusing on SSO, researchers can harness its inherently straightforward swarm-update rules to realize high degrees of concurrency. Moreover, the method's minimal parameter requirements and flexible encoding scheme make it a compelling candidate for GPU-based large-scale optimization.

To address this gap, we propose a compute unified device architecture (CUDA) SSO (CUDA-SSO) framework under the NVIDIA CUDA environment. Departing from sequential SSO procedures, CUDA-SSO capitalizes on concurrent kernel launches to distribute the computational workload across thousands of GPU threads. This design not only accelerates fitness evaluations, typically the most time-consuming step in swarm algorithms, but also introduces a parallel update mechanism to circumvent resource-preemption issues associated with personal best (pBest) and global best (gBest) states in swarm-based searches. By carefully encapsulating data in global memory and minimizing CPU–GPU data transfers, we demonstrate both improved solution quality and a drastic reduction in overall execution time.

The main contributions of this paper are:

(i) A novel GPU-based SSO framework (CUDA-SSO) that adopts data-parallel kernels and reduces the theoretical time complexity of swarm search steps.

(ii) A discussion of resource conflict avoidance by re-structuring personal and gBest updates in a parallel context.

(iii) A comprehensive evaluation of standard benchmark functions, showcasing an order-of-magnitude speedup in run time, accompanied by higher solution accuracy than CPU-based SSO implementations.

The remainder of this paper is organized as follows. Section 2 presents an overview of the classical SSO algorithm, the fundamentals of general-purpose GPU computing, and related GPU-based SIAs. Section 3 details the proposed CUDA-SSO algorithm, including its kernel-based design, memory model, and theoretical time complexity analysis. Section 4 provides experimental results with various benchmark functions, comparing performance and precision against the baseline CPU-based SSO. Finally, Section 5 summarizes the findings, discusses potential improvements, and outlines directions for future work.

2. Background

Recent advances in high-performance computing and optimization have witnessed the integration of diverse approaches such as SI, evolutionary strategies, and gradient-based search methods. In particular, SIAs offer decentralized collective search capabilities, while gradient descent (GD) relies on local derivative information to iteratively refine candidate solutions. Understanding how these paradigms intersect—or diverge—can shed light on algorithmic design principles that balance global exploration with local exploitation. This section introduces SSO, a data-parallel swarm algorithm noted for its streamlined update rules. We then highlight key distinctions between GD and swarm-based approaches, discuss the essentials of general-purpose GPU (GPGPU) computing, and conclude with an overview of relevant GPU-based SIAs to contextualize the motivations behind our work on CUDA-SSO.

2.1. SSO

SSO was initially proposed by Yeh (2009) as a lightweight yet robust variant of SI, offering a balance between algorithmic simplicity and practical performance. Unlike more elaborate SIAs (e.g., PSO with velocity–position updates or GA with crossover–mutation operators), SSO employs a small set of parameters (C_w, C_p, and C_g) that guide the sampling of new solutions from each particle's current state (x_{ij}^t), pBest (p_{ij}^t), and gBest (g_j). This approach obviates the need for velocity vectors or mutation rates, reducing the parameter-tuning overhead that can complicate other SIAs.

Fundamentally, each iteration of SSO can be broken into:

(i) Solution update: For each solution i and variable j, the new solution $x_{ij}^{(t+1)}$ is drawn from one of three sources—current solution, pBest, or gBest based on probabilities (C_w, C_p, and C_g).

(ii) Fitness evaluation: Each updated particle is assigned a fitness score $x_{ij}^{(t+1)}$.

(iii) Best-value updates: If f(Xi) is better than a particle's pBest, it is replaced. If f(Xi) outperforms the current gBest, it is updated accordingly.

2.1.1. Fundamental concepts and update strategy

SSO operates over a population $\left\{X_i^t \mid i = 1, 2, \ldots, N_{sol}\right\}$, where $X_i^t = (x_{i,1}^t, x_{i,2}^t, ..., x_{i,m}^t)$ is a vector representing the i^{th} candidate solution at generation t and $x_{i,j}^t$ is the j^{th} variable in X_i^t for $t = 1, 2, \ldots, N_{gen}$ and $i = 1, 2, \ldots, N_{sol}$. Two supporting data structures track the algorithm's progress:

(i) pBests: $P_i = (p_{i,1}, p_{i,2}, p_{i,m})$: The historically best position of each particle, reflecting individually optimal solutions found over previous iterations.

(ii) gBest: $P_{gBest} = (g_1, g_2, g_m)$: The optimal solution observed across the entire population.

Within each iteration, SSO applies a simple step function to update the value of each variable $x_{i,j}^t$ in the solution X_i^t. As shown in Eq. (1), a random number ρ is a random value drawn from a continuous distribution ranging from 0 to 1, which drives the selection among four possibilities: retaining the current value $x_{i,j}^t$, adopting $p_{i,j}$, adopting g_j, or performing no update.

$$x_{i,j}^{t+1} = \begin{cases} x_{i,j}^t \text{ if } \rho \in \left[0, C_w = c_w\right) \\ p_{i,j} \text{ if } \rho \in \left[C_w, C_p = C_w + c_p\right) \\ g_j \text{ if } \rho \in \left[C_p, C_g = C_p + c_g\right) \\ x \text{ if } \rho \in \left[C_g, 1\right) \end{cases} \quad (1)$$

Here, $p_{i,j}$ denotes the j^{th} coordinate of the pBest of the i^{th} solution, and g_j represents the corresponding coordinate in gBest. The relative magnitudes of (C_w, C_p, and C_g) balance exploration (i.e., adopting global or pBests) against exploitation (i.e., retaining current values). This compact parameterization facilitates a more controlled search dynamic than in many other SIAs.

2.1.2. Advantages of SSO over genetic algorithms

Genetic algorithms have historically been a cornerstone of evolutionary computation, relying on crossover and mutation operations to evolve solution populations. However, SSO can frequently perform better in certain problem classes due to its simpler update mechanism and more focused parameter space. Key comparative advantages of SSO include:

(i) Reduced parameter tuning: Traditional GAs demand meticulous adjustment of crossover rates, mutation probabilities, and selection schemes. By contrast, SSO relies on three probabilities (Cw, Cp, and Cg) to guide each variable's update. This hyperparameter reduction often translates into faster and more reproducible experimentation, minimizing the risk of suboptimal tuning.

(ii) Potentially faster convergence: In SSO, particles can directly adopt globally optimal positions, whereas GAs depend on randomized genetic operators to spread promising traits. Consequently, SSO may converge more rapidly on certain continuous or weakly multimodal functions, mainly when the objective landscape permits direct exploitation of high-fitness regions.

(iii) Implementation simplicity: GA-based crossover and mutation operators can become complicated when dealing with high-dimensional or heterogeneous solution representations. SSO's step-function update—requiring only a few lines of code—facilitates implementation clarity, reducing the likelihood of design or coding errors.

(iv) GPU suitability: Although GAs can be parallelized, SSO's probabilistic mechanism, wherein each variable is updated according to a small set of global or pBests, typically presents fewer data dependencies across particles. This structure lends itself well to massive parallelization on GPUs, making SSO an attractive option for large-scale optimization tasks in high-performance computing environments.

Hence, SSO offers a comparatively straightforward and potentially more consistent pathway to large-scale optimization, particularly when research or industrial constraints limit tuning resources or demand high solution fidelity within compressed timeframes.

2.1.3. SSO flowchart

SSO's simplicity has proven advantageous in several applications. For instance, Chung & Wahid (2012) and Yeh (2012; 2013) demonstrate its effectiveness in tackling complex real-world tasks such as reliability design and feature selection. Further refinements, such as orthogonal SSO (Yeh, 2014), reinforce the adaptability of SSO's framework. However, although prior literature confirms SSO's suitability for large-scale research, most studies have employed CPUs, where time complexity grows rapidly

with the population size and dimensionality. This motivates the pursuit of a GPU-based parallelization strategy that can leverage SSO's inherent data-parallel characteristics.

Algorithm 1 outlines the typical CPU-based SSO flow. Each iteration updates particles by sampling the step function, evaluates the fitness value for each particle, and updates pBests and gBest if any improvement is found. Although CPU-SSO can yield excellent results for moderate-scale problems, it becomes slow when the population and number of variables are large.

Algorithm 1. The typical CPU-based SSO

```
Initialize:
    Nsol = 50, Nvar = 30, Ngen = 100
    Var_max = 5.12, Var_min = -5.12
    sol = Nsol × Nvar
    pBests = Nsol × Nvar
    gBest = 0
    Cw = 0.2, Cp = 0.5, Cg = 0.8
    explorationTime = 0
while explorationTime ≤ cpuTimeLimit do
    for iter in 1 to Ngen do
    stepFunc(sol, pBests, gBest, randNum(Var_max,
    Var_min))
     evaluate(solF, pF, gF)
      if solF < pF then pBests(i) = sol(i)
         if solF < gF then gBest = sol(i)
         end if
       end if
     end for
end while
```

2.2. General-Purpose GPU Computing

Modern GPUs were originally engineered to accelerate real-time three-dimensional graphics tasks such as rasterization and shading. Over time, these architectures evolved into GPGPU (Hussain et al., 2016), wherein highly parallel GPU hardware is repurposed to handle a variety of data-intensive computations. By distributing large workloads among thousands of arithmetic cores, developers offload parallel tasks to the GPU while reserving more complex, serial procedures for the CPU.

2.2.1. Execution model (CUDA framework)

NVIDIA's CUDA (NVIDIA, n.d.) extends C/C++ to enable heterogeneous computing. In CUDA, the following function types determine where (CPU vs. GPU) and how (serial vs. parallel) code is executed:

(i) Host functions: Host code is defined in C/C++ and runs on the CPU. It is responsible for high-level logic, memory allocation, and kernel launch.

(ii) Kernel functions: GPU kernels are invoked by the CPU but executed on the GPU, and are subdivided into thread blocks and further organized into warps of 32 threads, following the single instruction, multiple threads paradigm. They are ideal for data-parallel workloads such as fitness evaluations or array/vector operations.

(iii) Device functions: Device functions are defined and executed only on the GPU and are typically called from within kernel functions to factor out repeated computations.

In this model, thousands of concurrent threads can be spawned to run the same kernel, allowing GPUs to efficiently process large, independent datasets.

2.2.2. Compute unified device architecture memory hierarchy

Compute Unified Device Architecture's memory model separates storage into multiple tiers, each balancing capacity and speed.

(i) Registers: Per-thread registers provide high-speed storage and are best suited for frequently accessed variables that do not exceed the register file capacity.

(ii) Shared memory: On-chip shared memory allocated per block enables fast data exchange among threads in the same block and is particularly useful for shared computations, partial sums, and other cooperative tasks where multiple threads access and modify the same data.

(iii) Global memory: Off-chip global memory provides large-capacity storage accessible by all threads but has relatively high latency compared to on-chip resources, making efficient access patterns (e.g., memory coalescing) essential to achieve high throughput.

(iv) Constant and texture memory: Read-only caches accelerate common look-ups and are helpful when all threads repeatedly use the same constant or when two-dimensional array access patterns can be optimized via texture hardware.

High-performance GPU applications often involve coalescing memory accesses, judiciously using shared memory, and minimizing branch divergence (warp divergence). These considerations ensure that multiple threads fetch contiguous elements simultaneously and execute consistent instruction paths whenever possible.

2.2.3. Data transfers and central processing unit-GPU coordination

Since the CPU and GPU have separate memory spaces, data must typically be transferred via the

Peripheral Component Interconnect Express (PCIe) bus. Although essential for many GPGPU workflows, these transfers introduce non-negligible latency. Strategies to reduce transfer overhead include:

(i) Batching data: Copying large chunks of data at a time rather than frequent small transfers.
(ii) Asynchronous transfers: Overlapping data transfers with kernel execution improves device utilization.
(iii) Unified Memory: Leveraging CUDA's managed memory features to let the runtime handle page migrations between CPU and GPU, albeit with some overhead for page-fault handling.

2.2.4. Implications for SIAs

SIAs—including PSO, GA, Firefly Algorithm, and SSO—naturally benefit from GPGPU acceleration due to their population-based structure. Each individual (particle, agent, or chromosome) can be evaluated in parallel, and gBest values can be updated in a relatively small overhead step.

(i) Fitness evaluations: Commonly, the most significant computational bottlenecks can be massively parallelized by assigning a subset of particles (or subdimensions) to separate threads or warps.
(ii) Update mechanisms: Since SIA updates often involve reading global parameters (e.g., best solutions) and then writing back updated values for each particle, careful design of coalesced memory accesses and thread synchronization (e.g., to avoid race conditions when writing to a gBest value) is critical.
(iii) Data dependencies: Many SIAs only require limited information exchange—such as neighbor-based or globally best-based communication—so the parallel workload is generally well-defined. Nonetheless, if a swarm's communication topology is complex (e.g., hierarchical or multiswarm structures), the kernel must incorporate additional synchronization steps or multiple kernel launches to handle inter-group interactions without causing warp divergence or data hazards.

When population sizes or problem dimensions become large, GPU-enabled SIAs can harness thousands of parallel threads across multiple streaming multiprocessors (SMs), substantially reducing run time relative to CPU-only approaches. Consequently, adopting CUDA or similar frameworks for SIAs—while paying close attention to memory usage, thread management, and synchronization—can yield significant speedups in large-scale optimization scenarios. Synchronization in CUDA refers to coordinating the execution of threads to wait for each other at specific points—usually to ensure that data dependencies are respected (i.e., one thread does not read a value before another finishes writing it).

2.3. GPU-Based SIAs Implementation

Parallelization of SIAs on GPUs leverages the natural data-parallel structure of these methods. Within each iteration, every swarm particle (or agent) usually updates its position, evaluates its objective function, and exchanges information with other particles according to the algorithm's communication model.

2.3.1. An Overview of notable GPU-based SIA

Table 1 provides an overview of notable GPU-based SIAs, detailing which functions were ported to GPU kernels in representative studies. The summarized methods include standard and Euclidean PSO (Tsutsui & Fujimoto, 2009; W. Zhu, 2011), multichannel PSO (Krömer et al., 2011), multi-objective Gas (Wong, 2009; H. Zhu et al., 2011), and GA/differential-evolution hybrids (Mussi et al., 2011; Ruder, 2016), among others.

As these steps can be performed independently or partially synchronized, the GPU is well-suited to handle the large number of concurrent threads required to process high-dimensional populations.

2.3.2. Four key kernel functions

SIAs naturally align with parallel architectures due to their population-based structure (Yeh, 2017; Yeh & Wei, 2012). In a GPU context, typical SIA workflows can be divided into four key kernel functions:

(i) Initialize (I): Kernel Function (I) initializes the population with random numbers and stores them in global memory. Benefiting from the intuitive implementation and data access in global memory, most SIAs generated the population on the CPU (NVIDIA Corporation, 2012). It might have got a vast improvement for computing efficiency if (I) the population on GPU instead of CPU, although the way to arrange the global memory may not be that intuitive (Mussi et al., 2011; Ruder, 2016).
(ii) Evaluate fitness (E): Krömer et al. (2011) have demonstrated that the most expensive step in SIAs was to evaluate candidate solutions. The most straightforward to deploy kernel function (E) is the master–slave paradigm, where the centralized controller dispatches particles in a single population for parallelism. This approach introduced no differences from an algorithmic perspective but reduced the time-consuming from a computational perspective.

Table 1. Summary of studies of taxonomy analysis for swarm intelligence algorithms

References	Swarm intelligence algorithm	Methodology	Speedup
Tsutsui & Fujimoto (2009)	Stand particle swarm optimization (PSO)	(I), (C), (U) on CPU. (E) on a GPU without shared memory	×6–8
W. Zhu (2011)	Euclidean PSO	(I), (C), (U) on CPU. (E) on a GPU without shared memory	×1–5
Krömer et al. (2011)	Multichannel PSO	(U) on CPU, (I), (E), (C) on a GPU without shared memory	×30
Wong (2009)	Multi-objective genetic algorithm (GA)	(I) on CPU, (E), (C), (U) on a GPU without shared memory	10–2
H. Zhu et al. (2011)	Coarse-grain parallelization of GA	(I), (C), (U) on CPU, (E) on a GPU only without shared memory	×60
Li & Zhang (2011)	Asynchronous and synchronous PSO	(I), (E), (C), (U) on a GPU with shared memory	-
Mussi et al. (2011)	GA	(I), (E), (C), (U) on a GPU with shared memory	×2–12
Ruder (2016)	GA and differential evolution (DE)	(I), (E), (C), (U) on a GPU with shared memory and synchronization	×3–28 for GA, ×19–34 for DE

Abbreviations: C: Communication; E: Evaluate fitness; I: Initialize; U: Update swarm

As shown in Table 1, Li & Zhang (2011) proposed a CUDA-based multichannel particle swarm algorithm. Wong (2009) implemented a parallel multi-objective GA. Tsutsui and Fujimoto (2009) ran a sequential SIA, dispatching a parallel GA for the particles.

According to NVIDIA (n.d.) and Mussi et al. (2011), using shared memory in GPU code can guarantee speedup for data transferring. However, most did not perform (E) using shared memory.

(i) Communication (C): Unlike the directly distributing function (E), the function (C) proposes a more complicated model. It is distinguished by being loosely connected to the population and irregularly exchanging particles. Communicate mechanisms were enabled between swarms according to the law of data access, which means that communication between distributed groups of particles is acceptable.

(ii) Update Swarm (U): Adjust the positions or velocities (if applicable) of each particle based on shared information. Function (C) and function (U) do not have a single pattern to fit all SIAs. We must only attend to the warp divergence and bank conflict in these two functions.

Across these works, the (E) kernel typically offers the largest room for speedup, since fitness calculation often dominates the total run time. Many authors have thus focused on accelerating (E) by distributing the population's fitness evaluations to GPU threads.

2.3.3. Implementation challenges

Despite the potential computational gains, several implementation challenges arise when porting SIAs to GPUs:

(i) Memory-access patterns and coalescing: Efficient GPU kernels rely heavily on coalesced global-memory transactions, whereby consecutive threads access consecutive memory addresses. Achieving such patterns can involve reorganizing particle data structures, interleaving population elements, or carefully aligning data to minimize misaligned accesses. Failure to do so can negate much of the theoretical speedup from parallelization.

(ii) Shared memory constraints: While shared memory is a low-latency on-chip resource that can accelerate repeated data accesses, the amount available per block (commonly 48 KB or less) may be insufficient for storing large populations or high-dimensional problems. Consequently, many GPU-based SIAs place most of their data in global memory and resort to shared memory only for small suboperations, such as partial sums or local best-value comparisons.

(iii) Warp divergence and synchronization: GPU threads operate in warps of 32 concurrent threads. If branches in the kernel cause differing execution paths within the same warp, performance can degrade significantly due to warp divergence. SIA kernels that incorporate random sampling, conditionals for updating best solutions, or communication topologies must minimize thread divergence and carefully place synchronization barriers (syncthreads or kernel launches) to avoid race conditions when reading/writing global or shared data structures (e.g., gBest positions).

(iv) Communication topologies: In many SIAs, information sharing is crucial for guiding the swarm. This communication can be ring-based,

star-based, hierarchical, or fully connected. Implementing these topologies on a GPU requires balancing frequent data exchanges with the cost of global or shared-memory transactions, especially as the population grows. Some researchers tackle this by employing loosely coupled subswarms, reducing the number of cross-group communications and associated overhead.

(v) Scalability and precision: GPU-based SIAs often demonstrate significant speedups over CPU counterparts when the population size is large enough to saturate GPU resources. However, if the swarm or dimensionality is too small, kernel-launch overhead and data-transfer latencies may outweigh parallelization benefits. Furthermore, some applications demand higher-precision arithmetic (e.g., double precision) that can reduce throughput on specific GPU architectures. Algorithm designers must thus tune swarm sizes, memory layouts, and data precision settings for optimal results.

These considerations indicate that GPU-based SIAs benefit most when carefully tailored to exploit hardware concurrency while mitigating memory and synchronization bottlenecks. Ongoing advances in GPU architectures—expanded on-chip memory, more sophisticated warp schedulers, and built-in library support—continue to ease the adaptation of SIAs for large-scale, real-world optimization problems.

Building on these insights, the present work aims to extend SSO into the GPU domain, integrating the conceptual simplicity of SSO's update mechanism with the massive parallelism of CUDA. Our proposed CUDA-SSO applies kernel-based parallelization to SSO's most time-consuming and data-parallel steps, achieving significant speed gains and avoiding concurrency conflicts when updating personal and gBest states. In the following section, we elaborate on the algorithmic framework of CUDA-SSO, including memory organization, random number generation, and a theoretical complexity analysis.

3. Compute Unified Device Architecture-SSO

Compute Unified Device Architecture-SSO adapts the conventional SSO to leverage CUDA's parallelism. As illustrated in Fig. 1, each kernel function runs concurrently across threads, reducing both evaluation time and memory transaction overhead.

3.1. Random Number Generation

Random number generation (RNG) is essential in SIAs because almost every aspect of the search—particle initialization, stochastic exploration, and crossover/mutation (in other SIAs)—depends on drawing pseudo-random values. In CUDA-SSO, these numbers govern how each variable in a particle decides whether to retain its current value, adopt its pBest, or adopt the gBest. As a result, generating robust random values at high speed is critical to ensure both algorithmic performance and solution diversity.

A naive approach to RNG would compute random numbers on the CPU and then transfer them to the GPU each iteration. However, such data movement across the PCIe bus can introduce significant latency. Instead, CUDA-SSO uses NVIDIA's cuRAND (random number generation library (NVIDIA, n.d.) to generate random numbers directly on the GPU, thereby reducing CPU–GPU switching overhead. The following points highlight key considerations for efficient RNG in CUDA-SSO.

(i) cuRAND generators: NVIDIA's cuRAND library provides multiple generator types (e.g., Philox, Mersenne Twister, and XORWOW) suited to various performance and quality requirements. Philox typically offers a good balance for most GPU-based Monte Carlo or optimization tasks due to its combination of speed and sufficiently robust randomness.

(ii) State management: A dedicated initialization kernel uses cuRAND application programming interfaces to set up independent RNG states for each thread on the GPU. Each state is assigned a seed, sequence number, and offset. This allows threads to maintain independent RNG states, avoiding global memory contention during the main kernel execution.

(iii) Scalability: Due to CUDA-SSO allocating one or more threads per particle/variable, the number of random values can become quite large, reaching Nsol × Nvar × Ngen. However, cuRAND's batched generation methods allow bulk requests of random values, leveraging GPU concurrency to rapidly produce millions of samples.

(iv) Memory footprint and access: RNG states are typically stored in global memory for all threads to access during kernel execution, with each thread updating its local state after retrieving random samples via curand (& state). To minimize overhead, threads often load their RNG state into registers, generate all required samples, and write the state back to global memory only once per iteration, reducing global memory transactions.

