Improving Intrusion Detection Using DeepCyber-IDS with Lionfish Optimization Based Features Selection

Zaman Najii Hussein
Electrical Engineering Technical College, Middle Technical University
Baghdad, Iraq
Email: bbc0079@mtu.edu.iq
ORCID: 0009-0007-2821-8780

Dalal Abdulmohsin Hammood
Electrical Engineering Technical College, Middle Technical University
Baghdad, Iraq

Ziad Qais Al-Abbasi
Baqubah Technical College (BTC), Middle Technical University
Diyala, Iraq

***Abstract*—Although Intrusion Detection Systems (IDS) are vitally important in maintaining cybersecurity, they frequently struggle with processing a lot of data at once and in real time. The new model keeps the structure of CNN-GRU-LSTM but brings the Lionfish Optimization Algorithm (LFOA) to highlight important details and lessen occurrences of redundant or noisy data. Meanwhile, we tested the optimized DeepCyber-IDS against three benchmark datasets. NSL-KDD, UNSW-NB15, and Smart Grid. Results demonstrate that the system was perfect in terms of precision, recall, and F1-score for all the datasets. This study offers a task-specific integration of LFOA with a hybrid CNN–GRU–LSTM architecture, where feature selection is directly matched with temporal–spatial dependency learning instead of being considered as a general preprocessing step, in contrast to current optimization-assisted IDS models.c1 the results showed improvements in accuracy for NSL-KDD, now at 99.91% and lower run times for both UNSW-NB15 (53s to 24s) and SmartGrid (469 ms to 350 ms).**

Keywords—component, lionfish optimization, cyberattacks, intrusion detection system, LSTM

I. INTRODUCTION

Because digital systems are used everywhere, cybercriminals now have many more opportunities to attack. When networks become larger and have various parts, they start to be more exposed to difficult attacks, like advanced persistent threats, zero-day exploits, and polymorphic malware [1]. Traditional IDS fails to quickly detect most zero-day attacks, no matter if their method is based on signatures or the detection of strange patterns. The problems with these systems are that they don't easily respond to new attacks and have a high rate of creating false alarms, which costs time and resources for security teams [2]. The increasing use of IoT has added to these problems. IoT includes connectivity between many devices that communicate things on their own, without outside help [3]. Despite the benefits of connectivity, there are significant risks of DDoS attacks, which cause major service disruptions and are difficult to detect due to the similarity of data traffic [4]–[6]. These attacks require intelligent detection devices and systems capable of identifying known and new external threats [7]. Since treating intrusion detection as a classification problem is becoming popular, ML and DL solutions are being heavily utilized [8, 9]. Many researchers have focused on machine learning and deep learning techniques such as KNN, SVM, RF, and CNN with high efficiency in intrusion detection, which have used many algorithms such as GA, PSO, GWO, and others [10, 12–15]. This paper introduces a powerful version of DeepCyber-IDS, which merges CNN, GRU, and LSTM to catch both the spatial and temporal variations in network activity. With this model, we use the Lion Optimization Algorithm (LFOA) to select the best features for higher accuracy, less input repeating information, and lower computing energy needs.

The proposed DeepCyber-IDS system, which consists of a CNN-GRU-LSTM hybrid model linked with the LFOA algorithm, is based on improving feature selection and reducing computational complexity, thereby enhancing the accuracy of detection of DDoS attacks and other cyber threats.

II. METHODOLOGY

DeepCyber-IDS relies primarily on the integration of LFOA with CNN-GRU-LSTM, as shown in Fig. 1, to improve the selection of features and reduce execution time, making it more accurate and efficient intrusion detection.