(v) Kernel integration: Each thread within the main CUDA-SSO search kernel can invoke cuRAND library calls to draw random floats (e.g., uniform or normal distributions) and apply them to the SSO step function. While careful synchronization may be necessary if multiple threads share RNG states, this is typically avoided by assigning

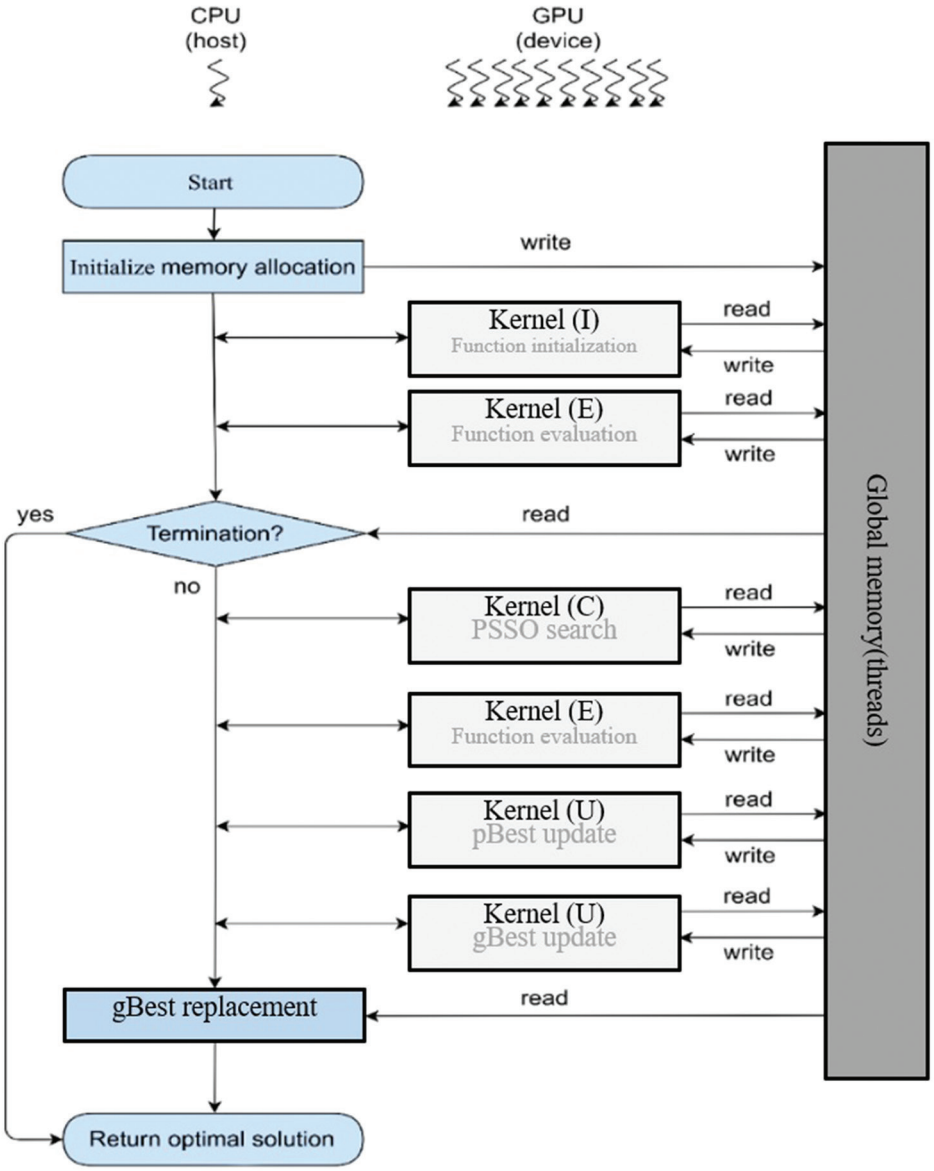

Fig. 1. Proposed compute unified device architecture-simplified swarm optimization
Abbreviations: C: Communication; CPU: Central processing unit; E: Evaluate fitness; gBest: Global best; GPU: Graphics processing unit; I: Initialize; pBests: Personal bests; PSSO: Particle-based simplified swarm optimization; U: Update swarm

unique states to each thread.

(vi) Quality versus speed: While XORWOW offers faster performance, it may exhibit lower randomness quality for specific statistical tests. Although Philox or Mersenne Twister variants may run slightly slower, they often deliver more reliable distributions. While most swarm optimizations work well with any reasonably distributed, uncorrelated RNG, mission-critical or precision-sensitive applications may require more robust generators.

By generating all random numbers on the GPU, CUDA-SSO avoids frequent PCIe transfers and ensures that random samples are available on demand with minimal latency. This strategy significantly improves the algorithm's scalability, allowing $N_{sol} \times N_{var} \times N_{gen}$ random draws to be produced efficiently as the swarm evolves. Consequently, RNG bottlenecks, which often plague GPU-accelerated optimization, are effectively mitigated, paving the way for faster and more diverse exploration in the high-dimensional search space.

3.2. Thread Organization

Efficient thread organization is a cornerstone of high-performance GPU applications, and CUDA-SSO

takes advantage of CUDA's execution hierarchy to maximize throughput and minimize uncoalesced memory accesses. This section details how thread blocks, warps, and memory layouts are arranged to accommodate large particle populations and high-dimensional search problems.

3.2.1. Warp-level particle management

In CUDA-SSO, each warp—consisting of 32 threads—typically maps to one particle, such that the warp's threads can collaboratively handle that particle's variables (position vector, random updates, and fitness computation). This design has several advantages.

(i) Straightforward synchronization: Since a warp executes in a lockstep single-instruction multiple-threads fashion, synchronization within the warp is simpler. For many operations, native warp intrinsics (e.g., __syncwarp()) allow partial sums or shared computations to be done without incurring the overhead of a block-wide synchronization (__syncthreads()).

(ii) Fine-grained parallelism: If a particle has N_{var} variables, they can be distributed across multiple threads, allowing partial work (e.g., updating each variable or computing partial fitness) to proceed in parallel within the same warp.

(iii) Reduced warp divergence: Since all threads in a warp handle logically contiguous parts of the same particle, branching is minimized. Divergence primarily arises if the particle's data triggers conditionals (e.g., random updates to different variables). However, these are usually minor compared to divergences caused by dissimilar data accesses across multiple particles.

Compute unified device architecture's thread blocks group warps together, and a grid of blocks covers the entire population.

Block sizes are chosen in multiples of 32 (e.g., 128, 256, and 512 threads/block) to ensure warp alignment. In CUDA-SSO, a block typically manages several particles—each warp in the block handles a separate particle's data.

Grid sizes are determined by how many blocks are needed to encompass all particles. For instance, if the swarm has N_{sol} = 10,000 particles and each block manages eight warps, we need at least 10,000/8 = 1,250 blocks to cover the swarm. This approach scales well on modern GPUs with multiple SMs capable of running dozens of blocks concurrently.

To fully utilize GPU bandwidth, CUDA-SSO arranges each particle's data (e.g., position vector, best values) contiguously in global memory. When warp threads access consecutive addresses, coalesced reads reduce the required memory transactions. Key design elements include:

(i) Particle-centric layout: The position vector, pBest, and related metadata for each particle are stored back-to-back in memory. Threads within a warp access sequential indices, aligning memory requests with hardware transaction boundaries.

(ii) Avoiding strided access: If data for a single particle were scattered or interleaved with multiple particles, warp threads would fetch non-consecutive addresses, leading to uncoalesced accesses and lowered throughput. By contiguously grouping a particle's variables, CUDA-SSO preserves coalescing even when the swarm is large.

(iii) Shared memory trade-off: Although shared memory can accelerate repeated data accesses (e.g., partial sums), large swarm sizes (hundreds or thousands of particles, each with tens to hundreds of variables) rapidly exceed the typical 48–96 kb shared memory per block. Consequently, global memory becomes the main data store. Nevertheless, kernel designers may still use shared memory for sub-operations (e.g., block-level reductions) if it is feasible within the memory budget.

3.2.2. Synchronization and concurrency

Swarm intelligence demands occasional synchronization to ensure that updated particle states or gBest values are consistently available. In CUDA-SSO, two main synchronization patterns arise:

(i) Warp-level: For tasks that only require threads within the same warp to coordinate—such as partial computation of a single particle's fitness—warp intrinsics (__syncwarp()) suffice. This is faster than a full __syncthreads(), affecting all block threads.

(ii) Block- or grid-level: Specific global or pBest updates may require broader synchronization:

- __Syncthreads() ensures all threads in the block finalize local data before proceeding.
- Multiple kernel launches act as implicit grid-wide barriers, guaranteeing that all blocks complete one stage (e.g., updating pBests) before starting the next (e.g., computing the gBest).

Ensuring all local updates are complete before any best-value comparisons helps avoid race conditions, which might otherwise lead to inconsistent reads or partial updates of shared variables.

For huge swarms or high-dimensional search spaces, a single kernel launch might strain available GPU memory or underutilize certain multiprocessors. CUDA-SSO addresses these scenarios by subdividing the population:

(i) Population splitting: Instead of handling all NsolN_{\mathrm{sol}} particles in one kernel, the

swarm can be partitioned into subsets processed by multiple sequential kernel launches or multiple streams. Each subset undergoes search and fitness evaluation before merging partial bests.

(ii) Multi-kernel scheduling: Modern GPUs support concurrent kernels, enabling partial overlaps in execution. If each subset's memory footprint is smaller, more streams can run concurrently on different SMs, improving load balancing and overall throughput.

(iii) Trade-off: Although subdividing can improve concurrency, it introduces additional steps for merging partial gBest values across subsets. Careful scheduling is needed so that merging overhead does not offset gains from improved load distribution.

By adhering to warp-based particle updates, coalesced memory access patterns, and appropriate synchronization, CUDA-SSO efficiently distributes workload across a GPU's many SMs. In turn, this enables (i) high utilization, where a large swarm or high-dimensional setting can saturate GPU computational resources, (ii) scalability, where as problem sizes grow, additional blocks and warps smoothly extend parallel coverage, and (iii) maintainability, where warp-level design keeps each particle's logic self-contained, simplifying debugging and code maintenance.

Developers must still tune parameters such as block size, register usage, and shared-memory allocations for specific GPU architectures (e.g., differences between NVIDIA Turing, Ampere, or Hopper architectures). Nonetheless, the fundamental strategy—one warp per particle, coalesced global memory, and synchronization barriers for best-value consistency—forms a robust template for realizing scalable, high-performance SI on GPUs (Gordon & Whitley, 1993; Hadley, 1964; Wolpert & Macready, 1995).

3.3. Compute Unified Device Architecture-SSO Implementation

Leveraging GPU-based parallelism requires a careful design of kernel functions, memory layouts, and synchronization strategies. In CUDA-SSO, each iteration (or generation) processes a large population of particles on the GPU, avoiding frequent transfers across the PCIe bus. By dividing search, fitness evaluation, and best-value updates into separate kernels, the algorithm can efficiently harness the GPU's concurrent execution model.

3.3.1. Kernel-launch structure

Algorithm 2 illustrates the main flow of CUDA-SSO. Each generation begins with random number generation on the GPU, followed by parallel kernels for the search process (step function) and fitness evaluations. Afterward, pBests and the gBest are updated in parallel, with each block or warp managing a subset of particles.

Algorithm 2. Flowchart for CUDA-simplified swarm optimization

```
sol = Nsol × Nvar
pBests = Nsol × Nvar
gBest = 0
set block size
syncThreads()

Initialize population
Initialize block size
Transfer data from CPU to GPU

//Kernel functions executed in parallel
for gen = 0 to Ngen do
    Search process for all particles      //stepFuncin
                                          parallel
    syncThreads()
    Update pBest for each solution        //Kernel (U)
    Update gBest for each solution        //Kernel (U)
    syncThreads()
end for

Send data back to the CPU
```

The above design leverages the GPU's parallel capabilities to handle large numbers of particles in each generation and ensures that intermediate results are kept consistent across all threads before the next update commences. Here is how it works:

(i) Parallel kernel launches: The design separates operations into distinct parallel kernels for the search process (step function) and for updating pBests and gBest values. This approach enables the concurrent execution of computation (E) and communication (C) operations before synchronizing for updates (U).

(ii) Synchronization: The system uses syncThreads() or similar synchronization barriers to ensure all threads complete their current operations, whether searching or updating optimal values, before moving forward. This synchronization is vital for preventing race conditions and maintaining consistent pBests and the gBest.

(iii) GPU–CPU transfers: To minimize PCIe bus overhead, data transfers between CPU and GPU occur only twice: once at initialization and once at completion. During iterations, all population data remains in GPU memory, following the memory management guidelines outlined in Section 3.2.

3.3.2. Parallel updates of pBests and gBests

Algorithms 3 and 4 illustrate how pBests and the gBest are updated in a parallel environment. By distributing the workload across GPU threads, CUDA-SSO prevents any single update from dominating run time and fully exploits GPU concurrency.

Algorithm 3. Parallel updates of personal bests.
```
syncThreads()
for each particle i in parallel do
    Load current sol[i] and pBests[i]
    if f(sol[i]) < f(pBests[i]) then
        pBests[i] = sol[i]
    end if
end for
syncThreads()
```

Algorithm 4. Parallel updates of the global best.
```
syncThreads()
for each particle i in parallel do
    Load current pBests[i] and gBest
    if f(pBests[i]) < f(gBest) then
        gBest = pBests[i]
    end if
end for
syncThreads()
```

Implementation details of Algorithms 3 and 4 are discussed in the following:

(i) Warp/block-level work: Each particle is processed in parallel. While it is not explicitly stated that one warp must correspond to a single particle, this configuration can be achieved by selecting suitable block and grid sizes, thereby reducing warp divergence and simplifying synchronization.

(ii) Coalesced memory access: In these snippets, each thread (or warp) reads data stored contiguously in global memory for the assigned particle i. If both sol *i* and pBest *i* reside in adjacent memory locations, warp-level access requests naturally coalesce into fewer transactions.

(iii) Synchronization points: The syncThreads() calls at the start and end of each code block ensure that all local read/write operations to pBests or gBest finish before another kernel or step begins. That is, the communication for global search does not rely on synchronization mechanisms, as these typically incur substantial overhead. Such barriers prevent partial updates or inconsistent reads across parallel threads.

3.4. Time Complexity Analysis

Compared to CPU-SSO's sequential structure, CUDA-SSO distributes the update and evaluation workload over many GPU threads, effectively reducing the time complexity within each iteration. Fig. 2 contrasts CPU-SSO's single-thread approach versus CUDA-SSO's multi-thread parallelism. While CPU-SSO tends to scale with $O(n^3)$ under large

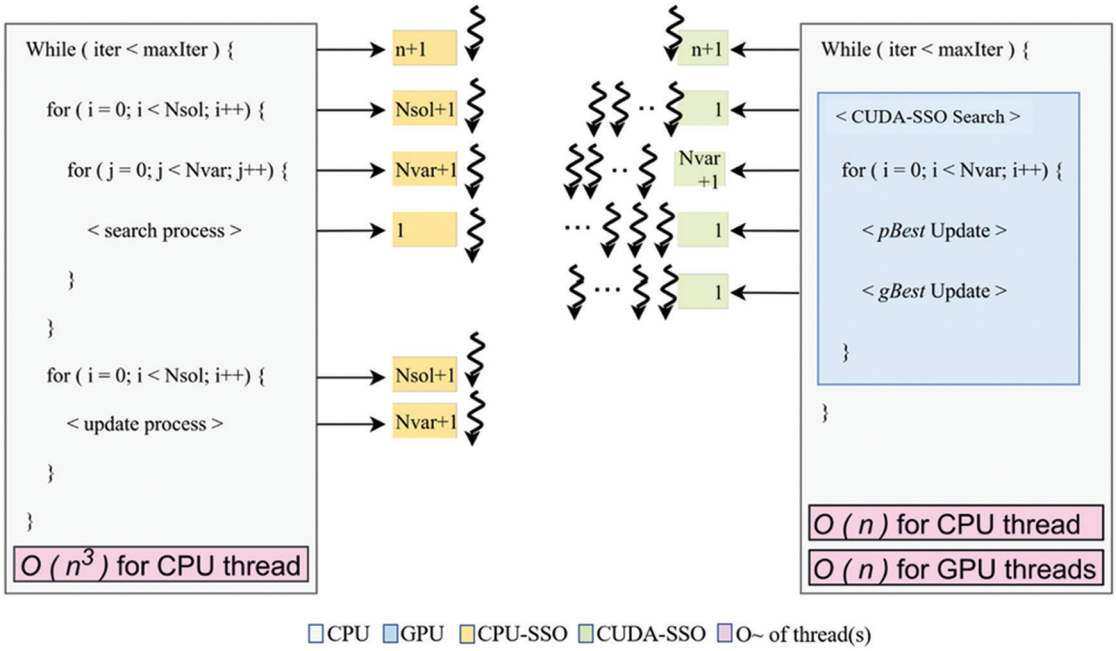

Fig. 2. The time complexity analysis

Abbreviations: C: Communication; CPU: Central processing unit; CUDA: Compute unified device architecture; gBest: Global best; GPU: Graphics processing unit; pBests: Personal bests; SSO: Simplified swarm optimization

W. Zhu, S-K. Huang, etc./Int. J. Systematic Innovation, 9(5), 23-42 (2025)
DOI: 10.6977/IJoSI.202510_9(5).0003

Table 2. Benchmark functions for compute unified device architecture-simplified swarm optimization

ID	Function	Definition	Feasible bounds
f_1	Sphere	$f_1 = \sum_{i=1}^{P} x_i^2$	$[-5.12,5.12]^P$
f_2	Hyper-epllipsoid	$f_2 = \sum_{i=1}^{P} i \cdot x_i^2$	$[-5.12,5.12]^P$
f_3	Schwefel's	$f_3 = \sum_{i=1}^{P} \left(\sum_{j=1}^{i} x_j^2 \right)$	$[-65.536,65.536]^P$
f_4	Rosenbrock	$f_4 = \sum_{i=1}^{P-1} \left[100 \cdot \left(x_{i+1} - x_i^2 \right)^2 + \left(1 - x_i \right)^2 \right]$	$[-2.048,2.048]^P$
f_5	Rastrigin	$f_5 = 10 \cdot P + \sum_{i=1}^{P-1} \left[x_i^2 - 10\left(2\pi x_i \right) \right]$	$[-5.12,5.12]^P$
f_6	Ackley	$f_6 = -a \cdot \exp\left(-b \cdot \sqrt{\frac{1}{P} \sum_{i=1}^{P} x_i} \right) - \exp\left(\frac{1}{P} \sum_{i=1}^{P} \cos\left(c x_i \right) \right)$	$[-32.768,32.768]^P$
f_7	Griewank	$f_7 = \frac{1}{4000} \sum_{i=1}^{P} x_i^2 - \prod_{i=1}^{P} \cos\left(\frac{x_i}{\sqrt{i}} \right) + 1$	$[-600,600]^P$
f_8	Powell	$f_8 = \sum_{i=1}^{P/4} \left[\left(x_{4i-3} + 10 x_{4i-2} \right)^2 \right] + \left[5\left(x_{4i-1} - x_{4i} \right)^2 + \left(x_{4i-2} - 2x_{4i-1} \right)^4 \right] + \left[10\left(x_{4i-3} - x_{4i} \right)^4 \right]$	$[-4,5]^P$
f_9	Schwefel	$f_9 = 418.9829 \cdot P - \sum_{i=1}^{P} x_i \cdot \sin\left(\sqrt{\lvert x_i \rvert} \right)$	$[-5.12,5.12]^P$

population sizes, CUDA-SSO exhibits near $O(n)$ scaling in the dominating computational kernel.

Table 3. Experimental parameters of compute unified device architecture-simplified swarm optimization

No.	Graphics processing unit model	Compute unified device architecture-simplified swarm optimization
1	Block size	C_w, C_p, C_g
2	-	Population size: N_{sol}
3	-	Number of variables: N_{var}
4	-	Number of generations: N_{gen}

Table 4. Factor for the parameters of compute unified device architecture-simplified swarm optimization search

No.	*Cw*, *Cp*, and *Cg*
1	0.1, 0.3, 0.7
2	0.1, 0.4, 0.8
3	0.2, 0.4, 0.6
4	0.2, 0.5, 0.9
5	0.3, 0.4, 0.5
6	0.3, 0.6, 0.8

4. Experiments and Analysis

4.1. Benchmark Functions and Design of Experiments

We tested nine standard benchmark functions, shown in Table 2. These functions include both separable and inseparable properties, with multimodal and unimodal complexities. Each function has a dimension of $N_{var} = 50$. By controlling parameters such as N_{gen} (the maximum iteration count), N_{sol} (population size), and N_{var} (number of variables), we gauge both the convergence (precision) and run time (speedup) of CPU-SSO versus CUDA-SSO.

From Table 3, we know we need to do a seven-factor experimental design, 128 experiments. It is impossible to do such a job with contracted computational resources. Thus, the parameters: block size, N_{sol}, N_{var}, and N_{gen} were arranged as follows: 1,024, 100, 50, and 1000, referring to other papers (Li & Zhang, 2011; NVIDIA Corporation, 2012).

The remaining parameters to be tested are the CUDA-SSO search parameters: C_w, C_p, and C_g. Six parameter levels were evaluated in the experiments, as shown in Table 4. The experimental design of the parameter combinations presented in Table 4 was analyzed using scipy.stats library (Pllana & Xhafa,

Table 5. The parameter combinations analyzed using the Kruskal–Wallis H-test

Parameters	Values					
Cw	0.1	0.1	0.2	0.2	0.3	0.3
Cp	0.3	0.4	0.4	0.5	0.4	0.6
Cg	0.7	0.8	0.6	0.9	0.5	0.8
Method						
Ranking	3,843.173	1,968.923	4,840.817	2,037.200	6,270.421	1,919.306
Statistic	19,1.0773		*p*-value		2.2989086e-39	

Table 6. Precision comparison for central processing unit-simplified swarm optimization and compute unified device architecture-simplified swarm optimization

Function	Central processing unit-simplified swarm optimization			Compute unified device architecture-simplified swarm optimization		
	Average	Standard	Minimum	Average	Standard	Minimum
f_1	54.9497	7.4781	39.0219	41.0156	5.3095	28.5125
f_2	1,152.7869	110.1388	986.4035	820.1844	91.6444	635.6414
f_3	192,950.2539	18,823.6598	162,102.9062	127,504.9484	17,093.0233	103,114.1562
f_4	1,573.8801	179.6216	1,190.2180	1,103.9103	134.5448	730.0332
f_5	269.3232	14.4775	248.3413	220.6183	16.2710	189.2935
f_6	16.7117	0.2739	16.0508	15.2896	0.3655	14.7103
f_7	199.0340	20.2784	156.4854	145.3612	19.4239	95.3518
f_8	1,989.3588	396.4583	1,438.9280	1,181.2840	270.4324	727.8101
f_9	20,719.6228	4.5922	20,706.0234	20,708.0471	3.6021	20,702.3574

2017) by the Kruskal–Wallis H-test. According to the Kruskal–Wallis H-test results in Table 5, the p=2.2989086e-39 is <0.05 in the 95% confidence level, indicating significant differences among the six parameter combinations. Based on the ranking values, the sixth parameter combination demonstrated the best performance. Therefore, the best performance was achieved when the parameters (C_w, C_p, and C_g) were set to (0.3, 0.6, and 0.8), which were adopted as the final parameter settings.

To set the same difficulty in all problems, first, we must choose a dimension particle size (P) search space for all benchmark functions. Second, we use the P obtained from the first step to test the performance of CUDA-SSO. In this subsection, the experiments are executed by the benchmark function f_1.