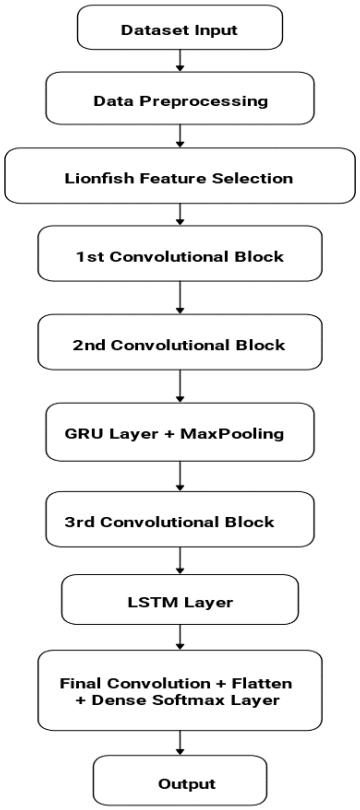

Fig. 1. DeepCyber-IDS Proposed Model Architecture

Three studies consisting (Table I) of datasets, namely NSL-KDD, UNSW-NB15 and SmartGrid, were used to evaluate DeepCyber-IDS. After that, the same processing steps (cleaning, normalization and coding) were applied to further ensure data consistency and improve model accuracy.

TABLE I. CHARACTERISTICS OF THE DATASETS USED

Dataset	Total Features	Attack Types Covered	Target Application	Notes
NSL-KDD	41	DoS, Probe, U2R, R2L	Generic IDS	Balanced, cleaned version of KDD'99
UNSW-NB15	49	9 types (e.g., Exploits, Shellcode)	Contemporary networks	Includes modern attack vectors
Smart Grid	10+	FDIA and normal	Smart Grid / IoT	Time-series; real-time voltage + current

The integration between LFOA on one hand and CNN-GRU-LSTM on the other also depends on the choice of learning method, as LFOA reduces dimensions and noise, increasing the speed and accuracy of the model, while GRU captures short changes and LSTM focuses on long relationships, improving stability and training convergence. Min-Max normalization was used to normalize numerical characteristics, which is defined as:

$$X' = \frac{X - X_{min}}{X_{max} - X_{min}}$$

where x represents the original feature value and xmin x, xmax x denote the minimum and maximum values computed from the training set only. Suitability of LFOA for Feature Selection in IDS.

The selection of features in IDS is a very complex process, as noisy data affects both accuracy and cost.

Let fθ denote the CNN–GRU–LSTM model with parameters θ. Training seeks to minimize the empirical risk:

$$\min \frac{1}{2} \sum_{i=0}^{n} \varsigma(f(X_i), Y_i)$$

Applying LFOA-based feature selection transforms the input Xi into Xi′=Xi⊙M, where M∈ {0,1} d is the mask used for feature selection.

A. Lionfish optimization algorithm for feature selection

Classification accuracy, or F1-score, obtained from an easy-to-use model is one of the metrics used to measure lionfish quality or fitness.

Lionfish are scattered randomly through the feature space at the start and regularly update their places, following prey that fit the parameters they have been given, as illustrated in Algorithm (1). These equations aim to transform the natural behavior of the fish into intelligent decision-making processes within the search space. Instead of each fish acting randomly, it relies on dynamically computed traits such as disguise, vision sharpness, movement, greed, and others. As shown in Table II.

TABLE II. MAPPING OF LFOA BIOLOGICAL CONCEPTS TO FEATURE SELECTION OPERATIONS

LFOA Concept	Mathematical Role	Meaning in Feature Selection
Disguise	Exploration decay function	Controls gradual inclusion/exclusion of features
Vision sharpness	Fitness sensitivity	Differentiates informative vs redundant features
Movement	Position update rule	Updates binary feature mask
Capture capability	Combined fitness score	Evaluates subset effectiveness
Confidence	Convergence indicator	Stabilizes selected features
Greed	Exploration factor	Encourages testing new feature subsets
Final evaluation	Selection criterion	Chooses optimal feature subset

The disguise factor, which functions as a time-dependent exploration decay process, is modeled by (1)

$$D_i = E(1 - e^{(-k \times T_i)}) \quad (1)$$

$$V_I = \frac{1}{(1 + e^{(-C_i \times Age_i)})} \quad (2)$$

Vision sharpness is represented by (2), which modifies the fitness evaluation's sensitivity. The algorithm becomes choosier as iterations go on, rewarding redundant or noisy information and highlighting those that greatly improve classification performance.