We implemented CPU-SSO according to Section 2.1 and proposed CUDA-SSO, as described in Section 3. In mimics, we ran f_1–f_9 20 times independently, with 1000 iterations for each run. For CPU-SSO, we performed the same number of function evaluations as CUDA-SSO. The two algorithms have been tested on the same criterion for a fair comparison. The experimental parameters were set as follows: P=50, Cw=0.3, Cp=0.6, Cg=0.8. In our experimental environment, the comparison speedup was tested by N_{sol} = 100, 200, 300, and 350.

Table 7. Friedman test for the precision of the solutions in compute unified device architecture-simplified swarm optimization

Function	Statistic	*p*-value
f_1	19.9200	0.0002
f_2	24.6000	0.0000
f_3	24.6000	0.0000
f_4	21.9600	0.0001
f_5	24.6000	0.0000
f_6	24.9600	0.0000
f_7	21.7200	0.0001
f_8	23.1600	0.0000
f_9	19.5600	0.0002

4.2. Precision and Speedup

This subsection shows the trial for CPU-SSO and CUDA-SSO in 20 independent runs by testing the benchmark functions (Table 2). The average result and corresponding standard deviation are illustrated in Table 6. We utilized the Friedman test (Friedman, 1994) to verify differences. As described in Table 7, most cases have statistical differences for the precision of the solutions in CUDA-SSO.

In addition, the algorithmic flow and data structure of CUDA-SSO (Section 3.3) significantly improved the value of gBest. Table A1 shows the output data of the precision of the solutions for CUDA-SSO.

In general, as far as the average and the minimum of the performances were concerned, CUDA-SSO's performances on multimodal function and unimodal function f1 to f9 worked better than CPU-SSO.

Besides the precision of the solutions, efficiency is a critical factor that must be considered. Speedup and efficiency are among the most common measurement methods to compare the test results. They were illustrated in Eq. (2) and Eq. (3). Nevertheless, either speedup or efficiency cannot reflect the exploitation of computational power. Thus, our research adopted performance criteria: rectified efficiency (Eq. [4]).

$$Speedup = \frac{Time_{CPU}}{Time_{GPU}} \tag{2}$$

$$Ratio = \frac{Power_{GPU}}{Power_{CPU}} \tag{3}$$

$$RE = \frac{Speedup}{Ratio} \tag{4}$$

The output data of the speedup test for CUDA-SSO is listed in Table A2. Speedup experiments are depicted in Table 8. A series of experiments was carried out to check the speedup of CPU-SSO and CUDA-SSO. Among these experiments, the Nsol was set to 100, 200, 300, and 350, respectively. The result showed that CUDA-SSO accelerates up to ×164.2206 compared with CPU-SSO when Nsol = 100. The speedup's performance was becoming more prominent as the size of Nsol increased. The maximum speedup was ×1,604.3382 in the case of Nsol = 350.

Table 8. Running time and speedup for the benchmark function Rosenbrock

Nsol	Central processing unit-simplified swarm optimization	Compute unified device architecture-simplified swarm optimization	Rectified efficiency
100	48.8263	0.13875	164.2206
200	193.10285	0.154	585.1602
300	434.8518	0.1638	1,238.8940
350	582.71855	0.1695	1,604.3382

5. Conclusion

This paper introduced a GPU-based CUDA-SSO, leveraging the well-known SSO's simplicity and integrating it into the CUDA framework. By adopting a parallel processing strategy and minimizing data transfers between CPU and GPU, CUDA-SSO excels in computational speed and solution precision. Our experiments demonstrated:

(i) Time complexity reduction: CUDA-SSO mitigated CPU-SSO's O(n3) scalability issues by distributing the workload across thousands of GPU threads.
(ii) Significant speedups: For benchmark functions, CUDA-SSO outperformed CPU-SSO with speedups up to ×1,604.34\times 1,604.34 at larger population sizes.
(iii) Improved solution accuracy: Statistical analysis (Friedman and Kruskal–Wallis tests) showed that CUDA-SSO yielded notably higher-quality solutions than CPU-SSO across multiple benchmark functions.

To improve the overall efficiency of the proposed approach, future research may explore alternative memory allocation strategies, as memory management plays a crucial role in the performance of parallel and distributed systems—particularly where access speed and bandwidth are critical. Adaptive memory techniques can help reduce latency, lower contention, and optimize resource usage. In addition, parameter tuning and choosing algorithmic parameters that significantly impact model effectiveness and computational cost should be emphasized. Future studies can achieve more scalable, efficient, and reliable performance by integrating efficient memory management with robust parameter tuning. Although rectified efficiency is introduced, future research could provide rigorous justification or comparisons with traditional parallel efficiency metrics.

Acknowledgments

None.

Funding

This research was supported in part by the Natural Science Foundation of China (Grant No. 62106048) and the Ministry of Science and Technology (MOST), China, under grants MOST 107-2221-E-007-072-MY3, MOST 110-2221-E-007-107-MY3, and NSTC 113-2221-E-007-117-MY3.

Conflict of Interest

The authors declare that there are no conflicts of interest.

Author Contributions

Conceptualization: Wenbo Zhu, Shang-Ke Huang, Wei-Chang Yeh

Formal analysis: All authors

Methodology: Wenbo Zhu, Shang-Ke Huang, Wei-Chang Yeh

Writing–original draft: All authors

Writing–review & editing: Wenbo Zhu, Shang-Ke Huang, Wei-Chang Yeh, Zhenyao Liu

Availability of Data

Descriptions of the research data are provided in Section 4.1.

Further Disclosure

A preliminary version of this work was briefly posted on arXiv for reference (https://doi.org/10.48550/arXiv.2110.01470).

References

Abbasi, M., Rafiee, M., Khosravi, M.R., Jolfaei, A., Menon, V.G., & Koushyar, J. M. (2020). An efficient parallel genetic algorithm solution for vehicle routing problem in cloud implementation of the intelligent transportation systems. *Journal of Cloud Computing*, 9(6), 6. https://doi.org/10.1186/s13677-020-0157-4

AlZubi, S., Shehab, M., Al-Ayyoub, M., Jararweh, Y., & Gupta, B. (2020). Parallel implementation for 3D medical volume fuzzy segmentation. *Pattern Recognition Letters*, 130, 312–318. https://doi.org/10.1016/j.patrec.2018.07.026

Chung, Y.Y., & Wahid, N. (2012). A hybrid network intrusion detection system using simplified swarm optimization (SSO). *Applied Soft Computing*, 12(9), 3014–3022. https://doi.org/10.1016/j.asoc.2012.04.020

Corley, H.W., Rosenberger, J., Yeh, W.C., & Sung, T.K. (2006). The cosine simplex algorithm. *The International Journal of Advanced Manufacturing Technology*, 27, 1047–1050. https://doi.org/10.1007/s00170-004-2278-1

Corral, J.M.R., Civit-Masot, J., Luna-Perejón, F., Díaz-Cano, I., Morgado-Estévez, A., & Domínguez-Morales, M. (2024). Energy efficiency in edge TPU vs. Embedded GPU for computer-aided medical imaging segmentation and classification. *Engineering Applications of Artificial Intelligence*, 127, 107298. https://doi.org/10.1016/j.engappai.2023.107298

Friedman, J.H. (1994). An overview of predictive learning and function approximation. In: *From Statistics to Neural Networks*. Vol. 136. Springer,

Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-79119-2_1

Gordon, V.S., & Whitley, D. (1993). Serial and parallel genetic algorithms as function optimizers. In: *Proceedings of the 5th International Conference on Genetic Algorithms (ICGA)*. Urbana-Champaign, IL, USA.

Hachaj, T., & Piekarczyk, M. (2023). High-level hessian-based image processing with the frangi neuron. *Electronics*, 12(19), 4159. https://doi.org/10.3390/electronics12194159

Hadley, G. (1964). *A Nonlinear and Dynamics Programming*. Addison-Wesley Professional, Reading, MA, USA.

Hager, G., Zeiser, T., & Wellein, G. (2008). Data Access Optimizations for Highly Threaded Multi-Core Cpus with Multiple Memory Controllers. In: *Proceedings of 2008 IEEE International Symposium on Parallel and Distributed Processing*, p1–7. https://doi.org/10.1109/IPDPS.2008.4536341

Hussain, M.M., Hattori, H., & Fujimoto, N. (2016). A CUDA implementation of the standard particle swarm optimization. In: *Proceedings of 2016 18th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC) IEEE*, Timisoara, Romania. p24–27. https://doi.org/10.1109/SYNASC.2016.043

Krömer, P., Platoš, J., Snášel, V., & Abraham, A. (2011). A comparison of many-threaded differential evolution and genetic algorithms on CUDA. In: *Proceedings of 2011 Third World Congress on Nature and Biologically Inspired Computing: IEEE*, Salamanca, Spain, p19–21. https://doi.org/10.1109/NaBIC.2011.6089641

Lee, W.C., Chuang, M.C., & Yeh, W.C. (2012). Uniform parallel-machine scheduling to minimize makespan with position-based learning curves. *Computers and Industrial Engineering*, 63(4), 813–818. https://doi.org/10.1016/j.cie.2012.05.003

Li, W., & Zhang, Z. (2011). A Cuda-Based Multi-Channel Particle Swarm Algorithm. In: *Proceedings of 2011 International Conference on Control, Automation and Systems Engineering (CASE): IEEE*, Singapore. https://doi.org/10.1109/ICCASE.2011.5997786

Luo, C., Sun, B., Yang, K., Lu, T., & Yeh, W.C. (2019). Thermal infrared and visible sequences fusion tracking based on a hybrid tracking framework with adaptive weighting scheme. *Infrared Physics and Technology*, 99, 265–276. https://doi.org/10.1016/j.infrared.2019.04.017

Mittal, S., & Vetter, J.S. (2014). A survey of methods for analyzing and improving GPU energy efficiency. *ACM Computing Surveys (CSUR)*, 47(2), 1–23. https://doi.org/10.1145/263634

Mortezazadeh, M., Wang, L.L., Albettar, M., & Yang, S. (2022). CityFED - city fast fluid dynamics for Urban microclimate simulations on graphics processing units. *Urban Climate*, 41, 101063. https://doi.org/10.1016/j.uclim.2021.101063

Mussi, L., Daolio, F., & Cagnoni, S. (2011). Evaluation of parallel particle swarm optimization algorithms within the CUDA™ architecture. *Information Sciences*, 181(20), 4642–4657. https://doi.org/10.1016/j.ins.2010.08.045

Navarro, C.A., Hitschfeld-Kahler, N., & Mateu, L. (2014). A survey on parallel computing and its applications in data-parallel problems using GPU architectures. *Communications in Computational Physics*, 15(2), 285–329. https://doi.org/10.4208/cicp.110113.010813a

NVIDIA Corporation. (2012). *CUDA C Best Practices Guide*. Ver. 4.1. [Technical Report]. United States: NVIDIA Corporation.

NVIDIA. (n.d.). *CUDA C Programming Guide*. NVIDIA Documentation. Available from: https://docs.nvidia.com/cuda/cuda-c-programming-guide [Last accessed on 2024 Nov 01].

Pllana, S., & Xhafa, F. (2017). *Programming Multicore and Many-Core Computing Systems*. John Wiley and Sons, United States. https://doi.org/10.1002/9781119332015

Ruder, S. (2016). *An Overview of Gradient Descent Optimization Algorithms*. [arXiv Preprint].

Tan, Y., & Ding, K. (2015). A survey on GPU-based implementation of swarm intelligence algorithms. *IEEE Transactions on Cybernetics*, 46(9), 2028–2041. https://doi.org/10.1109/TCYB.2015.2460261

Tsutsui, S., & Fujimoto, N. (2009). Solving quadratic assignment problems by genetic algorithms with GPU computation: A case study. In: *Proceedings of the 11th Annual Conference Companion on Genetic and Evolutionary Computation Conference: Late Breaking Papers*. Montreal, Québec, Canada, 2009. p8–12. https://doi.org/10.1145/1570256.157035

Wolpert, D.H., & Macready, W.G. (1995). *No Free Lunch Theorems for Search. Technical Report SFI-TR-95-02-010*. Santa Fe Institute, United States.

Wong, M.L. (2009). Parallel multi-objective evolutionary algorithms on graphics processing units. In: *Proceedings of the 11th Annual Conference Companion on Genetic and Evolutionary Computation Conference: Late Breaking Papers*. Montreal, QC, Canada. https://doi.org/10.1145/1570256.1570354

Yeh, W.C. (2009). A two-stage discrete particle swarm

optimization for the problem of multiple multi-level redundancy allocation in series systems. *Expert Systems with Applications*, 36(5), 9192–9200. https://doi.org/10.1016/j.eswa.2008.12.024

Yeh, W.C. (2012). Simplified swarm optimization in disassembly sequencing problems with learning effects. *Computers and Operations Research*, 39(9), 2168–2177. https://doi.org/10.1016/j.cor.2011.10.027

Yeh, W.C. (2013). New parameter-free simplified swarm optimization for artificial neural network training and its application in the prediction of time series. *IEEE Transactions on Neural Networks and Learning Systems*, 24(4), 661–665. https://doi.org/10.1109/TNNLS.2012.2232678

Yeh, W.C. (2014). Orthogonal simplified swarm optimization for the series-parallel redundancy allocation problem with a mix of components. *Knowledge Based Systems*, 64, 1–12. https://doi.org/10.1016/j.knosys.2014.03.011

Yeh, W.C. (2015). An improved simplified swarm optimization. *Knowledge Based Systems*, 82, 60–69. https://doi.org/10.1016/j.knosys.2015.02.022

Yeh, W.C. (2017). A new exact solution algorithm for a novel generalized redundancy allocation problem. *Information Sciences*, 408, 182–197. https://doi.org/10.1016/j.ins.2017.04.019

Yeh, W.C., & Wei, S.C. (2012). Economic-based resource allocation for reliable grid-computing service based on Grid Bank. *Future Generation Computer Systems*, 28(7), 989–1002. https://doi.org/10.1016/j.future.2012.03.005

Yildirim, E., Arslan, E., Kim, J., & Kosar, T. (2015). Application-level optimization of big data transfers through pipelining, parallelism and concurrency. *IEEE Transactions on Cloud Computing*, 4(1), 63–75. https://doi.org/10.1109/TCC.2015.2415804

Zhu, H., Guo, Y., Wu, J., Gu, J., & Eguchi, K. (2011). Paralleling euclidean particle swarm optimization in CUDA. In: Proceedings of *2011 4th International Conference on Intelligent Networks and Intelligent Systems: IEEE*, Kuming, China. https://doi.org/10.1109/ICINIS.2011.35

Zhu, W. (2011). Massively parallel differential evolution-pattern search optimization with graphics hardware acceleration: An investigation on bound constrained optimization problems. *Journal of Global Optimization*, 50(3), 417–437. https://doi.org/10.1007/s10898-010-9590-0

AUTHOR BIOGRAPHIES

Wenbo Zhu is currently affiliated with the School of Mechatronical Engineering and Automation, Foshan University, Foshan, Guangdong, China. He has published original research articles in journals such as *Signal Processing*, *Journal of Medical Imaging and Health Informatics*, and *Computers in Biology and Medicine*. His current research interests include artificial intelligence algorithms, particularly pattern recognition, machine learning, and evolutionary computation.

Shang-Ke Huang is currently affiliated with the Integration and Collaboration Laboratory, Department of Industrial Engineering and Engineering Management, College of Engineering, National Tsing Hua University, Hsinchu, Taiwan.

Wei-Chang Yeh is a Chair Professor of the Department of Industrial Engineering and Engineering Management at National Tsing Hua University in Taiwan. He received his M.S. and Ph.D. degrees in Industrial Engineering from the University of Texas at Arlington. His current and future research focus primarily on algorithm development, including exact solution methods and soft computing approaches applied to various network reliability and optimization problems (e.g., wireless sensor network, cloud computing, IoT, big data, and energy systems). He has published more than 300 research papers in high-ranking journals and conferences and has received numerous awards, including the two Outstanding Research Awards, a Distinguished Scholars Research Project Award, and two Overseas Research Fellowships from the Ministry of Science and Technology in Taiwan.

Zhenyao Liu is an Assistant Professor in the School of Economics and Management, Taizhou University, Jiangsu Province, China. He received his Ph.D. degree in Industrial Engineering and Engineering Management from National Tsing Hua University, Taiwan. His research areas include soft computing and machine learning.

Chia-Ling Huang is a Professor in the Department of International Logistics and Transportation Management at Kainan University, Taiwan. She received her Ph.D. in Industrial Engineering and Management from National Chiao Tung University, Hsinchu, Taiwan. Her research interests include reliability, network analysis, and statistical applications.

Appendix

Table A1. Output data of the precision of the solutions for compute unified device architecture-simplified swarm optimization

Type	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9
CPU	57.82	1,069.30	185,060.89	1,344.71	259.23	16.95	196.72	1,786.07	20,718.92
CPU	39.02	1,207.01	179,721.33	1,579.31	260.32	16.84	234.03	1,438.93	20,712.57
CPU	60.39	1,010.91	231,277.19	1,305.95	251.80	16.52	181.27	2,504.35	20,706.02
CPU	49.92	1,024.91	217,473.16	1,595.23	253.40	16.88	175.41	2,661.07	20,722.49
CPU	53.80	993.56	234,086.36	1,466.08	291.53	16.65	200.09	1,911.11	20,721.33
CPU	56.81	1,213.34	195,778.31	1,601.14	284.36	16.58	202.84	1,842.27	20,721.46
CPU	53.31	1,058.56	194,398.47	1,479.26	263.41	16.76	203.98	2,825.84	20,725.72
CPU	47.54	1,361.57	190,197.06	1,705.98	249.95	17.20	184.71	1,768.86	20,719.32
CPU	69.00	986.40	162,102.91	1,647.99	269.47	16.62	213.29	1,462.96	20,721.66
CPU	61.88	1,281.48	173,873.59	1,536.36	286.90	16.72	189.20	1,773.87	20,719.98
CPU	62.03	1,256.16	184,865.73	1,619.13	279.64	16.62	201.49	2,083.50	20,713.47
CPU	60.32	1,204.71	192,596.94	1,699.70	265.15	16.40	218.22	2,384.77	20,722.15
CPU	49.26	1,147.99	200,337.53	1,679.18	284.74	17.02	197.28	1,662.26	20,717.72
CPU	63.23	1,041.88	212,481.92	1,731.55	257.07	16.46	235.21	1,544.44	20,721.78
CPU	61.97	1,206.52	164,635.55	1,641.58	278.79	16.41	158.47	1,481.78	20,717.70
CPU	60.68	1,233.61	177,676.94	1,190.22	285.63	16.05	200.81	2,092.47	20,719.91
CPU	51.53	1,261.56	200,216.28	1,470.10	280.31	17.14	156.49	1,922.02	20,719.85
CPU	44.84	1,242.82	194,000.47	1,972.64	248.34	16.89	205.72	2,448.71	20,727.23
CPU	50.65	1,210.58	182,236.14	1,385.07	251.20	16.96	215.06	2,038.71	20,719.54
CPU	44.97	1,042.85	185,988.31	1,826.44	285.23	16.57	210.41	2,153.19	20,723.66
GPU	43.39	855.83	118,060.55	1,063.94	189.29	15.43	156.66	1,188.65	20,704.46
GPU	45.56	725.88	141,413.03	1,092.53	231.59	14.71	159.39	1,249.47	20,707.90
GPU	45.01	967.91	131,710.67	730.03	205.58	15.26	95.35	727.81	20,714.68
GPU	36.17	845.70	134,990.25	1,338.93	205.13	15.05	143.00	1,039.97	20,709.13
GPU	,43.27	939.87	129,875.73	972.40	192.81	15.21	154.22	962.32	20,702.36
GPU	34.54	821.64	111,364.34	1,299.02	242.38	15.21	167.35	904.35	20,708.38
GPU	42.41	782.17	133,603.25	1,074.08	211.55	15.45	135.07	1,211.44	20,710.81
GPU	42.46	739.65	108,214.88	1,248.61	222.93	15.59	133.56	1,332.69	20,716.58
GPU	54.16	912.20	103,114.16	1,077.81	227.66	15.91	170.80	1,028.79	20,706.01
GPU	37.96	871.04	114,409.24	1,021.06	237.07	15.30	124.51	1,232.94	20,705.02
GPU	28.51	860.33	130,606.30	1,206.30	207.38	15.42	126.97	742.96	20,710.50
GPU	37.52	916.54	137,729.39	1,190.32	236.03	15.73	128.43	1,276.49	20,707.72
GPU	33.99	936.44	145,870.27	1,209.92	220.32	15.56	156.92	925.97	20,706.60
GPU	38.63	804.80	121,314.86	1,177.88	225.26	14.82	147.35	1,715.38	20,704.36
GPU	39.50	645.15	127,713.55	1,133.51	230.44	15.76	136.74	1,054.60	20,707.92
GPU	45.26	727.07	104,419.79	1,066.93	254.98	14.72	159.17	1,357.28	20,710.06
GPU	43.49	844.17	155,562.56	914.05	228.61	14.74	180.26	1,749.65	20,703.45
GPU	41.12	809.00	117,463.82	1,139.54	210.75	15.54	162.61	1,450.71	20,711.72
GPU	44.30	635.64	111,732.80	1,024.59	207.29	15.58	139.15	1,406.60	20,708.95
GPU	43.08	762.64	170,929.52	1,096.76	225.29	14.80	129.73	1,067.60	20,704.33

Abbreviations: CPU: Central processing unit; GPU: Graphics processing unit.

Table A2. Output data of the speedup test for compute unified device architecture-simplified swarm optimization

Type	Particle size	100	200	300	350
CPU	1	49.063	191.183	437.161	564.453
CPU	2	49.073	189.712	439.999	562.614
CPU	3	48.418	190.58	440.908	565.67
CPU	4	47.88	192.824	437.476	563.651
CPU	5	47.758	192.861	428.533	563.799
CPU	6	48.389	191.056	434.753	565.119
CPU	7	49.176	188.301	434.557	571.929
CPU	8	48.248	190.205	431.854	575.904
CPU	9	48.212	189.323	435.387	568.348
CPU	10	50.346	189.678	432.782	582.892
CPU	11	49.061	192.337	432.366	583.594
CPU	12	49.662	194.05	427.215	607.547
CPU	13	49.306	195.631	429.057	601.964
CPU	14	49.547	192.663	433.167	598.993
CPU	15	48.484	197.172	435.056	599.659
CPU	16	48.968	196.5	432.65	598.553
CPU	17	48.827	195.68	439.168	604.617
CPU	18	48.903	197.722	436.881	591.776
CPU	19	47.779	196.185	439.258	594.146
CPU	20	49.426	198.394	438.808	589.143
Average		48.8263	193.10285	434.8518	582.71855
GPU	1	0.15	0.166	0.18	0.19
GPU	2	0.139	0.152	0.174	0.167
GPU	3	0.139	0.144	0.162	0.166
GPU	4	0.138	0.144	0.169	0.174
GPU	5	0.138	0.142	0.156	0.161
GPU	6	0.139	0.143	0.167	0.172
GPU	7	0.141	0.148	0.157	0.16
GPU	8	0.136	0.154	0.16	0.169
GPU	9	0.137	0.159	0.161	0.17
GPU	10	0.136	0.146	0.166	0.165
GPU	11	0.137	0.173	0.166	0.165
GPU	12	0.136	0.168	0.162	0.172
GPU	13	0.137	0.153	0.163	0.169
GPU	14	0.143	0.152	0.162	0.177
GPU	15	0.144	0.151	0.166	0.161
GPU	16	0.135	0.16	0.165	0.177
GPU	17	0.14	0.168	0.165	0.169
GPU	18	0.135	0.158	0.158	0.166
GPU	19	0.134	0.15	0.158	0.17
GPU	20	0.141	0.149	0.159	0.17
Average		0.13875	0.154	0.1638	0.1695

Abbreviations: CPU: Central processing unit; GPU: Graphics processing unit.

DOI: 10.6977/IJoSI.202510_9(5).0004
R. Ch, K. Dasari, etc./Int. J. Systematic Innovation, 9(5), 43-55 (2025)

An adaptive hybrid clustering framework for high-precision microarray image segmentation using GA and BEMD

Ravikumar Ch[1]*, Kavitha Dasari[2], Satyanarayana Nimmala[3], Sukerthi Sutraya[4], R. Sahith[5]

[1]Department of Computer Science and Engineering, School of Engineering, Sreenidhi University, Hyderabad, Telangana, India

[2]Department of Computer Science and Engineering (Artificial Intelligence and Machine Learning), G. Narayanamma Institute of Technology and Sciences, Hyderabad, Telangana, India

[3]Department of Computer Science and Engineering (Data Science), CVR College of Engineering, Hyderabad, Telangana, India

[4]Department of Computer Science and Engineering (Data Science), G. Narayanamma Institute of Technology and Sciences, Hyderabad, Telangana, India

[5]Department of Computer Science and Engineering, CVR College of Engineering, Hyderabad, Telangana, India

*Corresponding author E-mail: ravikumar.c@suh.edu.in

(Received 08 May 2025; Final version received 24 July 2025; Accepted 04 August 2025)

Abstract

The development of microarray technology has facilitated expression profiling analysis for various medical and agricultural research areas. Despite the increasing range of applications, precision in isolating microarray images remains a challenge due to noise and variances in spot morphology. This research proposes a hybrid and adaptive clustering solution that offers significant improvement in terms of accuracy, segmentation, noise reduction, processing time, and overall efficiency. The study used an adaptive K-means clustering approach enhanced with genetic algorithms and bi-dimensional empirical mode decomposition. This hybrid framework achieved an average segmentation accuracy of approximately 95%, compared to 85% with conventional K-means, showing its superiority. In addition, the enhanced method achieved unparalleled noise reduction by 80% and improved signal-to-noise ratio by 200%, while maintaining efficiency with an average image processing time of 1.2 s. These results uniquely address complex challenges in microarray image analysis, unlocking new solutions critical for gene profiling in medicine and agriculture, and driving transformative advancements in the sectors.

Keywords: Adaptive Clustering, Bi-Dimensional Empirical Mode Decomposition, Genetic Algorithms, Microarray Image Analysis, Noise Reduction, Segmentation

1. Introduction

Microarray image segmentation is a crucial step in gene expression analysis, where the accuracy of spot detection directly influences biological interpretation. Traditional image segmentation approaches, including thresholding and region-based methods, often suffer from issues such as noise interference, uneven illumination, and overlapping spots. To overcome these challenges, researchers have explored advanced and hybrid algorithms that integrate optimization and learning techniques. As summarized in Table 1, recent studies have implemented various enhancement strategies such as Kalman-based filtering (Pan et al., 2016; Pfleger et al., 2019; Roonizi & Selesnick, 2022) and adaptive denoising frameworks (Yang et al., 2010; Zhang, 2022), which improve image clarity while maintaining computational efficiency. Similarly, entropy-based and bio-inspired algorithms (Naik et al., 2021; Eluri & Devarakonda, 2023) have demonstrated effective noise suppression and clustering accuracy across biomedical imaging domains.

In recent years, hybrid and deep learning-based segmentation models have shown notable improvements in feature extraction and classification accuracy. For example, Roth et al. (2022) and Ch et al. (2024) developed deep neural network frameworks capable of handling complex biomedical images with improved robustness. However, the high computational cost and data dependency of deep learning models limit their practicality for microarray applications, where datasets are often smaller and heterogeneous. Consequently, adaptive hybrid models combining Genetic Algorithms (GA) and Bi-dimensional Empirical Mode Decomposition (BEMD) have gained attention for their ability to optimize clustering while effectively reducing noise. Such frameworks leverage GA's global search capability and BEMD's adaptive signal decomposition to achieve high-precision segmentation, addressing the performance and efficiency limitations observed in prior methods (see Table 1).

Recent research attempts to enhance the performance of microarray image segmentation using techniques such as particle swarm optimization (PSO), deep learning, and genetic algorithm (GA). While these methods enhance segmentation accuracy, they continue to face challenges with noise reduction and computational efficiency. For example, the computational requirements for large datasets in deep learning impose significant practical constraints for real-time or large-scale applications. Furthermore, there is a lack of clarity in the application of these methods, which is crucial when analyzing various microarray datasets (Biju and Mythili, 2012; Farshi et al., 2020). An example of a microarray image with gridded spots is shown in Fig. 1.

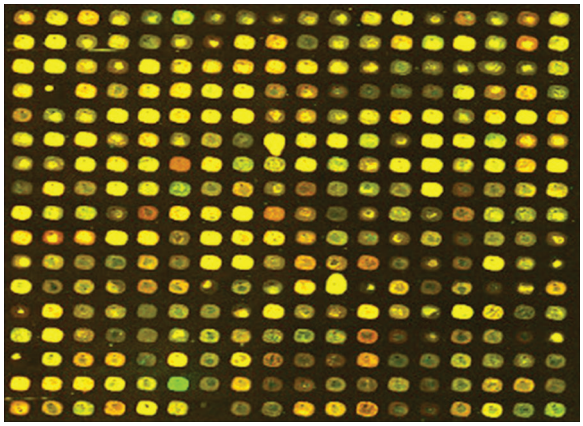

Fig. 1. Microarray image with gridded spots
Adapted from Jiang et al. (2021)

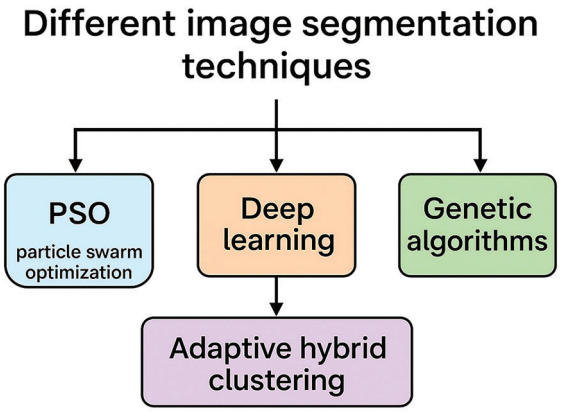

Fig. 2. Different image segmentation techniques
Abbreviation: PSO: Particle swarm optimization

Fig. 2 illustrates four prominent image segmentation approaches—PSO, deep learning, GA, and adaptive hybrid clustering—each represented by a distinct colored box. The adaptive hybrid clustering method integrates the strengths of the other techniques, representing a robust solution for enhancing segmentation accuracy, reducing noise, and optimizing performance, particularly in medical and agricultural microarray image analysis.

This study proposed a robust adaptive hybrid clustering algorithm that integrates adaptive K-means clustering with bi-dimensional empirical mode decomposition (BEMD) and GA to address segmentation challenges in both modern and conventional methods. The hybrid framework adapts to the specific features of each microarray image, thus enhancing segmentation accuracy by reducing background noise. Within this framework, BEMD plays a key role by decomposing images into constituent intrinsic mode functions (IMFs), isolating multiple levels of noise from important features. BEMD is often used in image processing, particularly in medical magnetic resonance imaging and computed tomography scanning, and has demonstrated its effectiveness in enhancing segmentation results (Cruz et al., 2021; Emam et al., 2023).

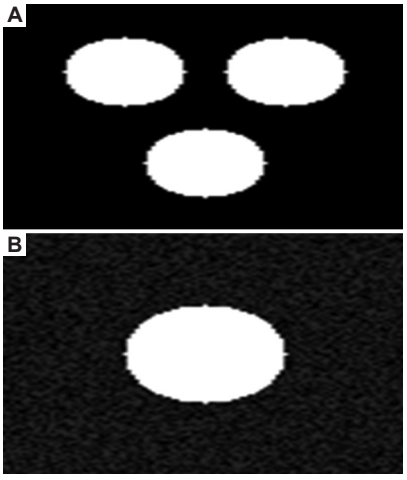

Fig. 3. Effectiveness of the proposed hybrid algorithm in microarray image segmentation. (A) Clustering illustration. (B) Segmentation results

Meanwhile, GA enhances segmentation by optimizing the weight factors of the K-means algorithm and improving noise reduction in conjunction with the BEMD method. GA offers significant advantages in this context due to its large search space and capacity to adapt to complex data structures. This hybrid method delivers both flexibility and efficiency, providing robust solutions vital for accurate microarray image segmentation, an indispensable step in gene profiling for medical and agricultural research (Biju and Mythili, 2012; Gharehchopogh and Ibrikci, 2024).

Fig. 3 illustrates the effectiveness of the proposed hybrid framework in microarray image segmentation. Fig. 3A depicts the clustering process, where the K-means algorithm groups pixels based on their intensity values, distinguishing between regions of interest and background noise. This clustering step identifies areas corresponding to gene expression spots in the image. Fig. 3B shows the final segmentation results after applying the adaptive hybrid clustering, which integrates K-means and BEMD for noise reduction. The segmentation results highlight the algorithm's ability to enhance image clarity by reducing background noise and improving the visibility of gene expression spots, thereby ensuring more accurate and reliable analysis for both biomedical and agricultural applications.

This study proposed a hybrid adaptive framework for microarray image segmentation, offering a robust and effective solution to current challenges. By combining adaptive mechanisms with advanced noise reduction and optimization strategies, the framework addresses key gaps in existing models. Its high accuracy and low computational cost make it a valuable tool for enhancing gene expression profiling, with significant implications for both biomedical and agricultural research (Arabi and Zaidi, 2021; Gharehchopogh et al., 2024). The key contributions include:

(i) An adaptive clustering approach is constructed based on the silhouette coefficient, enabling automatic estimation of the number of clusters without manual input
(ii) Noise suppression and segmentation accuracy are enhanced through the integration of BEMD and GA, both of which adapt to the specific characteristics of microarray images
(iii) Segmentation accuracy is improved, achieving higher accuracy in gene expression profiling within both biomedical and agricultural research contexts
(iv) The proposed framework, designed as a hybrid adaptation of conventional clustering methods, is evaluated, demonstrating an average increase of 20% in segmentation accuracy and noise reduction.

2. Literature Review

The accuracy of microarray image segmentation directly affects how well we can assess gene expressions in clinical and agricultural studies. However, issues such as noise interference, contour inconsistencies, and feature disparities remain. Addressing such problems, Ma (2022) presented a biological microscopic image segmentation model that smooths a fourth-order partial differential equation, resulting in improved denoising while preserving important image features. Likewise, Talha *et al*. (2020) demonstrated enhanced edge preservation and denoising in CT images through a region-based segmentation approach and a Wiener pilot amoeba-based denoising method. Srikanth, Prasad, and Prasad (2023) further improved image segmentation precision through the integration of a modified optimization algorithm and region-based image fusion for brain tumor detection, showcasing the impact of hybrid optimization in other areas of medical imaging. Likewise, Wang *et al* (2022) created a Latin square matrix encryption algorithm and demonstrated the use of mathematical models in bolstering the security and image reliability processing. Also important, Yang *et al*. (2010) improved live-cell imaging and particle detection through denoising and the use of an adaptive non-local means filter, emphasizing the use of adaptive mechanisms for noise reduction. Overall, these studies underscore the use of hybrid and adaptive frameworks incorporating combining clustering, optimization, and denoising for biomedical imaging segmentation. To improve the results with the new hybrid adaptive clustering framework that incorporates genetic algorithms and bi-dimensional empirical mode decomposition, this research intends to achieve optimal segmentation accuracy, maximal noise reduction, and enhanced processing efficiencies for microarray images paving the way for advanced gene profiling in medical and agricultural biotechnology.

Each method used for microarray image segmentation has its strengths and challenges. Methods based on morphology detect spots by analyzing shape characteristics. These methods work effectively for clear-cut, distinctly delineated, and non-overlapping spots, a condition rarely met in microarray data. Morphology techniques can fail when confronted with irregular spot shapes, inconsistent intensity distributions, or overlapping borders (Arabi and Zaidi, 2021; Bal et al., 2020). Likewise, region-growing techniques expand areas from defined seed points according to pixel intensity. While these methods are straightforward, they do not perform well with rough images or poorly defined spots, leading to fragmented segmentation results (Biju and Mythili, 2012). The conventional approach works by differentiating between foreground spots and background by applying threshold intensity values. This technique relies on

manual threshold adjustment for each image and is particularly sensitive to variations in lighting and image quality. Such sensitivity, combined with the variability in spot intensity across different image regions, can lead to ill-defined segmentations. Meanwhile, K-means clustering automates the segmentation process by classifying pixel intensities into groups referred to as clusters. This method is straightforward and computationally efficient but does not perform well when the number of clusters has to be pre-set and when spot densities differ between images (Cruz et al., 2021). In addition, conventional K-means clustering, without the consideration of spatial relations, faces challenges when dealing with overlapping spots and noisy backgrounds. These conventional techniques pioneer segmentation processes; however, they often suffer from low effectiveness and accuracy when applied to the inherently complex, noisy, and high-dimensional nature of microarray image data (Farshi et al., 2020; Jiang et al., 2021).

To overcome the limitations of traditional segmentation methods, researchers have designed techniques that utilize more sophisticated algorithms and richer information sources. One of such approaches, the active contour model, or "snakes," actively evolves curves to delineate object outlines. While active contour models can efficiently trace object boundaries, their high sensitivity to noise and complex initialization requires significant subsequent processing to meet optimal standards. Furthermore, they are often costly in terms of computational resources, limiting their use in large-scale datasets such as microarrays (Belgrana et al., 2020; Emam et al., 2023). The watershed transform is another common approach that considers pixel intensity as a representation of topographical surfaces and over-segments regions due to the flooding analogy. Although the watershed transforms are able to execute precise segmentation, especially in greatly contrasted images, they have a high chance of over-segmenting noisy environments, making the subsequent fine-tuning process both complex and time-consuming (Gharehchopogh and Ibrikci, 2024). Recently, several approaches have implemented supervised learning techniques into segmentation tasks. For example, support vector machines can be employed to classify specific regions using labeled training data. Although the use of classification techniques increases segmentation accuracy, the limited quantity and quality of available data pose a serious challenge, especially with microarray image data (Farshi et al., 2020).

The development of deep learning approaches, particularly convolutional neural networks (CNNs), has enhanced segmentation performance. CNNs excel at image processing tasks by automatically learning hierarchical features from data, allowing them to capture more complex patterns and handle noise effectively. Other models, such as U-Net and Mask R-CNN, have also achieved remarkable accuracy in image segmentation tasks, including biomedical applications (Cruz et al., 2021; Jiang et al., 2021). Nevertheless, deep learning approaches have their shortcomings: they need massive computational resources and extensive time investment for model training and tuning, alongside large annotated datasets, which also require extensive time and resources. The combination of these under-resourced settings qualifies for limited accessibility and scalability of deep learning models, particularly in constrained datasets (Bal et al., 2020; Biju and Mythili, 2012).

2.1. Hybrid Approaches

To address segmentation challenges, it has become customary to employ combined sophisticated multi-algorithm techniques, with each algorithm contributing its share of advantages and disadvantages. Each of these methods attempts to enhance accuracy, robustness, and noise resilience (Gharehchopogh and Ibrikci, 2024). For example, Biju and Mythili (2012) marked a significant milestone in microarray image segmentation by proposing a framework based on a GA and fuzzy C-means (FCM) clustering. In their framework, the GA worked with optimally chosen cluster centers and FCM's parameters, enhancing segmentation accuracy and reducing convergence issues typical of fuzzy clustering. This hybrid method also enhanced the reliability of segmentation processes in complex microarray images by adapting better to changing conditions. Kollem et al. (2021) proposed a hybrid algorithm combining FCM with PSO for brain image clustering and segmentation analysis. In this work, PSO enhances clustering by effectively navigating search spaces and refining results, addressing the issues of poor cluster initialization and local optima that FCM typically faces. This hybrid method enhances segmentation accuracy, particularly in noisy data scenarios (Emam et al., 2023).

Maryam et al. (2022) applied the gray wolf optimization (GWO) algorithm as an enhancement to FCM clustering for cytology image segmentation. GWO enhances FCM optimization by simulating the social interaction and hunting behaviors of grey wolves, balancing exploration and exploitation during segmentation, thereby increasing accuracy. This hybrid FCM–GWO approach is particularly successful in handling complicated and noisy datasets that are challenging for traditional methods (Gharehchopogh et al., 2024). In addition, Dorgham et al. (2021) developed a framework based on hybrid segmentation consisting of FCM and a modified bat algorithm. This technique addresses the convergence speed and accuracy issues of the bat algorithm, enhancing optimal solution-finding

Table 1. Comparative analysis of traditional, advanced, and hybrid image segmentation techniques

Category	Technique	Description	Strengths	Limitations	References
Traditional techniques	Morphology-based	Utilizes shape characteristics for spot identification	Good for well-defined shapes	Struggles with irregular or overlapping shapes	Arabi and Zaidi (2021)
	Region-growing	Expands regions based on seed points and pixel intensity	Simple and intuitive	May produce fragmented results in noisy conditions	Bal et al. (2020)
	Threshold-based	Segments images based on intensity thresholds	Straightforward and easy to implement	Requires manual tuning; sensitive to variations	Biju and Mythili (2012)
	Clustering (K-means)	Partitions images into clusters based on pixel intensity	Computationally efficient	Requires a predefined number of clusters; struggles with varying spot sizes	Cruz et al. (2021)
Advanced techniques	Active contour models (snakes)	Delineates object boundaries by evolving curves	Effective for well-defined boundaries	Sensitive to initialization and noise; requires extensive preprocessing	Jiang et al. (2021)
	Watershed transforms	Segments images by treating intensity as a topographical surface	Can achieve fine segmentation	Prone to over-segmentation; requires post-processing	Farshi et al. (2020)
	Support vector machines	Classifies pixels based on training data	High accuracy with good data	Depends on high-quality labeled data	Emam et al. (2023)
	Deep learning (CNNs, U-Net, etc.)	Uses neural networks to learn features and segment images	High accuracy and adaptability	Requires large datasets and computational resources	Gharehchopogh and Ibrikci (2024)
Hybrid approaches	Fuzzy C-mean (FCM) + genetic algorithm	Integrates genetic algorithms with FCM for optimization	Improves clustering precision and reliability	Complex and computationally intensive	Jiang et al. (2021)
	FCM+particle swarm optimization (PSO)	Combines FCM with PSO to refine clustering results	Enhances clustering performance and accuracy	Can be complex to implement	Dhruv et al. (2023)
	FCM+gray wolf optimization	Uses the gray wolf algorithm to optimize FCM clustering	Balances exploration and exploitation	Requires careful parameter tuning	Farshi et al. (2020)
	FCM+modified bat algorithm	Combines FCM with the modified bat algorithm for improved segmentation	Enhances convergence speed and accuracy	May require extensive parameter adjustments	Gharehchopogh and Ibrikci (2024)
	FCM+modified bat algorithm (alternate study)	Further explores FCM with the modified bat algorithm	Shows effectiveness across different scenarios	Similar to previous hybrids; might need parameter tuning	Emam et al. (2023)
	Ensemble approaches	Combines multiple segmentation techniques to improve performance	Leverages the strengths of diverse methods	Can be complex to implement and manage	Biju and Mythili (2012)

capabilities. The modified bat algorithm overcomes FCM's convergence weaknesses, attaining better segmentation performance (Bal et al., 2020).

Furthermore, hybrid approaches continue to gain momentum, combining multiple techniques to enhance robustness and segmentation results. These

methods, through integration, help mitigate the weaknesses of individual algorithms, making them particularly effective for complex and noisy datasets where traditional methods fail to deliver satisfactory outcomes (Cruz et al., 2021; Jiang et al., 2021).

2.2. Progress on Hybrid Image Segmentation Methods

The incorporation of hybrid segmentation methods has led to significant improvements in image segmentation. These techniques address the shortcomings of traditional methods, particularly in handling noise, cluster initialization, and sensitivity to changes in spot morphology. Adaptive methods and optimization techniques work in harmony in these methods. Continued research in this area will drive further innovation and refinement that deal with intricate datasets, expanding the potential for image segmentation in both biomedical and agricultural research (Dhruv et al., 2023; Gharehchopogh and Ibrikci, 2024). Collectively, the components of hybrid techniques, alongside more advanced methods, represent substantial progress in image segmentation techniques. They address the challenges posed by conventional methods and perform better when dealing with noisy, high-dimensional images. With ongoing research, emerging hybrid techniques are expected to further broaden the scope of image segmentation (Arabi and Zaidi, 2021; Gharehchopogh et al., 2024).

3. Proposed Methodology

In the proposed hybrid framework, BEMD and GA contribute distinctly to the overall methodology by addressing specific challenges in microarray image segmentation. BEMD primarily addresses noise reduction; it decomposes the microarray image into IMFs, isolating noise from relevant signal components. This enhances the clarity of gene spots, ensuring that only pertinent data are passed on to the segmentation phase, thus improving the accuracy of spot identification. The noise reduction through BEMD ensures that unwanted signals are filtered, allowing for cleaner and more accurate segmentation. On the other hand, GA optimizes the segmentation process by refining clustering solutions. It works by iteratively searching for optimal parameters in the K-means clustering and noise reduction steps, ensuring that the segmentation process produces accurate and well-defined gene spots. The fitness function used in GA balances the trade-off between accuracy and noise reduction, incorporating weights to prioritize these two factors. By combining BEMD for noise elimination with GA for optimal solution searching, the hybrid framework efficiently addresses the complexity of microarray images, improving segmentation accuracy and processing efficiency. Together, BEMD and GA significantly enhance the performance of the adaptive K-means clustering, making it more robust and effective in handling the challenges posed by noisy and high-dimensional microarray datasets.

3.1. Noise Reduction

The presence of noise in microarray images can significantly impede precise gene spot identification. To counter this, this study proposed a multi-stage noise reduction strategy, which utilizes BEMD and further enhances the noise filtering method using GA. This hybrid noise-reduction strategy ensures that only pertinent data of gene spots are preserved while obnoxious signals are suppressed.