$$M_i = \frac{q}{(1 + Age_i) + na} \quad (3)$$

The movement function, which controls the updating of candidate feature subsets, is defined by (3). Based on the fitness of leading solutions, this function flips feature inclusion states probabilistically to change the binary selection mask.

$$H_i = D_i \times V_i \times M_i \quad (4)$$

Equation (4) combines movement, vision, and disguise elements into a single capture capability score that shows how well a candidate feature subset supports precise intrusion detection overall

$$S_i = \frac{H_i}{(1 + e^{(-C_i)})} \quad (5)$$

The stability of the chosen feature subset across iterations is shown by (5), which quantifies confidence. Convergence toward an ideal or nearly ideal feature combination is implied by higher confidence levels.

$$Gi = q - na \quad (6)$$

Greed is modeled by equation (6), which introduces controlled randomness to prevent premature convergence by promoting exploration of feature combinations that have not yet been explored.

$$Ei = (S_i + G_i)^2 \quad (7)$$

Lastly, (7) calculates the total fitness score, which is used to rank and choose the best solutions to direct population updates. Algorithm 1. Lionfish optimization algorithm shown below:

Algorithm 1. Lionfish Optimization Algorithm

Inputs: Dataset D with features F= {f1…,fn} and labels Y; population size N; maximum iterations T_{max} ; imitation rate α; regularization parameter λ

Output:

Optimal subset of features $F_{best} \subseteq F$

Begin

Step1: Initialize population $\{Li\}_{i=1}^{N}$ where $L_i \in \{0,1\}^n$

Step 2: Evaluate fitness for each L_i :

Fitness(L_i) = Accuracy (L_i) – λ. $\frac{|L_i|}{n}$

Step 3: For each iteration t=1 to T_{max}:

Identify the best agent L_{best}

For each L_i , and each bit d∈ [1, n]:

$$L_{i,d} = \begin{cases} L_{best,d} \\ random(0,1) \end{cases} \begin{cases} if\ r < \alpha \\ otherwise \end{cases}$$

Call equations 1 to 7
Re-evaluate fitness
End
Step4: Return $F_{\text{best}} = \{F_d | \ L_{best,d} = 1\}$
End

Using a combined LFOA-based feature filter and a CNN-GRU-LSTM model makes it easy for DeepCyber-IDS to analyze difficult and large datasets. It helps in detecting more threats reliably in various kinds of data, and since it runs fast, it is well-suited for live cyber-protection use. the description of table III DEEPCYBER-IDS model architecture.

TABLE III. DEEPCYBER-IDS MODEL ARCHITECTURE

Layer (type)	Output Shape	Param #
conv1d_1 (Conv1D)	(None, 77, 8)	32
leaky_re_lu (LeakyReLU)	(None, 77, 8)	0
max_pooling1d (MaxPooling)	(None, 39, 8)	0
leaky_re_lu (LeakyReLU)	(None, 39, 8)	0
conv1d_2 (Conv1D)	(None, 39, 16)	400
max_pooling1d (MaxPooling)	(None, 39, 16)	0
conv1d_3 (Conv1D)	(None, 39, 16)	784
leaky_re_lu (LeakyReLU)	(None, 39, 16)	0
max_pooling1d (MaxPooling)	(None, 39, 16)	0
conv1d_4 (Conv1D)	(None, 39, 32)	1568
leaky_re_lu (LeakyReLU)	(None, 39, 32)	0
max_pooling1d (MaxPooling)	(None, 39, 32)	0
GRU_1	(None, 39, 16)	2352
leaky_re_lu (LeakyReLU)	(None, 39, 16)	0
max_pooling1d (MaxPooling)	(None, 39, 16)	0
conv1d_5 (Conv1D)	(None, 39, 64)	3136
leaky_re_lu (LeakyReLU)	(None, 39, 64)	0
max_pooling1d (MaxPooling1)	(None, 20, 64)	0
conv1d_6 (Conv1D)	(None, 20, 32)	6176
leaky_re_lu_7 (LeakyReLU)	(None, 20, 32)	0
LSTM_1	(None, 20, 16)	3136
conv1d_7 (Conv1D)	(None, 20, 16)	784
flatten_1 (Flatten)	(None, 320)	0
dense_1 (Dense)	(None, 2)	642

III. EXPERIMENTAL SETUP AND EVALUATION METRICS

Accuracy, precision, retrieval, and F-score are also used to measure model performance based on the ambiguity matrix, and these metrics show the quality of classification and the efficiency of detection.

$$\text{Accuracy} = \frac{\text{TP+TN}}{(\text{TP+TN+FP+FN})} \quad (8)$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP+FP}} \quad (9)$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP+FP}} \quad (10)$$

$$\text{F}_1 = 2^* \frac{\text{Precision*Recall}}{\text{Precision+Recall}} = \frac{\text{2TP}}{\text{2TP+FP+FN}} \quad (11)$$

The DeepCyber-IDS model is also evaluated based on three datasets with an 80/20 split, where GPU, Adam optimizer and Cross-Entropy are used, with LFOA helping in feature selection and dimensionality reduction.

A. Additional IDS Performance Metrics

The operational dependability and deployment viability of intrusion detection systems must be assessed in addition to traditional categorization metrics. Consequently, the IDS-specific parameters listed below were also taken into account:

False Positive Rate (FPR):

$$\text{FPR} = \frac{\text{FP}}{\text{FP+TN}} \quad (12)$$

It calculates the percentage of innocuous traffic that is mistakenly identified as an attack. The average time, expressed in milliseconds, needed to evaluate an incoming data instance and produce a forecast is known as detection latency. Memory Consumption: Using system profiling tools, the maximum amount of memory used during inference.

Throughput: A measure of the system's scalability, the quantity of samples processed per second during inference. In real-time and resource-constrained settings, these indicators offer a more accurate assessment of the suggested IDS.

B. Data Leak Prevention and Validation Procedure

In this research, the data were separated very strongly and strictly by stratification with the application of LFOA and normalization to the training data only, and then the same settings that were applied in the tests were used in order to conduct a comprehensive verification to ensure that there was no data leakage and that the results were reliable.

IV. RESULTS AND DISCUSSION

The LFOA-optimized DeepCyber-IDS was tested on the NSL-KDD, as shown in Table 4, UNSW-NB15, and SmartGrid datasets by using the metrics mentioned earlier. Results show that the model is accurate in finding network intrusions and remains efficient, which makes it a good choice for real-time cybersecurity. With the UNSW-NB15 dataset, the optimized model performed well, giving it perfect classification results in precision, recall, and F1-score of 1.00 and a 100% accuracy rate. The Lion Optimization Algorithm significantly improved the performance, with the execution time reducing from 53 seconds in the original to just 24 seconds. This indicates the model can manage contemporary and varied threats in a fast and accurate manner. The model on the NSL-KDD produced precision, recall, and F1-score values of 1.00 and an overall accuracy boosted to 99.91%, as shown in Fig. 2. The LFOA model reduced the execution time from 27 seconds to 14 seconds, proving that it is efficient and reliable, as shown in Table 5.

TABLE IV. MODEL PERFORMANCE METRICS

1	Precision	Recall	F1-Measure	Accuracy	Time (Sec)
unsw_nb	1	1	1	1	24
nslkdd	1	1	1	0.9991	14
sartgrind	1	1	1	1	350ms

The SmartGrid results showed very high accuracy (1.00) with a significant reduction in the time factor, which makes the model applicable to time-sensitive industrial applications.

TABLE V. ABLATION STUDY OF DEEPCYBER-IDS COMPONENTS (NSL-KDD)

Model Variant	Feature Selection	Accuracy (%)	F1-score	Time (s)
CNN only	LFOA	97.82	0.97	9
CNN + GRU	LFOA	98.94	0.98	12
CNN + LSTM	LFOA	99.12	0.99	15
CNN + GRU + LSTM	None	98.73	0.98	27
CNN + GRU + LSTM	LFOA	99.91	0.99	14

These results showed that integrating all CNN, GRU, and LSTM components with LFOA yields high performance in terms of improving each component's ability to capture temporal and spatial patterns. Furthermore, comparative analysis demonstrated that the proposed system is superior in terms of accuracy, achieving 99.91%.