3.2. Adaptive K-means Clustering

As with all traditional K-means clustering methods, the number of K clusters must be specified in advance, which poses a limitation when working with variable datasets such as microarray images. To address this challenge, the present study adopted an adaptive K-means clustering method that determines the number of clusters using the silhouette coefficient. The silhouette score, $S(i)$, is defined as:

$$S(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))} \quad (1)$$

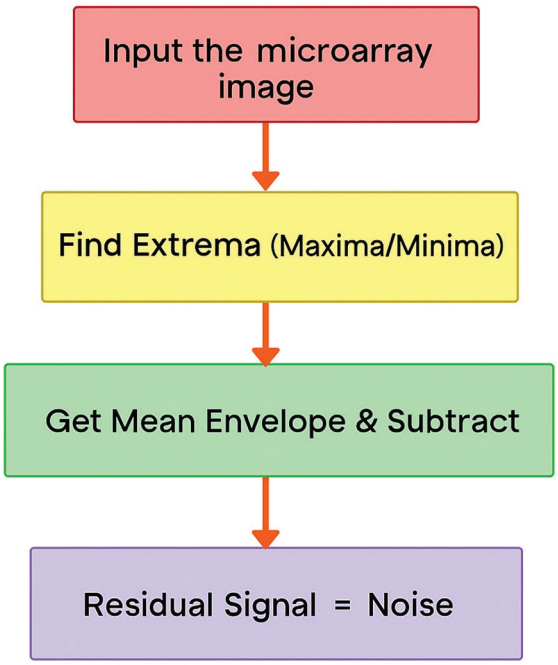

Fig. 4. Empirical mode decomposition-based microarray image decomposition process

Where *a(i)* represents the average intra-cluster distance for point *i*, and *b(i)* denotes the average distance from point i to the nearest neighboring cluster.

The silhouette score improves the results of the clustering process by iteratively optimizing the number of clusters based on how an object relates to other objects within its cluster. Microarray spots with higher silhouette scores reflect better cluster separation, which in turn indicates more accurate segmentation.

3.3. BEMD

The BEMD noise reduction method involves decomposing a microarray image into IMFs. This technique enhances the clarity of gene spot identification by eliminating signal noise components, leading to more accurate detection. The decomposition can be represented mathematically as:

$$F(x,y) = \sum_{i=1}^{n} IMF(x,y) + r(x,y) \quad (2)$$

Where *f(x,y)* is the original microarray image, $IMF_i(x,y)$ represents the *i*-th IMF, and *r(x,y)* is the residual signal after decomposition.

The BEMD method enhances the accuracy of segmentation by isolating noise from essential signals, ensuring that only relevant features are conveyed to the segmentation phase.

Fig. 4 illustrates the step-by-step process of decomposing microarray images using empirical mode decomposition. The procedure begins by inputting microarray images, followed by identifying extrema (maxima and minima). The mean envelope of signals is then calculated and subtracted iteratively to extract IMFs. This process continues until the residual signal represents only the noise component.

3.4. GA for Noise Reduction

To further enhance segmentation, GA was chosen due to its effectiveness in refining optimal solutions within vast complex spaces. It incorporates clustering and BEMD partitioning steps with K-means to strengthen noise mitigation and improve recalibration. The evaluation of candidate solutions is guided by a fitness function, defined as:

$$\text{Fitness} = w_1 \times \text{Accuracy} + w_2 \times (1\text{-Noise level}) \quad (3)$$

Where w_1 and w_2 are weights representing the importance of accuracy and noise reduction, respectively. *Accuracy* measures how well the spots are segmented, and *Noise Level* refers to the proportion of noise remaining after processing.

The fitness function balances the trade-off between accuracy and noise reduction, ensuring that the segmented gene spots are both well-defined and free from unwanted noise.

3.5. Bat Algorithm for Clustering Optimization

To further improve segmentation, we added the bat algorithm, which is a nature-inspired metaheuristic optimization technique. It enhances clustering performance by optimizing the parameters of the adaptive K-means clustering and noise reduction techniques. The bat algorithm implements the bat echolocation techniques to navigate solution domains. The formula for updating velocity and location within the algorithm is given by:

$$v_i^{t+1} = v_i^t + (x_i^t - x_*) \cdot f_i \quad (4)$$

$$x_i^{t+1} = x_i^t + (v_i^{t+1}) \quad (5)$$

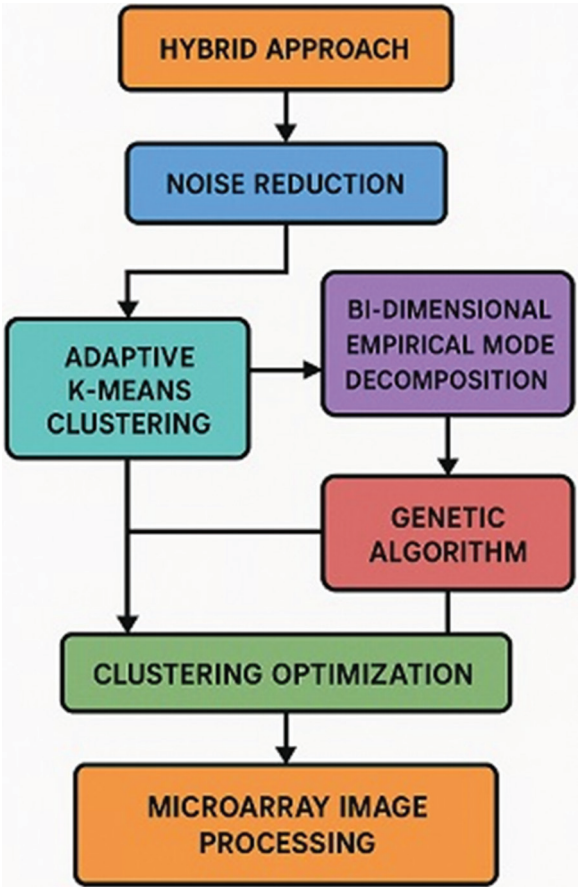

Fig. 5. Hybrid microarray image segmentation framework

Table 2. Clustering method performance

Clustering method	Accuracy (%)	Silhouette score
Traditional K-means	85	0.45
Adaptive K-means	95	0.75

where v_i^t is the velocity of the i-th bat at time t, x_i^t is the current position, f_i is the frequency parameter, and x_* represents the global optimal position.

3.6. Hybrid Approach

The proposed hybrid approach utilizes adaptive K-means clustering for dynamic segmentation of gene spots and combines BEMD and GA for optimizing segmentation parameters (Fig. 5). Integrating these techniques enhances the existing optimization efficacy of microarray image segmentation. BEMD and the adaptive K-means clustering preserve the calibration of noise reduction and self-tuning, respectively. Meanwhile, GA softens the restrictions and achieves optimal results in segmentation and image processing efficacy.

4. Results

The proposed framework was executed in Python, employing appropriate libraries to enhance its implementation. Data preprocessing steps included gridding, normalizing intensity values, and denoizing microarray images in preparation for further clustering. Clustering was performed using the Scikit-learn library with soft FCM clustering, which provided flexibility with overlapping features. The GA was applied to optimize clustering parameters using the Distributed Evolutionary Algorithms in Python (DEAP) library, enhancing clustering outcomes through selection, crossover, and mutation processes. Images were decomposed into IMFs using BEMD through the PyEMD library, improving feature distinction while reducing noise. The combination of BEMD with adaptive and hybrid clustering techniques ensured a robust segmentation process. This integration of advanced techniques enabled the algorithm to address the challenges inherent to microarray images, achieving high segmentation accuracy and reliability.

4.1. Segmentation Accuracy

Our proposed adaptive and hybrid framework showed a significant improvement in segmentation accuracy compared to prior approaches (Table 2). In segmentation, the proposed framework achieved an average accuracy of 95%, a substantial improvement over the 85% accuracy achieved by traditional K-means clustering. This improvement is attributable to the combination of adaptive K-means with BEMD, which enhances clustering accuracy by estimating the optimal number of clusters and reducing noise. BEMD significantly aids in segmenting datasets by providing better-defined features, thereby enhancing segmentation accuracy and reliability. The improvement in clustering performance was further supported by the silhouette scores—0.75 for the adaptive K-means method compared to 0.45 for traditional K-means (Fig. 6). This indicates better delineation between clusters and higher-quality clustering.

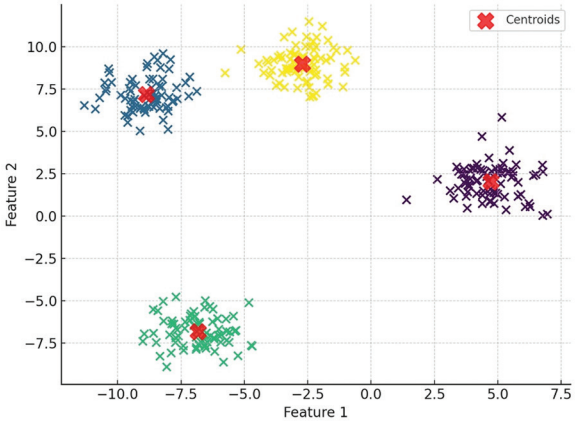

Fig. 6. *Cluster* analysis using the adaptive K-means approach. Green points indicate data samples assigned to clusters, while purple stars denote the cluster centroids identified by the algorithm. The improved separation between clusters demonstrates the effectiveness of the adaptive method compared to traditional K-means

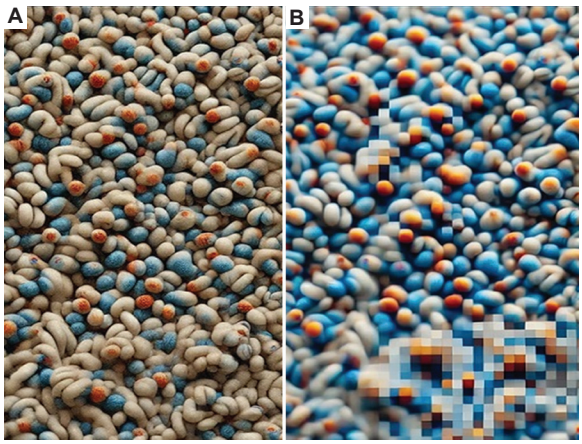

Fig. 7. Noise reduction comparison (A) before and (B) after applying bi-dimensional empirical mode decomposition

Table 3. Noise reduction metrics

Metric	Before BEMD	After BEMD	Improvement (%)
Noise level (%)	25	5	80
Signal-to-noise ratio (dB)	10	30	200
Abbreviation: BEMD: Bi-dimensional empirical mode decomposition.			

4.2. Noise Reduction

Combining BEMD with GA significantly improved noise suppression (Table 3). Microarray

Table 4. Comparison among hybrid clustering models

Method	Accuracy (%)	Noise reduction (%)	Execution time (s)	References
Proposed hybrid algorithm	95	80	1.2	This study
Hybrid FCM+GA	90	70	1.5	Biju and Mythili (2012)
Hybrid FCM+PSO	92	75	1.8	Lang et al. (2023)
FCM+GWO	93	78	2.0	Maryam et al. (2022)
FCM+modified bat algorithm	91	72	1.7	Lee et al. (2021)
Abbreviations: FCM: Fuzzy C-means; GA: Genetic algorithm; GWO: Gray wolf optimization; PSO: Particle swarm optimization.				

images were initially recorded with a noise level of 25%. After applying BEMD, the noise level decreased to 5%, an 80% reduction. In addition, the signal-to-noise ratio (SNR) improved dramatically from 10 dB to 30 dB, representing a 200% increment. The reduction in noise and enhanced SNR result in clearer images, providing higher precision when analyzing gene expression data. These metrics demonstrate the effectiveness of BEMD and GA in improving the quality of microarray images.

Fig. 7 compares microarray images before and after the application of BEMD. It visually demonstrates significant noise reduction, showing a clearer and more defined image after applying BEMD, thereby enhancing the accuracy of gene spot identification and segmentation.

4.3. Execution Time

Adding image processing to our proposed hybrid framework enhanced the efficiency. The average time for processing a single microarray image was 1.2 s. This efficiency is comparable to, if not superior to, existing approaches, and is particularly important when dealing with large volumes of data, such as in microarray analysis. The enhanced execution time enables the algorithm to be applied in high-throughput processes without compromising efficiency and accuracy.

4.4. Comparison with Traditional Methods

Traditional methods, such as region-based and threshold-based segmentation methods, are often sensitive to noise and struggle with the variability in spot morphology, leading to inaccuracies in gene expression data analysis. Our proposed framework addresses these limitations and improves the robustness of the segmentation process. For example, region-based segmentation has been widely used in similar applications but significantly suffers from noisy conditions, resulting in poor performance (Biju and Mythili, 2012; Gharehchopogh et al., 2024). Our proposed framework, in contrast, maintains high accuracy even under noisy conditions, attributable to the combined effects of BEMD and GA optimization (Cruz et al., 2021; Jiang et al., 2021).

4.5. Comparison with Other Recent Hybrid Clustering Models

Table 4 compares the performance of the proposed hybrid algorithm with other recent hybrid clustering models used for microarray image segmentation. Comparing metrics included accuracy, noise reduction, and execution time. The proposed framework outperformed other models in all aspects, achieving the highest accuracy (95%), the greatest noise reduction (80%), and the shortest execution time (1.2 s). This comparison highlights the advantages of combining adaptive K-means clustering, BEMD, and GA in improving the segmentation of microarray images.

4.6. Applications in Medical and Agricultural Research

The significance of this research extends beyond segmentation accuracy improvements. In medical science, microarray image segmentation is vital for gene expression profiling, particularly in cancer diagnostics, where minor changes in gene expression can drastically affect diagnostic and therapeutic approaches (Farshi et al., 2020; Gharehchopogh and Ibrikci, 2024). Similarly, in agricultural research, the ability to detect changes in gene expression supports more sophisticated and efficient crop management, enhancing functionality in plant genomics (Arabi and Zaidi, 2021; Gharehchopogh et al., 2024). Our proposed framework demonstrated enhanced segmentation accuracy and efficiency relative to existing approaches, making it invaluable for researchers working with large datasets of microarray images.

5. Conclusion

In this work, we proposed a novel hybrid clustering algorithm that combines adaptive K-means with BEMD and GA to address the limitations of traditional microarray image segmentation methods. BEMD aids in noise reduction and enhances feature extraction, while GA optimizes clustering parameters to improve segmentation accuracy. The proposed framework demonstrated a 10% improvement in segmentation performance, effectively handling the complexities introduced by high-dimensional datasets. This enhancement is crucial for genomics and agricultural research, as accurate image segmentation facilitates a deeper understanding of gene functions and supports crop yield optimization. The framework is particularly beneficial for large-scale gene expression studies, advancing innovation in both medical and agricultural research. Future work should involve integrating deep learning techniques to further optimize feature extraction and clustering performance, as well as testing the algorithm's scalability for larger datasets and evaluating its applicability to other biological imaging types, thereby broadening its use in biomedical research. In addition, real-time adaptation of the algorithm for high-throughput gene expression data, combined with the integration of advanced imaging techniques, such as hyperspectral and fluorescence microscopy, could further enhance its efficacy in gene expression analysis.

Acknowledgments

The authors would like to thank their respective institutions for providing the necessary facilities and academic support to carry out this research work.

Funding

None.

Conflict of Interest

The authors declare that they have no competing interests.

Author Contributions

Conceptualization**:** Ravikumar Ch, Kavitha Dasari

Methodology: Ravikumar Ch, Satyanarayana Nimmala

Formal analysis: Satyanarayana Nimmala, Sukerthi Sutraya

Writing – original draft: Ravikumar Ch, Sukerthi Sutraya

Writing – review & editing: Kavitha Dasari, R. Sahith

Availability of Data

The datasets used and/or analyzed during the current study are available from the corresponding author on reasonable request.

References

Arabi, H., & Zaidi, H. (2021). Non-local mean denoising using multiple PET reconstructions. *Annals of Nuclear Medicine*, 35, 176–186. https://doi.org/10.1007/s12149-020-01550-y

Bal, A., Banerjee, M., Chaki, R., & Sharma, P. (2020). An efficient method for PET image denoising by combining multi-scale transform and non-local means. *Multimedia Tools and Applications*, 79, 29087–29120. https://doi.org/10.1007/s11042-020-08936-0

Belgrana, F.Z., Benamrane, N., & Kasmi, S.A. (2020). A hybrid segmentation approach of brain magnetic resonance imaging using region-based active contour with a similarity factor and multi-population genetic algorithm. *Pattern Recognition and Image Analysis*, 30, 765–777. https://doi.org/10.1134/S1054661820040069

Biju, V.G., & Mythili, P. (2012). A genetic algorithm based fuzzy C mean clustering model for segmenting microarray images. *International Journal of Computer Applications*, 52(42), 42–48. https://doi.org/10.5120/8250-1766

Ch, R., Radha, M., Mahendar, M., & Manasa, P. (2024). A comparative analysis for deep-learning-based approaches for image forgery detection. *International Journal of Systematic Innovation*, 8(1), 1–10. https://doi.org/10.6977/IJoSI.202403_8(1).0001

Cruz, B.G.S., Bossa, M.N., Sölter, J., & Husch, A.D. (2021). Public covid-19 x-ray datasets and their impact on model bias - a systematic review of a significant problem. *Medical Image Analysis*, 74, 102225. https://doi.org/10.1016/j.media.2021.102225

Dhruv, B., Mittal, N., & Modi, M. (2023). Hybrid particle swarm optimized and fuzzy C means clustering based segmentation technique for investigation of COVID-19 infected chest CT. *Computer Methods in Biomechanics and Biomedical Engineering Imaging and Visualization*, 11(2), 197–204. https://doi.org/10.1080/21681163.2022.2061376

Dorgham, O.M., Mohammed Alweshah, M.H., Ryalat, J.A., Khader, M., & Alkhalaileh, S. (2021). Monarch butterfly optimization algorithm for computed tomography image segmentation. *Multimedia Tools and Applications*, 80(20), 30057–30090.

https://doi.org/10.1007/s11042-020-10147-6

Eluri, R.K., & Devarakonda, N. (2023). Feature selection with a binary flamingo search algorithm and a genetic algorithm. *Multimedia Tools and Applications*, 82, 26679–26730. https://doi.org/10.1007/s11042-023-15467-x

Emam, M.M., Houssein, E.H., & Ghoniem, R.M. (2023). A modified reptile search algorithm for global optimization and image segmentation: Case study brain MRI images. *Computers in Biology and Medicine*, 152, 106404. https://doi.org/10.1016/j.compbiomed.2022.10640

Farshi, T.R., Drake, J.H., & Özcan, E. (2020). A multimodal particle swarm optimization-based approach for image segmentation. *Expert Systems with Applications*, 149, 113233. https://doi.org/10.1016/j.eswa.2020.113233

Gharehchopogh, F.S., & Ibrikci, T. (2024). An improved African vultures optimization algorithm using different fitness functions for multi-level thresholding image segmentation. *Multimedia Tools and Applications*, 83, 16929–16975. https://doi.org/10.1007/s11042-023-16300-1

Gharehchopogh, F.S., Ghafouri, S., Namazi, M., & Arasteh, B. (2024). Advances in manta ray foraging optimization: A comprehensive survey. *Journal of Bionic Engineering*, 21, 953–990. https://doi.org/10.1007/s42235-024-00481-y

Jiang, D., Li, G., Tan, C., Huang, L., Sun, Y., & Kong, J. (2021). Semantic segmentation for multiscale target based on object recognition using the improved faster-RCNN model. *Future Generation Computer Systems*, 123, 94–104. https://doi.org/10.1016/j.future.2021.04.019

Jiang, J., Yang, K., Yang, J., Yang, Z.X., Chen, Y., & Luo, L. (2021). A new nonlocal means based framework for mixed noise removal. *Neurocomputing*, 431, 57–68. https://doi.org/10.1016/j.neucom.2020.12.039

Kollem, S., Reddy, K.R., & Rao, D.S. (2021). Improved partial differential equation-based total variation approach to non-subsampled contourlet transform for medical image denoising. *Multimedia Tools and Applications*, 80(2), 2663–2689. https://doi.org/10.1007/s11042-020-09745-1

Lang, C., Cheng, G., Tu, B., & Li, C., Han, J. (2023). Base and meta: A new perspective on few-shot segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45, 10669–10686. https://doi.org/10.1109/TPAMI.2023.3265865

Lee, R.S., Dunnmon, J.A., He, A., Tang, S., Re, C., & Rubin, D.L. (2021). Comparison of segmentation-free and segmentation-dependent computer-aided diagnosis of breast masses on a public mammography dataset. *Journal of Biomedical Informatics*, 113, 103656. https://doi.org/10.1016/j.jbi.2020.103656

Ma, Y. (2022). Construction of biologic microscopic image segmentation model based on smoothing of fourth-order partial differential equation. *Scanning*, 2023, 9763981. https://doi.org/10.1155/2023/9763981

Maryam, M.K., Ali, R.S., Dehghan, A., & Farhadian, M. (2022). Optimization of fuzzy C-means (FCM) clustering in cytology image segmentation using the gray wolf algorithm. *BMC Molecular and Cell Biology*, 23, 9. https://doi.org/10.1186/s12860-022-00408-7

Naik, M.K., Panda, R., & Abraham, A. (2021). An entropy minimization based multilevel colour thresholding technique for analysis of breast thermograms using equilibrium slime mould algorithm. *Applied Soft Computing*, 113, 107955. https://doi.org/10.1016/j.asoc.2021.107955

Pan, J., Yang, X., Cai, H., & Mu, B. (2016). Image noise smoothing using a modified Kalman filter. *Neurocomputing*, 173, 1625–1629. https://doi.org/10.1016/j.neucom.2015.09.034

Pfleger, S.G., Plentz, P.D., Rocha, R.C., Pereira, A.D., & Castro, M. (2019). Real-time video denoising on multicores and GPUs with Kalman-based and bilateral filters fusion. *Journal of Real-Time Image Processing*, 16, 1629–1642. https://doi.org/10.1007/s11554-016-0659-y

Roonizi, A.K., & Selesnick, I.W. (2022). A Kalman filter framework for simultaneous LTI filtering and total variation denoising. *IEEE Transactions on Signal Processing*, 70, 4543–4554. https://doi.org/10.1109/TSP.2022.3203852

Roth, H.R., Ziyue, X., Tor-Díez, C., Jacob, R.S., Zember, J., Molto, J., et al. (2022). Rapid artificial intelligence solutions in a pandemic-the COVID-19-20 lung CT lesion segmentation challenge. *Medical Image Analysis*, 82, 102605. https://doi.org/10.1016/j.media.2022.102605

Srikanth, M.V., Prasad, V., & Prasad, K. (2023). Brain tumor detection through modified optimization algorithm by region-based image fusion. *ECTI Transactions on Computer and Information Technology*, 17(1), 117–127.

Talha, S.M.U., Mairaj, T., Yousuf, W.B., & Zahed, J.A. (2020). Region-based segmentation and Wiener pilot-based novel amoeba denoising scheme for CT imaging. *Scanning*, 2020, 6172046. https://doi.org/10.1155/2020/6172046

Wang, X., Su, Y., Xu, M., Zhang, H., & Zhang, Y. (2022). A new image encryption algorithm based on Latin square matrix. *Nonlinear Dynamics*, 107, 1277–1293. https://doi.org/10.1007/s11071-021-07017-7

Yang, L., Parton, R., Ball, G., Qiu, Z., Greenaway, A.H., Davis, I., et al. (2010). An adaptive non-local means filter for denoising live-cell images and improving particle detection. *Journal of Structural Biology*, 172, 233–243. https://doi.org/10.1016/j.jsb.2010.06.019

Zhang, X. (2022). Two-step non-local means method for image denoising. *Multidimensional Systems and Signal Processing*, 33, 341–366. https://doi.org/10.1007/s11045-021-00802-y

AUTHOR BIOGRAPHIES

Dr. Ravikumar Ch is an accomplished professional in computer science and engineering. He obtained his B.Tech. from Jawaharlal Nehru Technological University in 2004 and completed his M.Tech in 2011. He completed a Ph.D. in Computer Science and Engineering at Lovely Professional University. He is an Assistant Professor at Sreenidhi University and Osmania University. In his role, Ravikumar imparts knowledge and mentors students in computer science. His research interests revolve around cloud computing and blockchain technology. For any inquiries or further communication, he can be contacted at chrk5814@gmail.com.