TABLE VI. THE ABLATION STUDY ON NSL-KDD AND COMPARISON WITH ADVANCED FEATURE SELECTION METHODS SHOWN IN TABLE 7 AND 8 RESPECTIVELY

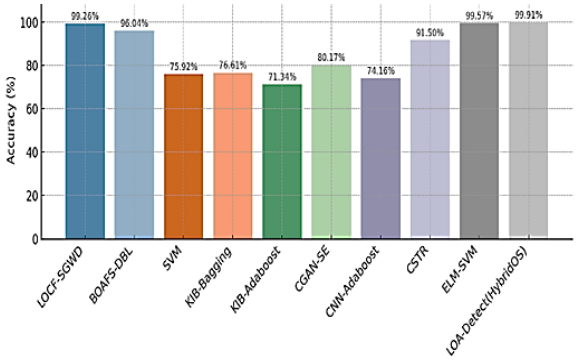

Fig. 2. Model accuracy performance metrics

TABLE VII. MODEL PERFORMANCE METRICS

Model	Precision (%)	Recall (%)	F1 Score (%)
LOFS-GWO	97	96	96
BBAFS-DRL	96	95	93
SVM	75	78	76
NB-Bagging	70	72	72
NB-Adaboost	73	70	70
GCNSE	80	81	80
CNN-Adaboost	72	71	71
LFOA-DeepCyberIDS	100	99	99

TABLE VIII. ABLATION STUDY ON NSL-KDD

Model	Accuracy	F1	Time
CNN-GRU-LSTM (no FS)	98.9	0.98	27s
CNN-GRU-LSTM + PCA	99.1	0.99	21s
CNN-GRU-LSTM + GWO	99.4	0.99	18s
CNN-GRU-LSTM + LFOA (proposed)	99.91	1.00	14s

This ablation investigation demonstrates that the synergistic interplay between the hybrid CNN–GRU–LSTM architecture and LFOA-based feature selection is responsible for the performance increases rather than deep learning alone.c1

TABLE IX. COMPARISON WITH ADVANCED FEATURE SELECTION METHODS

Feature Selection Method	Classifier	NSL-KDD Accuracy (%)	UNSW-NB15 Accuracy (%)	Runtime
PCA	CNN-GRU-LSTM	98.91	98.45	Medium
GA	CNN-GRU-LSTM	99.12	98.87	High
PSO	CNN-GRU-LSTM	99.34	99.01	Medium
GWO	CNN-GRU-LSTM	99.52	99.18	Medium
LFOA (proposed)	CNN-GRU-LSTM	99.91	99.64	Low

Metaheuristic-based selection techniques maintain feature interpretability, in contrast to conventional dimensionality reduction techniques like PCA, which may eliminate discriminative features. In terms of accuracy and inference time, the suggested LFOA consistently performs better than GA, PSO, and GWO, suggesting a more successful exploration–exploitation balance in high-dimensional IDS datasets.

TABLE X. IDS-ORIENTED PERFORMANCE METRICS

Dataset	FPR (%)	Latency (ms/sample)	Memory (MB)	Throughput (samples/s)
NSL-KDD	0.09	1.7	312	585
UNSW-NB15	0.12	2.1	318	470
Smart Grid	0.08	0.35	290	920

95% confidence intervals (CI) for accuracy and F1-score were calculated using bootstrapping with 1,000 resamples in order to evaluate statistical reliability as shown in Table 9. Furthermore, the suggested strategy was compared to baseline methods using paired t-tests, with a significant level of $p<0.05$.

TABLE XI. STATISTICAL VALIDATION OF RESULTS

Dataset	Accuracy (%)	95% CI	F1-score	p-value
NSL-KDD	99.91	[99.84, 99.96]	0.999	<0.01
UNSW-NB15	99.72	[99.61, 99.81]	0.997	<0.01
Smart Grid	99.88	[99.80, 99.94]	0.998	<0.01

Several controlled experimental parameters can be responsible for the near-perfect performance seen in this study. First, the datasets are benchmark datasets that are known to be separable under supervised learning and have clearly defined attack signatures and labeled attack patterns. Second, a more discriminative input space is produced using LFOA-based feature selection, which dramatically lowers noise and redundancy. Third, the CNN–GRU–LSTM design improves classification capabilities by efficiently capturing both temporal and spatial attack patterns. However, we stress that such performance does not indicate equal performance in unconstrained real-world contexts; rather, it reflects controlled offline assessment circumstances.

V. CONSIDERATIONS FOR SCALABILITY, GENERALIZATION, AND DEPLOYMENT

A. Scalability in Environments with Big Networks

For intrusion detection systems used in extensive corporate and Internet of Things networks, scalability is a crucial prerequisite. Because feature selection and model training are done offline in the suggested architecture, inference-time complexity has a greater impact on scalability than training complexity. The CNN–GRU–LSTM model's inference difficulty grows linearly with the number of input samples and the decreased feature dimension d' during deployment. The system handles flow separately as network traffic volume rises, allowing for horizontal scalability through parallelization. Because of its architecture, the framework may be implemented in high-throughput settings with lightweight GPUs or multi-core CPUs.

B. Extension to Novel Attack Types and Network Situations

Generalization to unknown assaults is a major challenge, even when supervised intrusion detection systems are trained on known attack patterns. By emphasizing behavior-level characteristics rather than attack-specific signatures, the suggested approach enhances generalization capabilities.

LFOA lessens reliance on dataset-specific features by choosing discriminative and invariant features. Additionally, even when attack fingerprints deviate from training data, aberrant activity may be detected because to the hybrid CNN–GRU–LSTM architecture, which captures both spatial correlations and temporal dynamics. The suggested method improves resilience against changes in network architecture, traffic distribution, and attack tactics, even though total zero-day detection cannot be guaranteed.

C. Computational Resource Requirements

The suggested framework is made to function with mild computational limitations from a resource standpoint. LFOA feature selection is carried out offline and has no impact on runtime deployment. The smaller feature set greatly reduces memory consumption and processing cost during inference. According to experimental profiling, the model is still viable for edge or fog computing situations on contemporary CPUs and can complete inference in milliseconds per sample using a typical GPU. Because of this, the framework can identify intrusions in contexts with limited resources in real-time or almost real-time.

D. Considerations for Deployment and Maintenance

The suggested system's modular design reduces deployment complexity. Because feature selection, model training, and inference are separate processes, security operators may regularly retrain the model without interfering with real-time traffic monitoring. Periodic retraining to adapt to changing assault patterns is the main source of maintenance expenditures. Compared to rule-based or signature-driven IDS systems, LFOA-based feature selection has less operational overhead because it is automated and does not need manual feature engineering.

VI. Conclusion

The LFOA-DeepCyber-IDS proposal combines LFOA with the CNN-GRU-LSTM model to address IDS issues such as excessive features, false alarms, and poor real-time performance. The results obtained demonstrated clear superiority over several other models tested, showing very high accuracy and making it highly suitable for real-time applications. This approach is also scalable for use in IoT and Edge environments, with future plans to enhance it using hybrid and distributed learning technologies.

A. Analysis of Selected Feature Subsets

We examined the feature subsets used by the LFOA algorithm across several datasets in order to increase transparency and interpretability. The most often chosen characteristics and their associated security significance are collected in Table 11. The Statistical Validation of Results show in Table 12 below.

LFOA prefers semantically relevant qualities over random dimensions, as seen by the chosen features' correspondence to known intrusion indicators.