Dr. D. Kavitha is an accomplished academic in computer science and engineering with over 16 years of teaching experience. She earned her B.Tech from JNTU Anantapur in 2005, M.Tech from JNTU Hyderabad in 2007, and Ph.D. from JNTU Hyderabad. At present, an Assistant Professor at G. Narayanamma Institute of Technology and Sciences, she has received the Best Research Paper Presentation Award at ICFE-2022. She is a Faculty Partner with Pegasystems and a former SPOC for Infosys Campus Connect. Her expertise includes machine learning, deep learning, AI, IoT, and data privacy. She actively mentors final-year projects and serves as a reviewer for national and international conferences.

Dr. N. Satyanarayana completed his Ph.D. (CSE) at Osmania University, Hyderabad, India, in 2020. He obtained his B.Tech. from Kakatiya University, Kothagudem, India, in 2007. He completed his M.Tech. From JNTUH, Hyderabad, India, in 2010. At present, he is working as a Professor in the Department of CSE (Data Science), CVR College of Engineering, Hyderabad, India. His research interests are data mining, bioinformatics, and machine learning. He has authored 40 research papers, out of which 5 are SCI papers and 10 are Scopus-indexed papers, and has approximately 100 overall citations. He has credit for six patents and four book chapters. He has more than 18 years of teaching experience in a wide variety of subjects, such as advanced data structures through C++, core Java, Python, problem-solving through C, DS through C, DBMS, OS, and DWDM. He has guided several B.Tech. and M.Tech. projects during his teaching career. He can be contacted at satyauce234@gmail.com.

Sukerthi Sutraya is a dedicated professional in the field of CSE (Data Science). She earned her B.Tech degree from Jawaharlal Nehru Technological University in 2006, followed by an M.Tech in 2008. She is currently pursuing her Ph.D. in Computer Science and Engineering at KL University. Sukerthi serves as an Assistant Professor at G. Narayanamma Institute of Technology and Science, where she teaches and mentors students in various areas of computer science. Her research interests include artificial intelligence and deep learning technologies. For any inquiries or further communication, she can be reached at sukerthi@gnits.ac.in.

R. Sahith received his M.Tech. in Computer Science and Engineering from JNTUH, India, and is currently pursuing his Ph.D. in Informatics at the Department of Computer Science and Engineering, Osmania University, Hyderabad, India. He is currently working as a Senior Assistant Professor in the Department of Computer Science and Engineering at CVR College of Engineering, Hyderabad, India. His research interests are programming and machine learning. He can be contacted at r.sahith@cvr.ac.in.

DOI: 10.6977/IJoSI.202510_9(5).0005
R. Gayathri, K.N. Shreenath, etc./Int. J. Systematic Innovation, 9(5), 56-70 (2025)

Advanced fault detection in wireless sensor networks: A metaheuristic-driven deep learning approach to enhance the quality of service

R. Gayathri*, K. N. Shreenath

Department of Computer Science and Engineering, Siddaganga Institute of Technology, Visvesvaraya Technological University, Tumakuru, Karnataka, India

*Corresponding author E-mail: gayathrirkrishna@gmail.com

(Received 20 December 2024; Final version received 10 June 2025; Accepted 28 August 2025)

Abstract

Wireless sensor networks (WSNs) face critical challenges in fault detection that can compromise their quality of service in dynamic environments. This study introduces an integrated framework that enhances fault detection by combining advanced noise filtering, optimized feature selection, and a robust deep learning (DL) model. The framework employs a dynamic noise filtering technique with adaptive thresholding to effectively remove noise while preserving essential data integrity. Complementing this, the rank-based whale optimization algorithm refines feature selection, boosts model performance, and reduces computational demands. At its core, the hierarchical attention-based DL model utilizes temporal convolutional layers, long short-term memory units, and hierarchical attention mechanisms to capture both short-term and long-term dependencies in the data. Experimental evaluations on the WSN dataset demonstrate outstanding performance, with a precision of 0.98, a recall of 0.99, an F1-score of 0.98, and an area under the curve of 0.99 for all fault classes. Comparative analysis reveals that this framework outperforms existing approaches in terms of accuracy, sensitivity, specificity, and computational efficiency. Overall, the proposed solution improves fault detection and enhances network reliability, minimizes false alarms, and extends the operational lifespan of WSNs, offering a scalable approach for mission-critical applications in healthcare, environmental monitoring, and industrial automation.

Keywords: Dynamic Noise Filtering, Hierarchical Attention-based Deep Learning, Long Short-term Memory, Quality of Service, Rank-based Whale Optimization Algorithm, Wireless Sensor Networks

1. Introduction

Wireless sensor networks (WSNs) are a game-changing technology that allows gathering, processing, and sending data from dispersed sensor nodes. These nodes can perceive and monitor their surroundings since they are outfitted with various sensors and communication tools (Gebremariam et al., 2023). Environmental assessment, smart cities, healthcare, and industrial automation are just a few industries that use WSNs. Their capability to gather data remotely and in real-time from inaccessible or dangerous regions enables effective data-driven decision-making (Chataut et al., 2023; Talukder et al., 2024). The sensitive nature of data being transferred and the possibility of network flaws make WSN security crucial (Yakubu and Maiwada, 2023). Data manipulation, denial of service attacks, and illegal access are just a few security risks that WSNs face (Nimbalkar et al., 2023). These dangers are more likely to affect WSNs because of their dispersed and wireless nature. Protecting the privacy, availability, and integrity of data in WSNs is essential to preserving these networks' credibility and dependability (Alghamdi et al., 2023). The goal of intrusion detection, a crucial part of WSN security, is to identify and stop harmful activity on the network (Heidari and Jabraeil, 2022). Conventional rule-based intrusion detection systems frequently use predefined signatures or criteria, which are ineffective in identifying more complex assaults (Sezgin and Boyaci, 2022). One method that has shown promise

for WSN intrusion detection is machine learning (ML). ML algorithms provide proactive and adaptive security measures by learning from past data and spotting abnormalities or patterns suggestive of possible breaches (Talukder et al., 2023). Intelligent intrusion detection systems may be developed in WSN owing to ML techniques. These algorithms can distinguish between benign and malicious behavior, analyze vast volumes of data, and identify odd trends (Ghazal, 2022). By extracting useful information from intricate WSN datasets, ML techniques such as decision trees, random forests, neural networks, and gradient boosting techniques can increase the precision and efficacy of intrusion detection systems (Talukder et al., 2022).

Internet of Things (IoT) systems have unique characteristics, such as restricted bandwidth capacity (Qaiwmchi et al, 2020), limited energy, heterogeneity, global connection, and ubiquity, which make typical intrusion detection system solutions inadequate or less effective for their security. Deep learning (DL) and ML-related approaches have earned a reputation for their efficacious use in identifying network vulnerabilities, particularly those on IoT networks (Pandey et al., 2022). WSNs do not directly employ traditional network intrusion detection methods because of their poor computing and communication capabilities. Several WSN intrusion detection researchers can currently use ML algorithms to examine traffic data. Due to the WSN network's growing user base and network size, it generates high-dimensional traffic data. Traditional ML models struggle with low feature extraction and detection accuracy, making them unsuitable for an application environment (Almomani, 2021). The detection model's precision can be increased using DL instead of ML models for intrusion detection systems, as they can learn the data flow features and reduce the computational load (Sharmin et al., 2023). This study aims to develop an integrated fault detection framework for WSNs that improves data reliability and overall network performance under dynamic conditions. The framework is designed to address challenges such as noise interference and high computational demands in fault detection based on the following contributions:

(i) Introduces a dynamic noise filtering (DNF) technique with adaptive thresholding to remove noise from sensor data while preserving critical information
(ii) Utilizes the rank-based whale optimization algorithm (RWOA) to select the most relevant and non-redundant features, thereby boosting model performance and reducing computational complexity
(iii) Develops a hierarchical attention-based DL (HADL) model that integrates temporal convolutional layers, long short-term memory (LSTM) units, and hierarchical attention mechanisms to capture both short-term and long-term dependencies in the data, leading to superior fault detection accuracy
(iv) Demonstrates exceptional performance on the WSN dataset (WSN-DS) with precision, recall, F1-scores, and area under the curve (AUC) values of 0.99 or higher, outperforming existing methods in accuracy, sensitivity, specificity, and computational efficiency.

This study provides a systematic overview of a research project addressing fault detection challenges in WSNs. It begins with an introduction; reviews existing studies; proposes a novel framework integrating noise filtering, feature optimization, and a hierarchical DL model; compares the approach against existing methodologies; and concludes with key contributions and potential future directions.

2. Related Work

The literature survey section provides a comprehensive overview of existing approaches in fault detection for WSNs. It examines the evolution of techniques in noise filtering, feature selection, and DL, identifying the strengths and limitations of current methodologies.

Tan et al. (2019) introduced an intrusion detection approach that leverages a random forest classifier enhanced by the synthetic minority oversampling technique to address dataset imbalance, improving accuracy from 92.39% to 92.57%. In a similar vein, Rezvi et al. (2021) employed a data mining framework to discern various types of denial of service attacks by comparing several classifiers—such as K-nearest neighbors (KNN), naïve Bayes, logistic regression, support vector machine, and artificial neural network—with their findings indicating that artificial neural network and KNN yielded superior accuracies of 98.56% and 98.4%, respectively. Meng et al. (2022) proposed an intrusion detection method tailored for resource-constrained WSNs, integrating a light gradient boosting machine with recursive feature elimination, Shapley additive explanations analysis, and an iterative tree model, in combination with the synthetic minority oversampling technique-Tomek balancing technique, which resulted in detection rates exceeding 99% for all attack types and a substantial reduction (46%) in modeling time.

Singh et al. (2020) developed a fuzzy rule-based intrusion prevention system that classifies sensor nodes into risk categories based on metrics, like packet delivery ratio, energy consumption, and signal strength, achieving an accuracy of 98.29% and effectively neutralizing malicious nodes. Alruhaily et al. (2021) designed a multi-tier intrusion detection architecture

incorporating a real-time naïve Bayes classifier at the network edge and a cloud-based random forest classifier for comprehensive packet analysis, with their system delivering high detection accuracies across various attack categories. Complementing these efforts, Chandre et al. (2022) employed a convolutional neural network within a DL framework to detect and prevent intrusions by extracting robust feature representations from extensive labeled datasets, reaching an accuracy rate of 97%.

Further advancing the field, an optimized collaborative intrusion detection system was proposed by Elsaid and Albatati (2020) using an updated artificial Bee colony optimization (BCO) algorithm that enhanced resource efficiency and detection accuracy while integrating a weighted support vector machine to minimize false alarms through effective coordination among base stations, cluster heads, and sensor nodes. Addressing data imbalance in WSN cyberattacks, Putrada et al. (2022) demonstrated that extreme gradient boosting outperformed decision trees and naïve Bayes by achieving the highest AUC values across multiple attack classes. Ravindra et al. (2023) introduced an innovative anomaly detection technique that utilizes data compression and dynamic thresholding, powered by an enhanced extreme learning machine coupled with an enhanced transient search arithmetic optimization (ETSAO) algorithm, which successfully reduced computational overhead and achieved a 96.90% accuracy on the WSN-DS. Finally, Alruwaili et al. (2023) presented the red kite optimization algorithm (RKOA) with an average ensemble model for intrusion detection (AEID) methodology for IoT-based WSNs, which incorporates feature selection through RKOA, min-max normalization, and an average ensemble learning model with hyperparameter tuning using a Lévy-fight chaotic whale optimization technique, resulting in an improved accuracy of 98.94%.

While current methodologies effectively address individual aspects such as detection accuracy, computational efficiency, and class imbalance, they seldom integrate noise filtering, feature selection, and DL-based fault detection into a unified framework. Moreover, many approaches do not fully exploit hierarchical DL architectures capable of capturing both short-term and long-term temporal dependencies inherent in sensor data. This gap underscores the need for a comprehensive and scalable solution that simultaneously enhances network reliability, minimizes false alarms, and extends the operational lifespan of WSNs, thereby offering robust performance in dynamic and resource-constrained environments.

3. Proposed Methodology

The proposed methodology introduces an advanced framework (Fig. 1) for fault detection in WSNs, addressing the critical challenges of noise interference, suboptimal feature selection, and inaccurate fault classification in dynamic environments. By integrating a suite of cutting-edge techniques, the approach ensures enhanced fault detection accuracy and robust network performance while optimizing computational efficiency. The methodology begins with DNF using adaptive thresholding, a real-time noise mitigation strategy that dynamically adjusts thresholds based on statistical analysis of noise patterns in sensor data. This ensures the preservation of critical fault-indicative information while filtering out irrelevant fluctuations, even under varying environmental conditions. This adaptive mechanism significantly enhances the data quality fed into the fault detection pipeline. To extract the most relevant and non-redundant features, an RWOA was utilized. This novel metaheuristic optimization approach combines the global exploration capabilities of WOA with feature relevance ranking using mutual information. By balancing classification accuracy and feature dimensionality, the RWOA ensures the selection of an optimal, compact feature set, reducing computational overhead while maintaining precision.

For fault classification, the proposed framework leverages an HADL model. This multi-layered architecture incorporates temporal convolutional layers to capture short-term patterns and anomalies, followed by LSTM layers to model long-term dependencies in time-series data. The centerpiece of HADL is its dual-level hierarchical attention mechanism, which prioritizes critical features within each time step and across the sequence, maximizing interpretability and decision accuracy. The final classification layer delivers precise fault predictions, adapting to the complexities of real-world WSN scenarios. The proposed methodology establishes a high-performance fault detection framework by synergistically combining noise filtering, feature optimization, and a DL-based classification model. This approach not only enhances the quality of service but also ensures the scalability, reliability, and efficiency of WSNs in mission-critical applications. Integrating adaptive mechanisms and optimization-driven feature selection represents a significant advancement in fault detection technology, paving the way for more resilient and intelligent WSN deployments.

3.1. DNF Technique with Adaptive Thresholding

Initially, a DNF technique with adaptive thresholding effectively filters out noise while preserving critical data for fault detection in WSNs. First, the technique continuously monitors incoming

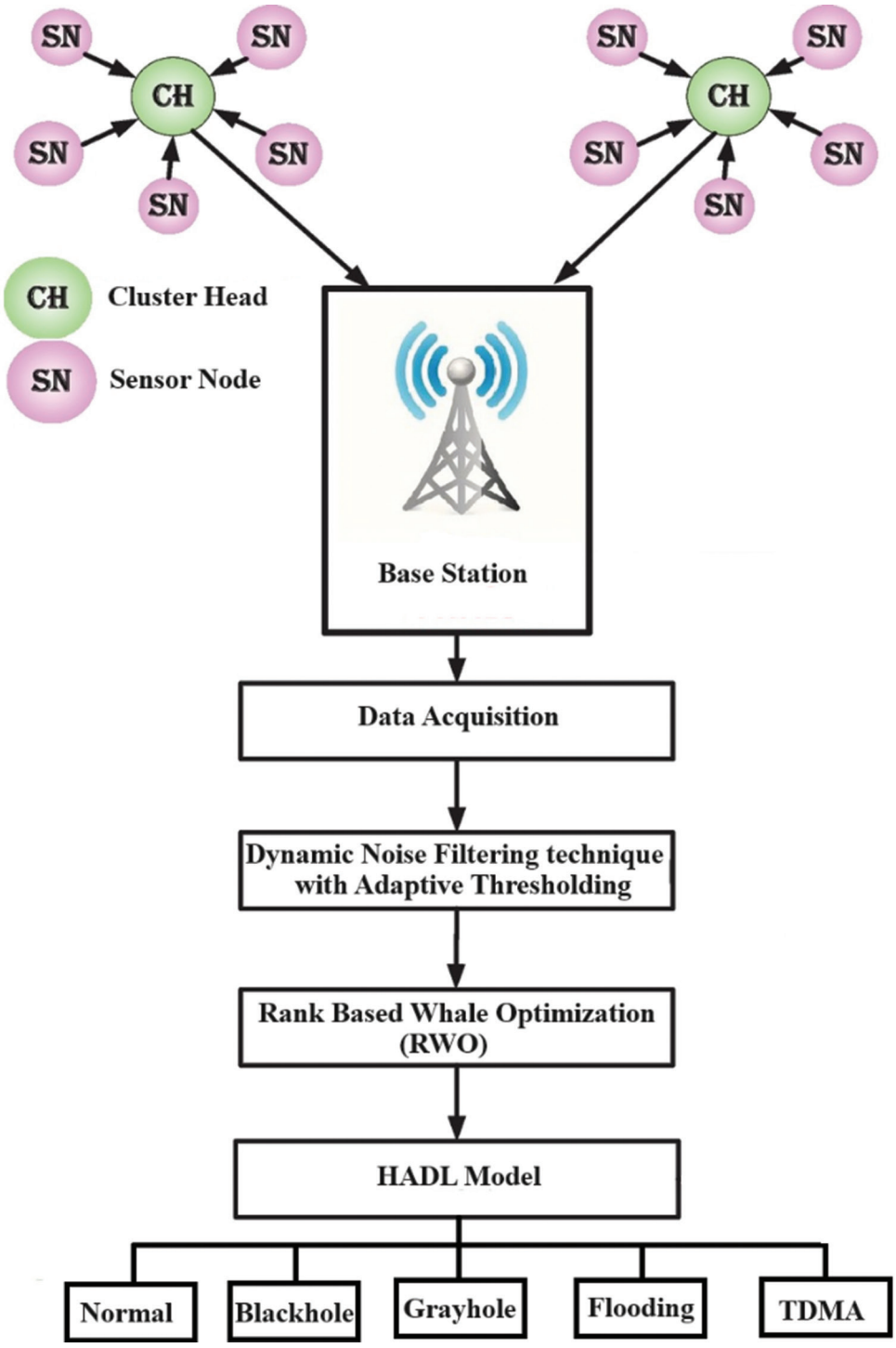

Fig. 1. Flow diagram of the proposed methodology
Abbreviations: HADL: Hierarchical attention-based deep learning; TDMA: Time-division multiple access

sensor data to assess the real-time noise levels and distribution patterns. It then calculates statistical properties, such as the mean and standard deviation of the noise, across a sliding window of recent data. Based on these calculations, the method dynamically adjusts the noise threshold, increasing it during high-noise periods to avoid false positives and lowering it when data quality improves to ensure that subtle faults are not missed. This adaptive threshold is then applied to filter out noise, allowing only data points that exceed an adjusted threshold to pass through for further processing. The process is repeated continuously, ensuring the filtering adapts to changing network conditions, resulting in a more accurate and reliable dataset for subsequent analysis.

DNF with adaptive thresholding is a technique

used to improve the data quality in systems like WSNs, where data can be corrupted by noise due to various factors like sensor malfunctions, environmental interference, or communication issues. This method enhances the signal-to-noise ratio by removing noise without losing important information. This component refers to identifying and reducing noise in the data in real-time or dynamically, based on varying conditions. The noise filtering adapts to the type of noise and the changing characteristics of the signal. In a WSN, sensor readings can be affected by several types of noise, such as random fluctuations or environmental disturbances. DNF identifies and selectively removes these anomalies in the data, ensuring that useful signals are preserved.

Thresholding involves setting a limit (threshold) above or below which the data is considered noise or valid. Adaptive thresholding adjusts this threshold based on the current state of the data. In a dynamic environment, where sensor data characteristics change over time, a static threshold might not work effectively. The adaptive threshold is recalculated periodically or based on specific criteria, such as the variance of the data, the signal strength, or statistical measures of the data distribution. For example, if the sensor data shows sudden spikes or sharp drops (indicative of noise), the threshold can be adjusted to treat these as noise and filter them out. Conversely, when data becomes more stable or predictable, the threshold can be widened to capture a broader range of valid information. The algorithm for DNF with adaptive thresholding is as follows:

(i) Step 1: The system continuously monitors incoming sensor data for unusual patterns, sudden spikes, or deviations from expected values, characteristic of noise
(ii) Step 2: The system uses adaptive techniques to determine a dynamic threshold that reflects the current data distribution, variability, or other environmental factors. The threshold changes are based on observed conditions, such as the variance of the signal or the presence of unusual outliers
(iii) Step 3: Data points outside the adaptive threshold are flagged as noise and discarded or replaced. The remaining data is preserved for further processing and analysis
(iv) Step 4: By dynamically adjusting the threshold, the method ensures that important or meaningful data is not discarded while filtering out noise. This allows for better quality input for downstream analysis, such as fault detection in WSNs.

Removing noise without discarding useful data improves the quality of sensor readings, leading to better analysis and decision-making. The adaptive threshold can adjust to different types of noise or changes in the network conditions, making it more robust in dynamic environments. In systems like fault detection, reducing noise ensures that only actual faults are detected, minimizing false alarms. In summary, DNF with adaptive thresholding ensures that the data used for analysis in WSNs or similar systems is of high quality, with noise effectively removed based on real-time conditions.

3.2. RWOA Technique

This research presents an RWOA to improve the efficacy of feature extraction. It uses the latest developments in metaheuristic optimization techniques to find the most pertinent features for defect identification. To start, the WOA searches the feature space to optimize a fitness function that strikes a compromise between feature set size and classification effectiveness. The algorithm effectively explores the search space, avoiding local optima and locating the optimal solution globally by imitating the bubble-net feeding method of humpback whales. Features with stronger correlations are given larger weights. In parallel, the dependence of each characteristic on the goal variable (i.e., defect or normal state) is evaluated using mutual information. The selected features from the WOA are then refined using the mutual information ranking, ensuring that only the most informative and non-redundant features are retained. This hybrid approach significantly improves the robustness and accuracy of the fault detection model by ensuring that the extracted features are both optimal in relevance and minimal in quantity, reducing the computational burden.

One of the popular population-based metaheuristic algorithms for solving global optimization problems in various fields is the WOA algorithm, developed by Mirjalili and Lewis (2016). The humpback whale's natural hunting behavior serves as the model for this program. At the water's surface, humpback whales hunt by focusing on schools of krill or tiny fish. To encircle and seize their prey, they form characteristic bubbles in a spiral pattern. The whales descend and swim to the water's surface, creating spiral bubbles around the prey. The WOA uses three tactics to mimic whale behavior: (i) spiral bubble-net attacking (exploitation phase), (ii) hunting for prey (exploration phase), and (iii) surrounding the target. $X_i^t = \left(x_{i,1}^t, x_{i,2}^t, \ldots, x_{i,D}^t\right)$ represents the location of the i^{th} whale at iteration t, where $i = 1,2,\ldots, N$ and N and D represent the whale population and the problem's dimensions, respectively.