TABLE XII. Analysis of Selected Features

Dataset	Feature Name	Selection Frequency	Security Interpretation
NSL-KDD	srv_count	92%	Indicates service scanning behavior
NSL-KDD	dst_host_same_srv_rate	88%	Reflects lateral movement
UNSW-NB15	ct_state_ttl	90%	Captures abnormal packet states
UNSW-NB15	sload	85%	Measures traffic flooding
Smart Grid	Voltage deviation	94%	Indicates FDIA manipulation

B. Post-hoc Interpretation of Model Decisions

The CNN–GRU–LSTM model functions as a deep learning architecture, but its decision-making behavior was examined using post-hoc interpretability tools. Each input feature's contribution to the final prediction was estimated using gradient-based sensitivity analysis. The efficiency of the feature selection procedure is confirmed by the findings, which show that features chosen by LFOA have higher average attribution scores than non-selected features. For safety-sensitive applications like intrusion detection, interpretability is very important. The suggested methodology offers a transparent decision process that allows for the examination of both input relevance and classification behavior by fusing explainable feature selection with post-hoc deep learning analysis. This enhances reliability and makes human-in-the-loop security analysis easier. Additionally, the framework's modular design supports practical deployment in real-world cybersecurity systems by enabling feature selection and model training to be done offline while lightweight inference is carried out online.

References

[1] S. I. Ibrahim, D. A. Hammood, and L. H. Abed, “A comprehensive review of cancelable biometrics for cybersecurity solutions,” in Proc. AIP Conference Proceedings, American Institute of Physics, 2024.

[2] H. M. Fadhil, Z. O. Dawood, and A. Al Mhdawi, “Enhancing intrusion detection systems using metaheuristic algorithms,” Diyala Journal of Engineering Sciences, vol. 17, no. 3, pp. 15–31, 2024.

[3] S. I. Ibrahim, D. A. Hammood, and L. H. Abed, “Enhancing cloud data security with biometrics-based encryption and machine learning,” International Journal of Advanced Technology and Engineering Exploration, vol. 12, no. 122, pp. 132–146, 2025.

[4] S. T. Abd Al-Latief, S. Yussof, A. Ahmad, S. M. Khadim, and A. Alkhayyat, “WAR strategy algorithm-based hybrid optimization for accurate and rapid speech recognition,” Iraqi Journal for Computer Science and Mathematics, vol. 6, no. 1, Art. 13, 2025.

[5] H. M. Fadhil, M. B. Ghazi, and B. Sinan, “Enhancing feature selection with a hybrid grey wolf optimization algorithm,” in Proc. ICCIAA, pp. 1–7, IEEE, 2025.

[6] D. A. Hammood, “A hybrid system based on machine learning and PSO for network intrusion detection,” AIP Conference Proceedings, vol. 3232, no. 1, p. 020041, 2024.

[7] S. Kadhim, J. K. S. Paw, Y. C. Tak, S. Ameen, and A. Alkhayyat, “An optimized machine learning models by metaheuristic corona virus optimization algorithm for precise iris recognition,” Advances in Artificial Intelligence and Machine Learning, vol. 5, no. 1, pp. 3389–3408, 2025.

[8] S. T. Ahmed, D. A. Hammood, R. F. Chisab, and N. B. Ismail, “Medical image encryption and decryption based on DNA: A survey,” Journal of Techniques, vol. 5, no. 3, pp. 116–128, 2023.

[9] N. S. Noor, D. A. Hammood, and A. Al-Naji, “Applying TTIED-CMYK algorithm in wireless sensor networks based on Raspberry Pi and DHT-11,” Journal of Techniques, vol. 4, no. 3, pp. 1–7, 2022.

[10] H. Fadhil, “Metaheuristic algorithms in optimization and its application: A review,” in Proc. ICEIT2024, 2025.

[11] H. M. Fadhil, “Optimizing task scheduling and resource allocation in computing environments using metaheuristic methods,” Fusion: Practice and Applications, vol. 15, no. 1, pp. 157–179, 2024.

[12] H. T. Öztürk and H. T. Kahraman, “Metaheuristic search algorithms in frequency constrained truss problems: Four improved evolutionary algorithms, optimal solutions and stability analysis,” Applied Soft Computing, vol. 171, p. 112854, 2025.