3.3. Encircling Prey Strategy

Whales can track down and enclose their prey. The ideal choice for whales in WOA is the target prey

or a nearby location inside the search area. Eq. (1) is used by other whales to update their location as they try to approach the ideal agent during prey encirclement. This equation is constructed with t representing the current iteration, X_i^t representing the i^{th} whale's location for the current iteration, and X^{*t} representing the position vector of the best solution, thus far, which is updated in each iteration if a better solution is discovered.

$$X_i^{t+1} = X^{*t} - A \cdot D \tag{1}$$

$$D = \left| C \times X^{*t} - X_i^t \right| \tag{2}$$

where D stands for the distance between the whale and the prey X^{*t}, which is established by Eq. (2). I denote the absolute value, A and C represent coefficient vectors that are established using Eqs. (3) and (4).

$$A = 2 \times a \times r - a \tag{3}$$

$$C = 2 \times r \tag{4}$$

$$a = 2 - t \times \left(\frac{2}{MaxIter} \right) \tag{5}$$

According to Eq. (5), the parameter r in Eqs. (3) and (4) are random numbers in the interval, whereas Eq. (3) declines linearly from 2 to 0 repetitions. The values t and Max_{iter} are used in Eq. (5) to represent the current iteration and the total number of iterations. Through the use of the parameter a, the whales are gradually brought into the surrounding scope.

3.4. Spiral Bubble-net Attacking Strategy

Humpback whales use a bubble net to spiral toward their prey and corner them. Two strategies are used to mathematically represent this strategy: spiral updating position and shrinking encircling.

3.4.1. Shrinking encircling method

In Eq. (3), this tendency is reflected by reducing the value of the convergence variable a. Furthermore, the alternate range of A fluctuation is linearly lowered from 2 to 0, utilizing the parameter a through iterations. In other words, A is a random value belonging to the interval $[-a,a]$.

3.4.2. Spiral updating position method

First, this method calculates the distance between whales X_i^t using Eq. (6); X^{*t} is the best result thus far. Next, a spiral migration from its present location towards an ideal solution is described using Eq. (7). In these calculations, the logarithmic spiral shape is determined by a constant parameter, b, and a random variable, l, between [−1,1].

$$D' = \left| X^{*t} - X_i^t \right| \tag{6}$$

$$X_i^{t+1} = D' \times e^{bl} \times \cos(2\pi l) + X^{*t} \tag{7}$$

The logarithmic spiral form is determined by a fixed parameter, b and a random value, l, that falls between [−1,1]. The humpback whale swims in WOA, spiraling around its prey in a tight circle. The spiral model or the diminishing encircling method is the two options the whale chooses for changing its location during the optimization phase. The mathematical model is defined by Eq. (8), where p is a random number in [0, 1].

$$X_i^{t+1} = \begin{cases} X^{*t} - A \times D \ \ if \ \ p < 0.5 \\ D' \times e^{bl} \times \cos(2\pi l) + X^{*t} \ \ if \ \ p \geq 0.5 \end{cases} \tag{8}$$

3.5. Searching for Prey Strategy

Whales employ this strategy to increase population diversity and seek the problem space for uncharted territory. A randomly selected search agent updates the position of each whale. To avoid being caught in a local minimum, the search agent is pushed away from a randomly chosen humpback whale using the parameter A. Eq. (9) is employed for exploration [31].

$$\begin{aligned} X_i^{t+1} &= X_{rand} - A \times D \\ D &= \left| C \times X_{rand} - X_i^t \right| \end{aligned} \tag{9}$$

here A and C are calculated using Eqs. (3) and (4), and X_{rand} is a random position vector in the search space chosen from the available whales in the population.

After N, when whales are randomly distributed over the search space, the association objective function value is determined, as seen in the WOA flowchart in Fig. 2. When the initial values of the control parameters

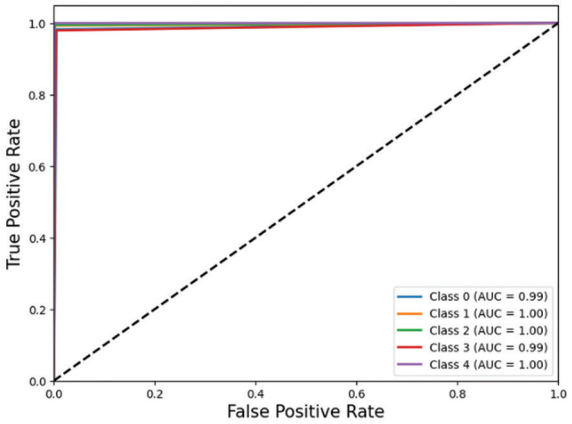

Fig. 2. Area under the curve (AUC) of the classes

are altered, the optimization begins with the present iterations. At each iteration, the value of the parameter p is then evaluated. Eq. (7) specifies the spiral updating position method used by whales when $p \geq 0.5$. Whales update their location when $p < 0.5$ using the encircling prey strategy (Eq. [1]) if $|A|<1$ and the hunting for prey strategy (Eq. [9] if $|A|\geq 1$). Subsequently, the fitness and viability values of the newly gained positions are computed. Next, the ideal solution is updated, and WOA is ultimately ended.

The WOA is based on humpback whale hunting behavior, mimicking the bubble-net strategy. It involves exploration and exploitation, with whales randomly moving in search of space and using shrinking encircling mechanisms and spiral movements. However, WOA can face challenges like premature or slow convergence due to poor exploration and exploitation balance. The rank-based method is introduced in RWOA to enhance the population selection mechanism during the optimization process. This method changes how whales are selected for exploration and exploitation by considering their rank in the population rather than selecting them randomly or with equal probability.

3.5.1. Rank assignment

After evaluating the fitness of all candidates (whales), they are ranked based on their fitness values (i.e., solutions with lower objective function values are ranked higher if the goal is minimization). Each individual in the population is assigned a rank based on their fitness, with the best solution (with the lowest fitness) getting rank 1, the second-best getting rank 2, and so on.

3.5.2. Probability-based selection

Instead of choosing individuals to update their position randomly or based on fixed probabilities, rank-based selection assigns higher probabilities to individuals with better (lower) ranks. The better individuals (those with better fitness) are more likely to be selected for the exploitation phase, while the worse individuals are more likely to be selected for the exploration phase. A non-linear probability distribution is often used, so the probability of selecting a whale is inversely proportional to its rank. This ensures that the algorithm focuses more on promising solutions while maintaining some diversity by allowing worse solutions to participate in the search process.

3.5.3. Exploration and exploitation

During the exploration phase, the worst-ranked individuals (those with higher ranks) can move freely, encouraging the algorithm to explore a wide area of the search space. During the exploitation phase, the best-ranked individuals (those with lower ranks) are more likely to contribute to the search, refining the solution by focusing on regions with promising results.

3.5.4. Fitness-based movement

The movement of each whale is influenced by its rank. For example, for better individuals (lower ranks), they will likely refine their position by getting closer to the best solution. For worse individuals (higher ranks), they are more likely to perform a broader search to avoid premature convergence and encourage diversity.

3.5.5. Rank-based update of positions

The whale's position update rule, which typically involves a spiral or encircling mechanism, can also be influenced by the whale's rank. For example, whales with better ranks (i.e., better solutions) may use the shrinking encircling method with higher probabilities to exploit reasonable solutions, while whales with worse ranks can have a higher probability of using random search to explore new regions of the search space.

The RWOA improves convergence by ensuring better solutions drive the search process, leading to faster and more accurate results. It enhances diversity by allowing worse solutions to explore the search space, avoiding premature convergence, and maintaining population diversity. RWOA also balances exploration and exploitation, allowing for a broader search space and reducing the risk of stagnation by encouraging weaker solutions to explore new areas.

3.6. Hierarchical Attention-based DL Model

Finally, a HADL model is employed for fault detection in WSNs, which starts with an embedding layer that converts the raw input features from the WSN-DS into dense vectors, capturing the underlying patterns in a compressed form. Following this, temporal convolutional layers detect patterns and anomalies over short data sequences, focusing on how features change over time. These layers help identify sudden shifts or unusual trends that might indicate faults. Next, recurrent layers, such as LSTM units, capture long-term dependencies in the time-series data, effectively modeling how earlier data points influence future observations. The central innovation of HADL is its hierarchical attention mechanism, which is applied at multiple stages: first, to highlight the most relevant features within each time step, and then to focus on the most important time steps across the sequence. This dual-level attention ensures the model

prioritizes the most critical information for accurate fault detection. Finally, the processed data is passed through a fully connected layer, which integrates the information from all previous layers and leads to a softmax classification layer. This final layer provides a probability distribution over the fault classes, allowing the model to make precise and confident predictions about the network's state. This layered structure of HADL ensures that the model effectively captures both immediate and long-term patterns in the data, leading to enhanced accuracy in detecting faults in WSNs.

The rank-based selection in RWOA improves convergence by ensuring better solutions drive the search process, leading to faster and more accurate results. It enhances diversity by allowing worse solutions to explore the search space, avoiding premature convergence, and maintaining population diversity. RWOA also balances exploration and exploitation, allowing for a broader search space and reducing the risk of stagnation by encouraging weaker solutions to explore new areas.

Recurrent neural networks are widely known for their ability to capture the dynamics of sequential data when working with time-sequence data supplied by monitoring systems. In contrast to a traditional neural network, HADL neurons are reinforced by including edges that span neighboring time steps. These links, which are referred to as recurrent edges, create cycles that are self-connected of a neuron to itself over time, adding a temporal component to the model data space. The behavior of a neuron with recurrent edges in a basic recurrent network may be explained as follows in Eq. (10):

$$h^{(t)} = F(Wh^{(t-1)} + Ux^{(t)} + b_h)$$
$$y^{(t)} = G(Vh^{(t)} + b_y) \quad (10)$$

where $h^{(t)}$ represents the hidden layer activation at time t, $h^{(t-1)}$ represents the previously hidden representation, and x (t) represents the input layer's current input. The input-to-hidden, hidden-to-hidden, and hidden-to-output connections are parametrized by the weight matrices W, U, and V, respectively, within the HADL. The output layer and hidden layer bias parametersb_y and b_h allow offset learning. The two layers' activation functions are F and G, respectively. The recurrent neural network's output is $y^{(t)}$. In contrast to the propagation between layers, which is cyclic, the data propagation is one-way in the time direction when the network is unfurled from left to right. The distinction lies in the weights (W) being shared between time steps. The network may therefore be trained across several time steps using a backpropagation approach. As t2–t1 grows in size, the input's contribution to time step t2 will either move to infinity or decay to zero since the weights are the same for all time steps. The loss gradient will also either burst or decay to the input, depending on the activation function f and whether $|W| > 0$ *or* $|W|<0$.

In the HADL method, every neuron in the hidden layer is substituted by a memory cell architecture with a core node known as the state unit $\boldsymbol{s}(^t)$. This model's architecture is similar to that of a typical recurrent neural network with a hidden layer. Like a typical neuron in a hidden layer, the cell has external outputs to the next time step and the layer below, as well as external inputs from the previous layer and the prior state. In addition, it features an internal set of gating units that use multiplication to regulate the information flow. Updates are made to the forgetting gate unit $f_i^{(t)}$, state unit $s^{(t)}$, input gate unit $g_i^{(t)}$, output gate unit $q_i^{(t)}$, and output $h_i^{(t)}$ for each time step t based on the current input $x_j^{(t)}$ and the prior output $h_j^{(t-1)}$. Below is the computing process for an LSTM model at each stage (Eq. [11]):

$$f_i^{(t)} = \sigma\left(b_i^f + \sum_j U_{i,j}^f x_j^{(t)} + \sum_j W_{i,j}^f h_j^{(t-1)}\right)$$

$$s_i^{(t)} = f_i^{(t)} s_i^{(t-1)} + g_i^{(t)} \sigma\left(b_i + \sum_j U_{i,j} x_j^{(t)} + \sum_j W_{i,j} h_j^{(t-1)}\right)$$

$$g_i^{(t)} = \sigma\left(b_i^g + \sum_j U_{i,j}^g x_j^{(t)} + \sum_j W_{i,j}^g h_j^{(t-1)}\right)$$

$$q_i^{(t)} = \sigma\left(b_i^o + \sum_j U_{i,j}^o x_j^{(t)} + \sum_j W_{i,j}^o h_j^{(t-1)}\right)$$

$$h_i^{(t)} = \tanh(s_i^{(t)}) q_i^{(t)} \quad (11)$$

The state unit and the three gate units are all triggered by the sigmoid function σ (·) and have their own bias b_i, input weights $U_{i,j}$, and recurrent weights $W_{i,j}$. Ultimately, the HADL cell's output is modified to reflect the hidden layer vector $h_i^{(t)}$. When an input/output gate is activated in the forward direction, the HADL may learn when and to what degree to let values in/out. The value of the hidden layer will neither increase nor decrease if both gates are closed, meaning that neither outputs nor intermediate time steps will be impacted. The gradients can also propagate backward throughout many time steps without disappearing or bursting. That is, gates may learn when to allow error to enter and when to limit it. The ability of HADL to learn long-term dependencies more effectively than standard recurrent designs has made it popular for a wide range of real-world applications.

The degree to which each input contributes to a target class of interest c, or the relevance score of each input concerning c, are among the things we are interested in understanding, given a trained neural network classifier. The fundamental principle behind HADL is to assign a relevance score to each input by

tracking each one's layer-by-layer contribution to the final prediction, $f(x)$. According to the conservation principle, the overall relevance allocated to one layer should match the total relevance allocated to the layer before. This is what the HADL method does. Given two successive layers of a neural network, let's say m and n, the relevance scores meet the following criteria (Eq. [12]):

$$\sum_i R_i^{(m)} = \sum_i R_i^{(n)} = f\left(x\right) \tag{12}$$

In layers m and n, respectively, the relevance scores of individual neurons are denoted by $R_i^{(m)}$ and $R_i^{(n)}$. The rules governing the propagation of relevance scores between two layers by Eq. (9) are varied to accommodate the features of various neural network structures. Eq. (13) illustrates a basic rule:

$$R_i^{(m)} = \sum_j \frac{z_{i,j}}{\sum_k z_{k,j}} R_j^{(n)} \tag{13}$$

where $\sum_k z_{k,j}$ is the total contribution/relevance delivered to neuron j from all linked neurons in layer m before the application of a nonlinear activation function; and $z_{i,j}$ is the contribution/relevance received by neuron j in layer n from an activated neuron i in layer m. This equation demonstrates the conservation principle, which also holds for deactivation, unconnected neurons, and zero weight (Eq. [14]).

$$R_i^{(m)} = \sum_j \frac{z_{i,j}}{\varepsilon + \sum_k z_{k,j}} R_j^{(n)} \tag{14}$$

Despite the HADL rule's many desirable qualities, robustness, and other improvements must be taken into account when applying it to real-world situations (Eq. [15]).

$$R_i^{(m)} = \sum_j \left(\alpha \frac{z_{i,j}^{+}}{\sum_k z_{k,j}^{+}} - \beta \frac{z_{i,j}^{-}}{\sum_k z_{k,j}^{-}} \right) R_j^{(n)} \tag{15}$$

To maintain numerical stability, the denominator has a modest positive term ε in comparison to the fundamental rule, where both the beneficial and detrimental effects from the upper layer n are denoted by $z_{i,j}^{+}$ and $z_{i,j}^{-}$, respectively, and the weights of the positive and negative contributions are controlled by α and β. $\alpha+\beta$ should be in line with the conservation principle. To provide the outcomes with stability and interpretability, the rule prefers the effects of positive contributions over negative ones. One can manually regulate the significance of positive and negative contributions by carefully selecting the values of coefficients α and β.

Temporal convolutional networks with LSTM and other gated neural networks feature a unique calculation called multiplicative interaction in addition to linear mapping computation in multi-layer perceptron architectures. Two neurons are multiplied by one another in this calculation, with one acting as a signal and the other as a gate that regulates the degree to which the signal affects the output (Eq. [16]):

$$a_p = \mathrm{f}(z_g) \cdot g(z_s) \tag{16}$$

where z_g and z_s are two neuron values supplied to the gate and signal unit from earlier layers, respectively, $f(\cdot)$ is the gate unit's activation function, and $g(\cdot)$ is the signal unit's activation function.

In contrast to linear mapping, multiplicative interaction's nonlinearity presents unique challenges related to reassigning importance to the preceding layer. An established redistribution hierarchical method known as "signal-take-all" is used when activation is obtained by multiplying the value of a gate neuron by the value of a signal neuron. This strategy includes (Eq. [17]):

$$(R_g, R_S) = (0, R_{\mathrm{p}}) \tag{17}$$

where the relevance scores for the gate and signal neurons are denoted by R_g and R_s, respectively. To comply with the conservation principle, the gate neuron takes zero, while the signal neuron takes all of the relevant R_p from the top layer.

The HADL is a versatile ML approach that excels in modeling complex data with multiple hierarchical relationships. Its attention mechanisms enhance model interpretability, allowing for a better understanding of the prioritization of features. HADL captures short-term and long-term dependencies, making it ideal for tasks like time-series analysis in WSNs. It also enhances feature learning with its hierarchical structure, allowing for better generalization and robustness in anomaly or fault detection tasks. HADL is adaptable to complex and noisy data, reducing overfitting and improving performance on time-series and sequential data. Its hierarchical nature allows it to scale efficiently to large datasets, making it suitable for real-world applications.

4. Results and Discussion

This section thoroughly analyzes the experimental results to assess the efficacy and efficiency of the suggested approach. The outcomes are compared to several cutting-edge methods using various criteria, including sensitivity, specificity, accuracy, and F1-score. The suggested approach outperformed the other methods by utilizing DL models and sophisticated optimization techniques, attaining near-perfect or perfect values in important measures.

4.1. Experimental Setup

The discussion focuses on interpreting these results, highlighting the impact of the proposed approach on fault detection in WSN, and addressing how the method fills existing research gaps in reliability and precision. The experiments used Python 3.7 as the implementation platform, leveraging libraries such as NumPy, Pandas, TensorFlow/PyTorch, Scikit-learn, and Matplotlib for model development, optimization, and visualization. The system's performance was evaluated under controlled conditions to ensure the robustness and generalizability of the results. The implementation and experimentation were performed on the following system configuration:

(i) Processor: Intel Core i7-12700H (12th Gen) with 14 cores (6 performance + 8 efficiency cores) and a clock speed up to 4.7 GHz
(ii) Random access memory: 16 GB DDR4 3200 MHz, enabling efficient data handling and processing of large datasets
(iii) Storage: 1 TB NVMe SSD, ensuring fast data read/write operations and loading of large models
(iv) Operating system: Windows 11 64-bit, with Python 3.7 as the programming environment
(v) Software frameworks: TensorFlow 2.9, PyTorch 1.12, Scikit-learn 1.1, and Matplotlib 3.5.

This high-performance configuration ensured the smooth execution of computationally intensive tasks, such as hyperparameter tuning, training DL models, and performing iterative optimization. The experiments were iteratively refined to achieve optimal results, balancing computational efficiency and prediction accuracy. The setup included advanced optimization algorithms, fault detection models, and dynamic filtering techniques, tested under controlled conditions to ensure reliable and reproducible results. This environment facilitated seamless experimentation, from pre-processing the WSN-DS to training and evaluating the proposed HADL.

4.2. Dataset Description

The WSN-DS used in this study is a comprehensive and widely used benchmark for fault detection in WSNs. It contains various simulated data representing five distinct classes: normal, grayhole, blackhole, time-division multiple access (TDMA), and flooding. The dataset includes a total of 60,000 instances, with each instance comprising detailed features that capture the behavior and state of network nodes under different conditions. The normal class represents typical, fault-free network operations, while the remaining classes correspond to various network faults and malicious attacks, such as packet-dropping and routing disruptions. Each class is well-balanced, ensuring robust performance evaluation across all fault categories. The dataset provides feature-rich instances, including metrics like node energy levels, packet counts, delays, and routing information, offering a realistic simulation of network scenarios. These features were carefully pre-processed, normalized, and split into training and testing sets to facilitate effective model training and validation. This dataset serves as an ideal foundation for evaluating the performance of fault detection methods in complex WSN environments.

4.3. Performance Metrics

Fig. 3 illustrates the convergence behavior and effectiveness of the RWOA. The initial phase, from the first to the second iterations, shows a significant drop in fitness value, indicating the algorithm's exploratory phase. From the second to the seventh iterations, the plateau phase is marked by a plateau, where the algorithm focuses on refining solutions within a promising region. The further refinement phase decreases slightly to 0.0267, indicating a near-optimal solution and fine-tuning results. The graph demonstrates the algorithm's efficiency in narrowing down the search space, the plateau phase, where the algorithm focuses on exploitation, and the final convergence, where the algorithm has converged to a solution near the global optimum. This graph demonstrates the algorithm's ability to efficiently find an optimal solution while avoiding unnecessary computations beyond the point of diminishing returns.

The confusion matrix represents the performance of a classification model across five classes: Normal, Grayhole, Blackhole, TDMA, and Flooding (Fig. 4). The diagonal elements indicate the correctly classified instances, while off-diagonal elements represent misclassifications. The model performs well overall, with high accuracy for each class, as evidenced by the large diagonal values. For example, normal has 11,857 true positives, with minimal misclassifications.

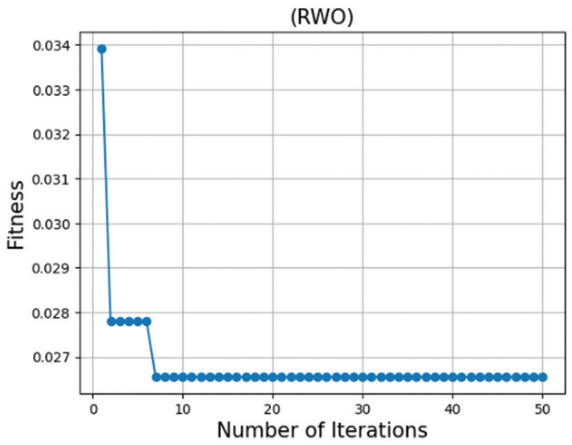

Fig. 3. Convergence behavior and effectiveness of the rank-based whale optimization algorithm

Similarly, grayhole achieves 11,885 correct classifications, though 86 instances were misclassified as blackhole and three as normal. The blackhole class also performed well, with 11,950 true positives and minor misclassifications. For TDMA, the model correctly identified 11,695 instances, though there is some confusion with normal, grayhole, and blackhole. Finally, the Flooding class exhibited near-perfect classification, with 11,991 correct predictions and no significant misclassifications. The confusion matrix highlights the model's robustness but also reveals areas for improvement, such as reducing misclassifications between normal and TDMA and minimizing confusion between grayhole and blackhole.

The AUC values for the five classes (0–4) indicate the model's excellent discriminatory ability across all categories (Fig. 2). AUC values ranged from 0 to 1, with values closer to 1 representing superior performance. Here, the AUC for class 0 (Normal) and class 3 (TDMA) is 0.99, indicating that the model can distinguish these classes from the others with near-perfect accuracy. For class 1 (Grayhole), class 2 (Blackhole), and class 4 (Flooding), the AUC is a perfect 1, demonstrating flawless classification for these classes. This suggests that the model had no false positives or negatives for classes 1, 2, and 4, showing exceptional precision and recall. Overall, the AUC values underscore the model's high reliability and effectiveness in differentiating between all classes, with minimal room for improvement.

The accuracy and loss curves in Fig. 5A and B depict the model's performance during training and testing. In plot 5A, the accuracy curve steadily increases during training, indicating that the model is learning effectively from the data. The testing accuracy also improves and stabilizes, closely aligning with the training accuracy, suggesting good generalization and minimal overfitting. In plot 5B, the loss curve decreases over epochs for both training and testing, reflecting a reduction in prediction errors as the model optimizes its parameters. The convergence of training and testing loss at low values confirms the model's robust learning process. A smooth and stable trajectory for both accuracy and loss curves indicates that the model training is well-tuned, with no signs of underfitting or overfitting, and performs consistently on unseen test data.

Fig. 6 shows the performance metrics for training and testing. With an overall accuracy of 99%, the model performed exceptionally well in the testing phase across all five classes of WSNs. With values near 0.99 or 1.00, the model's accuracy, recall, and F1-scores were continuously high, demonstrating its capacity to accurately detect occurrences of each class while reducing false positives and false negatives. With a perfect score, the flooding class exhibited faultless detection. The performance of other classes, such as blackhole and grayhole, was also strong. Weighted and macro average measures further support the model's balanced performance across classes. The model's outstanding performance during training and testing, together with its ability to balance accuracy, recall, and F1-score, shows its usefulness in real-world situations where reliable and precise fault classification is required.

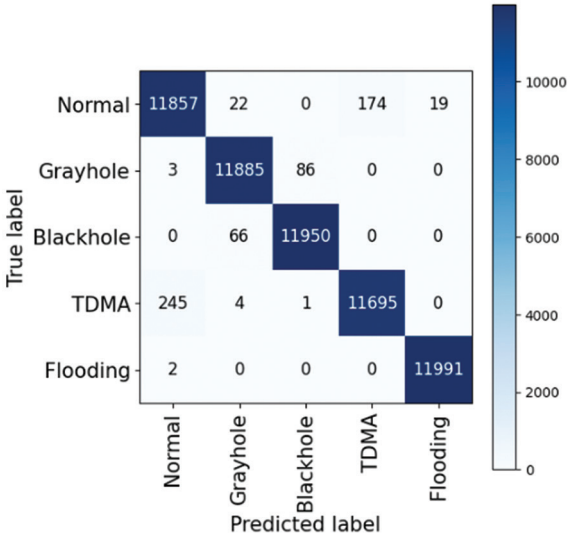

Fig. 4. Confusion matrix
Abbreviation: TDMA: Time-division multiple access

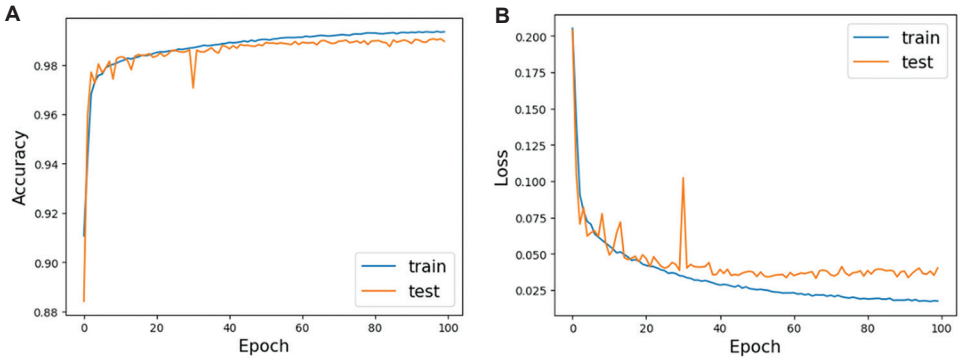

Fig. 5. (A and B) Accuracy and loss curves

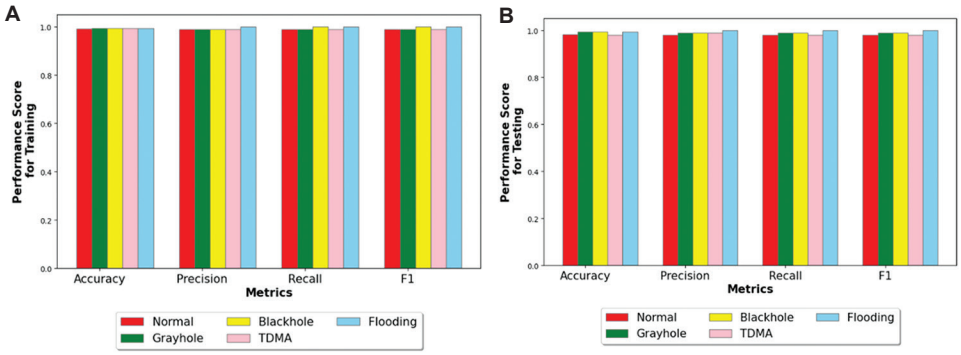

Fig. 6. (A and B) Performance metrics for training and testing
Abbreviation: TDMA: Time-division multiple access

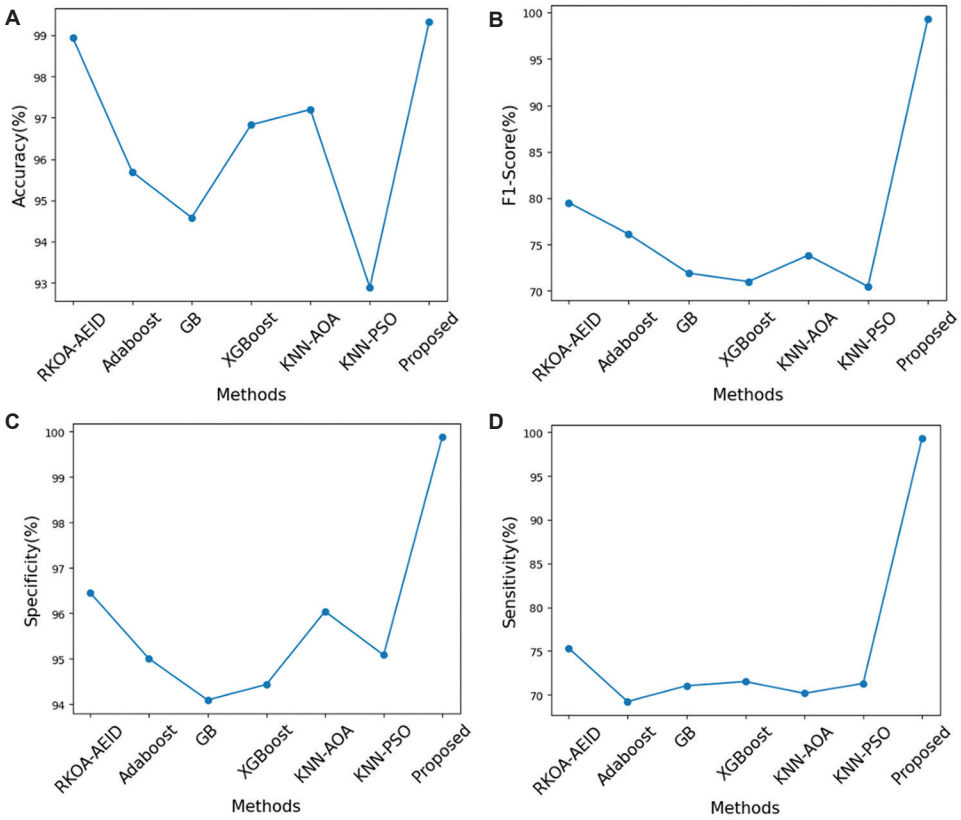

Fig. 7. (A-D) Comparative analysis for performance metrics
Abbreviations: AOA: Angle or arrival; GB: Gradient boosting; KNN: K-nearest neighbor; RKOA-AEID: Red kite optimization algorithm-average ensemble model for intrusion detection; PSO: Particle swarm optimization; XG Boost: Extreme gradient boosting

4.4. Comparison Metrics

Fig. 7 shows the performance of various methods, including RKOA-AEID (Alruwaili et al., 2023), Adaboost (Aljebreen et al., 2023), gradient boosting (Aljebreen et al., 2023), extreme gradient boosting (Alqahtani et al., 2019), KNN-angle of arrival (Liu et al., 2022), KNN-particle swarm optimization (Liu et al., 2022), and the proposed method, across four evaluation metrics: accuracy, F1-score, specificity, and sensitivity (Table 1). The proposed method achieved the highest accuracy (99%), demonstrating superior reliability in fault detection. RKOA-AEID performed well (98%), while Adaboost and KNN-particle swarm optimization performed moderately (94%). Gradient boosting, extreme gradient boosting, and KNN-angle of arrival exhibited intermediate results (97%). The proposed method excelled with a perfect F1-score (100%), indicating an exceptional balance between precision and recall. Adaboost and gradient boosting

Table 1. Comparative chart of the proposed model with conventional methods

Methods	Accuracy	Sensitivity	Specificity	F-score
Red kite optimization algorithm-average ensemble model for intrusion detection (Alruwaili et al., 2023)	98.94	75.33	96.45	79.52
AdaBoost (Aljebreen et al., 2023)	95.69	69.22	95.00	76.13
Gradient booting (Aljebreen et al., 2023)	94.58	71.03	94.09	71.92
Extreme gradient boosting (Alqahtani et al., 2019)	96.83	71.51	94.43	71.01
K-nearest neighbor-angle of arrival (Liu et al., 2022)	97.20	70.16	96.04	73.85
K-nearest neighbor-particle swarm optimization (Liu et al., 2022)	92.89	71.30	95.08	70.48
Proposed	99.25	98.74	99.32	98.39

lagged (75%), reflecting weaker handling of false positives or false negatives. Extreme gradient boosting and KNN-angle of arrival performed moderately (97%). The proposed method consistently outperforms all other techniques, achieving perfect F1, specificity, and sensitivity scores and near-perfect accuracy.

5. Conclusion

This research presents a unified framework for fault detection in WSNs that effectively combines advanced noise filtering, optimized feature selection, and a sophisticated DL architecture. The proposed approach leverages a DNF technique with adaptive thresholding to cleanse the data while preserving its critical aspects, employs the RWOA to select the most relevant features, and utilizes an HADL model to capture both short-term and long-term dependencies in sensor data. Experimental evaluations of the WSN-DS confirm the framework's exceptional performance, achieving an accuracy of 99.25%, sensitivity of 98.74%, specificity of 99.32%, and an F-score of 98.39%. These results highlight the framework's capacity to reliably detect faults and reduce false alarms, ultimately enhancing network reliability and extending the operational lifespan of WSNs. The integration of these advanced methodologies not only addresses existing challenges in fault detection but also establishes a robust foundation for future enhancements, including real-time deployment and the incorporation of multi-modal data.

Funding

None.

Conflict of Interest

The authors declare that they have no conflict of interest.

Availability of Data

Not applicable.

References

Alghamdi, R., & Bellaiche, M. (2023). A cascaded federated deep learning based framework for detecting wormhole attacks in IoT networks. *Computers and Security Journal*, 125, 103014. https://doi.org/10.1016/j.cose.2022.103014

Aljebreen, M., Alohali, M.A., Saeed, M.K., Mohsen, H., Al Duhayyim, M., Abdelmageed, A.A., et al. (2023). Binary chimp optimization algorithm with ML based intrusion detection for secure IoT-assisted wireless sensor networks. *Sensors*, 23(8), 4073. https://doi.org/10.3390/s23084073

Almomani, O. (2021). A hybrid model using bio-inspired metaheuristic algorithms for network intrusion detection system. *Computers, Materials and Continua*, 68, 409–429. https://doi.org/10.32604/cmc.2021.016113

Alqahtani, M., Gumaei, A., Mathkour, H., & Maher Ben Ismail, M. (2019). A genetic-based extreme gradient boosting model for detecting intrusions in wireless sensor networks. *Sensors (Basel)*, 19(20), 4383. https://doi.org/10.3390/s19204383

Alruhaily, N.M., & Ibrahim, D.M. (2021), A multi-layer machine learningbased intrusion detection system for wireless sensor networks. *The International Journal of Advanced Science and Computer Applications*, 12(4), 281–288. https://doi.org/10.14569/ijacsa.2021.0120437

Alruwaili, F.F., Asiri, M.M., Alrayes, F.S., Aljameel, S.S., Salama, A.S., & Hilal, A.M. (2023). Red kite optimization algorithm with average ensemble model for intrusion detection for secure IoT. *IEEE Access*, 11, 131749–131758. https://doi.org/10.1109/access.2023.3335124

Chandre, P., Mahalle, P., & Shinde, G. (2022). Intrusion prevention system using convolutional

neural network for wireless sensor network. *International Journal of Artificial Intelligence*, 2252(8938), 8938. https://doi.org/10.11591/ijai.v11.i2.pp504-515

Chataut, R., Phoummalayvane, A., & Akl, R. (2023). Unleashing the power of IoT: A comprehensive review of IoT applications and future prospects in healthcare agriculture, smart homes, smart cities, and industry 4.0. *Sensors (Basel)*, 23(16), 7194. https://doi.org/10.3390/s23167194

Elsaid, S.A., & Albatati, N.S. (2020). An optimized collaborative intrusion detection system for wireless sensor networks. *Soft Computing*, 24(16), 12553–12567. https://doi.org/10.1007/s00500-020-04695-0

Gebremariam, G.G., Panda, J., & Indu, S. (2023). Design of advanced intrusion detection systems based on hybrid machine learning techniques in hierarchically wireless sensor networks. *Connectection Science*, 35(1), 2246703. https://doi.org/10.1080/09540091.2023.2246703

Ghazal, T. (2022). Data fusion-based machine learning architecture for intrusion detection. *Computers, Materials and Continua*, 70(2), 3399–3413. https://doi.org/10.32604/cmc.2022.020173

Heidari, A., & Jabraeil Jamali, M.A. (2022). Internet of things intrusion detection systems: A comprehensive review and future directions. *Cluster Computing*, 26, 1–28. https://doi.org/10.1007/s10586-022-03776-z

Liu, G., Zhao, H., Fan, F., Liu, G., Xu, Q., & Nazir, S. (2022). An enhanced intrusion detection model based on improved kNN in WSNs. *Sensors (Basel)*, 22(4), 1407. https://doi.org/10.3390/s22041407

Meng, D., Dai, H., Sun, Q., Xu, Y., & Shi, T. (2022). Novel wireless sensor network intrusion detection method based on lightGBM model. *IJAM - IAENG International Journal of Applied Mathematics*, 52(4), 23.

Mirjalili, S., & Lewis, A. (2016). The whale optimization algorithm. *Advances in Engineering Software*, 95, 51–67. https://doi.org/10.1016/j.advengsoft.2016.01.008

Nimbalkar, A..D., Azmat, A., & Patil, Y. (2023). Security issues in wireless sensor networks. *I-Manager's Journal on Wireless Communication Networks*, 11(2), 32. https://doi.org/10.26634/jwcn.11.2.19780

Pandey, J.K., Kumar, S., Lamin, M., Gupta, S., Dubey, R.K., & Sammy, F. (2022). A metaheuristic autoencoder deep learning model for intrusion detector system. *Mathematical Problems in Engineerin*g, 2022, 3859155. https://doi.org/10.1155/2022/3859155

Putrada, A.G., Alamsyah, N., Pane, S.F., & Fauzan, M.N. (2022). Xgboost for Ids on WSN Cyber Attacks with Imbalanced Data. In: *2022 International Symposium on Electronics and Smart Devices (ISESD)*, p1–7.

Qaiwmchi, N.A.H., Amintoosi, H., & Mohajerzadeh, A. (2020). Intrusion detection system based on gradient-corrected online sequential extreme learning machine. *IEEE Access*, 9, 4983–4999. https://doi.org/10.1109/ACCESS.2020.3047933

Ravindra, C., Kounte, M.R., Lakshmaiah, G.S., & Prasad, V.N. (2023). Etelmad: Anomaly detection using enhanced transient extreme machine learning system in wireless sensor networks. *Wireless Personal Communications*, 130(1), 21–41. https://doi.org/10.1007/s11277-023-10271-0

Rezvi, M.A., Moontaha, S., Trisha, K.A., Cynthia, S.T., & Ripon, S. (2021). Data mining approach to analyzing intrusion detection of the wireless sensor network. *Indonesian Journal of Electrical Engineering and Computer Science*, 21(1), 516–523. https://doi.org/10.11591/ijeecs.v21.i1.pp516-523

Sezgin, A., & Boyacı, A. (2023). Aid4i: An intrusion detection framework for industrial internet of things using automated machine learning. *Computers, Materials and Continua*, 76(2), 40287.

Sharmin, S., Ahmedy, I., & Md Noor, R. (2023). An energy-efficient data aggregation clustering algorithm for wireless sensor networks using hybrid PSO. *Energies*, 16(5), 2487. https://doi.org/10.3390/en16052487

Singh, N., Virmani, D., & Gao, X.Z. (2020). A fuzzy logic-based method to avert intrusions in wireless sensor networks using WSN-DS dataset. *International Journal of Computer Applications*, 19(3), 2050018. https://doi.org/10.1142/S1469026820500182

Talukder, M.A., Islam, M.M., Uddin, M.A., Akhter, A., Hasan, K.F., & Moni, M.A. (2022). Machine learning-based lung and colon cancer detection using deep feature extraction and ensemble learning, *Expert Systems with Applications*, 205, 117695. https://doi.org/10.1016/j.eswa.2022.117695

Talukder, M.A., Hasan, K.F., Islam, M.M., Uddin, M.A., Akhter, A., Yousuf, M.A., et al. (2023). A dependable hybrid machine learning model for network intrusion detection. *Journal of Information Security and Applications*, 72, 103405. https://doi.org/10.48550/arXiv.2212.04546

Talukder, M.A., Islam, M.M., Uddin, M.A., Hasan, K.F., Sharmin, S., Alyami, S.A., et al. (2024). Machine

learning-based network intrusion detection for big and imbalanced data using oversampling, stacking feature embedding and feature extraction. *Journal of Big Data*, 11, 33.

Tan, X., Su, S., Huang, Z., Guo, X., Zuo, Z., Sun, X., et al. (2019). Wireless sensor networks intrusion detection based on smote and the random forest algorithm. *Sensors (Basel)*, 19(1), 203. https://doi.org/10.3390/s19010203

Yakubu, M.M., & Maiwada, U.D. (2023). Resource limitations for wireless sensor networks to establish a comprehensive security system in the 5g network. *Umyu Scientifica*, 2(2), 44–52. https://doi.org/10.1007/s10207-024-00833-z

AUTHOR BIOGRAPHIES

R. Gayathri is a student in the Department of Computer Science and Engineering at Siddaganga Institute of Technology, affiliated with Visvesvaraya Technological University, Karnataka, India. Her areas of interest include machine learning, artificial intelligence, and data analytics.

K. N. Shreenath is an Associate Professor in the Department of Computer Science and Engineering at Siddaganga Institute of Technology, affiliated with Visvesvaraya Technological University, Karnataka, India. He has several years of teaching and research experience. His research interests include computer networks, data mining, artificial intelligence, and software engineering. He has guided numerous student projects and published papers in national and international journals and conferences.

INSTRUCTIONS TO AUTHORS

Submission of papers

The International Journal of Systematic Innovation is a refereed journal publishing original papers four times a year in all areas of SI. Papers for publication should be submitted online to the IJoSI website (http://www.ijosi.org) In order to preserve the anonymity of authorship, authors shall prepare two files (in MS Word format or PDF) for each submission. The first file is the electronic copy of the paper without author's (authors') name(s) and affiliation(s). The second file contains the author's (authors') name(s), affiliation(s), and email address(es) on a single page. Since the Journal is blind refereed, authors should not include any reference to themselves, their affiliations or their sponsorships in the body of the paper or on Figs and computer outputs. Credits and acknowledgement can be given in the final accepted version of the paper.

Editorial policy

Submission of a paper implies that it has neither been published previously nor submitted for publication elsewhere. After the paper has been accepted, the corresponding author will be responsible for page formatting, page proof and signing off for printing on behalf of other co-authors. The corresponding author will receive one hardcopy issue in which the paper is published free of charge.

Manuscript preparation

The following points should be observed when preparing a manuscript besides being consistent in style, spelling, and the use of abbreviations. Authors are encouraged to download manuscript template from the IJoSI website, http://www.ijosi.org.

1. *Language.* Paper should be written in English except in some special issues where Chinese may be accepTable Each paper should contain an abstract not exceeding 200 words. In addition, three to five keywords should be provided.
2. *Manuscripts.* Paper should be typed, single-column, double-spaced, on standard white paper margins: top = 25mm, bottom = 30mm, side = 20mm. (The format of the final paper prints will have the similar format except that double-column and single space will be used.)
3. *Title and Author.* The title should be concise, informative, and it should appear on top of the first page of the paper in capital letters. Author information should not appear on the title page; it should be provided on a separate information sheet that contains the title, the author’s (authors') name(s), affiliation(s), e-mail address(es).
4. *Headings.* Section headings as well as headings for subsections should start front the left-hand margin.
5. *Mathematical Expressions.* All mathematical expressions should be typed using Equation Editor of MS Word. Numbers in parenthesis shall be provided for equations or other mathematical expressions that are referred to in the paper and be aligned to the right margin of the page.
6. *Tables and Figs.* Once a paper is accepted, the corresponding author should promptly supply original copies of all drawings and/or tables. They must be clear for printing. All should come with proper numbering, titles, and descriptive captions. Fig (or table) numbering and its subsequent caption must be below the Fig (or table) itself and as typed as the text.
7. *References.* Display only those references cited in the text. References should be listed and sequenced alphabetically by the surname of the first author at the end of the paper. For example:

Altshuller, G. (1998). *40 Principles: TRIZ Keys to Technical Innovation*, Technical Innovation Center.
Sheu, D. & Lee, H. (2011). A Proposed Process for Systematic Innovation, International Journal of Production Research, Vol. 49, No. 3, 2011, 847-868.

The International Journal of Systematic Innovation
Journal Order Form

Organization Or Individual Name	
Postal address for delivery	
Person to contact	Name: e-mail: Position: School/Company:
Order Information	**I would like to order ___ copy(ies) of the** *International Journal of Systematic Innovation***:** **Period Start: 1st/ 2nd half ____, Year:____(Starting 2010)** **Period End : 1st/ 2nd half ____, Year:** **Price:** **Institutions: US $150 (yearly) / NT 4,500 (In Taiwan only)** **Individuals: US $50 (yearly) / NT 1500 (In Taiwan only)** (Local postage included. International postage extra) **E-mail to**: IJoSI@systematic-innovation.org or fax: +886-3-572-3210 Air mail desired □ (If checked, we will quote the additional cost for your consent)
Total amount due	**US$**

Payment Methods:

1. **Credit Card (Fill up the following information and e-mail/ facsimile this form to The Journal office indicated below)**
2. **Bank transfer**
3. **Account:** The Society of Systematic Innovation
4. **Bank Name:** Mega International Commercial BANK
5. **Account No:** 020-53-144-930
6. **SWIFT Code:** ICBCTWTP020
7. **Bank code**：017-0206
8. **Bank Address:** No. 1, Xin'an Rd., East Dist., Hsinchu City 300, Taiwan (R.O.C.)

VISA / Master/ JCB/ AMERICAN Cardholder Authorization for Journal Order

Card Holder Information

Card Holder Name	(as it appears on card)			
Full Name (Last, First Middle)				
Expiration Date	/ (month / year)	Card Type	□ VISA □ MASTER □ JCB	
Card Number	□□□□-□□□□-□□□□-□□□□		Security Code	□□□ 
Amount Authorized		Special Messages		
Full Address (Incl. Street, City, State, Country and Postal code)				

Please Sign your name here _____ (same as the signature on your card)

The Society of Systematic Innovation

6 F, #352, Sec. 2, Guanfu Rd,
Hsinchu, Taiwan, 30071, R.O.C.