Metaheuristic Algorithms with Applications

www.maa.reapress.com

Metaheur. Algor. Appl. Vol. 2, No. 3 (2025) 309–323.

Paper Type: Original Article

Metaheuristic Optimization Algorithms for Cybersecurity: A Multi-Domain Experimental Study on Intrusion Detection, Cryptographic Key Optimization, and Malware Classification

Amir Ekbatanifard*

Department of Computer Engineering, La.C., Islamic Azad University, Lahijan, Iran; amirekbatani@gmail.com.

Citation:

Received: 16 November 2024 *Revised: 27 February 2025* *Accepted: 06 May 2025*	Ekbatanifard, A. (2025). Metaheuristic optimization algorithms for cybersecurity: A multi-domain experimental study on intrusion detection, cryptographic key optimization, and malware classification. *Metaheuristic algorithms with applications*, *2*(3), 309-323.

Abstract

The escalating sophistication of cyber threats demands adaptive, intelligent security mechanisms that transcend the limitations of conventional rule-based and signature-driven approaches. This paper presents a comprehensive metaheuristic-based security optimization framework that addresses three critical cybersecurity problems simultaneously: 1) Network Intrusion Detection System (NIDS) feature selection and classifier optimization using Genetic Algorithm (GA), Particle Swarm Optimization (PSO), Grey Wolf Optimizer (GWO), Whale Optimization Algorithm (WOA), and Harris Hawks Optimization (HHO), 2) cryptographic Substitution-Box (S-box) generation and key scheduling optimization for symmetric ciphers, and 3) malware classification via metaheuristic-optimized ensemble learning. Extensive experiments were conducted on four benchmark datasets — NSL-KDD, UNSW-NB15, and CICIDS-2017 for intrusion detection, and Malimg for malware classification — under rigorous experimental conditions including 10-fold cross-validation and 30 independent runs per configuration. In the intrusion detection domain, GWO-Random Forest (RF) achieved the highest accuracy of 99.41% on NSL-KDD with a 78.0% feature reduction, selecting only 9 of 41 original features. For cryptographic S-box generation, HHO produced S-boxes with an average nonlinearity score of 112 (maximum possible: 120), approaching the quality of the Advanced Encryption Standard (AES) standard S-box while exhibiting a differential uniformity of 6. In the malware classification domain, PSO-optimized ensemble classifiers attained an F1-score of 98.76% on the Malimg dataset. Statistical significance was confirmed via Friedman test ($\chi^2 = 18.93$, $p < 0.001$) and pairwise Wilcoxon signed-rank tests. This study provides the first comprehensive multi-domain comparison of modern metaheuristic algorithms across the cybersecurity spectrum, offering practitioners evidence-based guidance for algorithm selection in diverse security applications.

Keywords: Metaheuristic algorithms, Cybersecurity, Intrusion detection, Feature selection, Cryptography, S-box optimization, Malware classification, Network security, Grey wolf optimizer, Harris hawks optimization.

1 | Introduction

The contemporary digital landscape is characterized by an unprecedented volume and sophistication of cyber threats. According to recent cybersecurity reports, the global cost of cybercrime is projected to exceed $10.5

Corresponding Author: amirekbatani@gmail.com

https://doi.org/10.48313/maa.v2i3.52

trillion annually by 2025, with ransomware attacks occurring every 11 seconds and zero-day exploits increasing by over 60% compared to the previous decade [1]. The proliferation of Internet-of-Things (IoT) devices, cloud computing infrastructures, and 5G networks has dramatically expanded the attack surface available to adversaries, creating an urgent need for intelligent, adaptive security mechanisms capable of operating at scale and in real time.

Traditional cybersecurity approaches, while foundational, exhibit critical limitations when confronted with modern threat landscapes. Signature-based Intrusion Detection Systems (IDS) — exemplified by Snort [2] and Suricata — rely on pre-defined attack patterns and are inherently incapable of detecting zero-day exploits, polymorphic malware, and Advanced Persistent Threats (APTs) that deviate from known signatures. Similarly, conventional cryptographic designs, while mathematically robust under ideal conditions, may exhibit vulnerabilities to side-channel attacks, fault injection, and differential power analysis when implementation parameters are not optimally configured. Static ensemble classifiers for malware detection suffer from the curse of dimensionality and frequently exhibit degraded performance when confronted with novel malware families not represented in training data.

Metaheuristic optimization algorithms offer a compelling paradigm for addressing these limitations. Inspired by natural phenomena — evolutionary processes, swarm intelligence, and predatory behaviors — metaheuristics provide population-based, gradient-free optimization strategies capable of navigating vast, complex, and multimodal search spaces. Their applicability to cybersecurity is multifaceted: in intrusion detection, metaheuristics can optimize feature selection to reduce dimensionality while preserving discriminative power [3]; in cryptography, they can generate highly nonlinear Substitution-Boxes (S-boxes) and optimize key scheduling parameters [4]; and in malware classification, they can tune ensemble classifier architectures and weight configurations to maximize detection performance [5].

Despite significant individual contributions, the existing literature lacks a comprehensive, multi-domain study that systematically compares modern metaheuristic algorithms across diverse cybersecurity applications under uniform experimental conditions. Prior studies have typically focused on a single security domain with one or two algorithms, making cross-domain and cross-algorithm comparisons difficult. Furthermore, recently proposed algorithms — particularly Harris Hawks Optimization (HHO) [6] and the Grey Wolf Optimizer (GWO) [7] — have demonstrated superior performance in various optimization benchmarks but remain underexplored in cybersecurity contexts.

This paper addresses these gaps through the following contributions:

I. Comprehensive multi-domain comparison: five metaheuristic algorithms (Genetic Algorithm (GA), Particle Swarm Optimization (PSO), GWO, Whale Optimization Algorithm (WOA), and HHO) are systematically evaluated across three distinct cybersecurity domains — intrusion detection, cryptographic S-box generation, and malware classification — under identical experimental protocols.

II. Novel GWO-based wrapper feature selection for IDS: a GWO-based wrapper method is proposed for simultaneous feature selection and Random Forest (RF) hyperparameter optimization in network intrusion detection, achieving 99.41% accuracy on NSL-KDD with a 78.0% feature reduction.

III. HHO-based S-box generator: a HHO approach is introduced for generating cryptographically strong 8×8 S-boxes, achieving a nonlinearity score of 112 with a differential uniformity of 6, approaching the quality of the Advanced Encryption Standard (AES) Rijndael S-box.

IV. PSO-optimized ensemble malware classifier: a PSO framework is presented for optimizing both the selection and weighting of base learners in an ensemble malware classification system, yielding a 98.76% F1-score on the Malimg dataset.

The remainder of this paper is organized as follows. Section 2 reviews related work across the three security domains. Section 3 presents the methodology, including problem formulations, algorithm configurations, and experimental design. Section 4 reports experimental results and statistical analyses. Section 5 discusses

findings, practical implications, and limitations. Section 6 concludes the paper and outlines future research directions.

2 | Literature Review

2.1 | Metaheuristics in Intrusion Detection

The application of metaheuristic algorithms to IDS has attracted substantial research attention over the past decade. Feature selection, in particular, has been identified as a critical preprocessing step for IDS, given that benchmark datasets such as NSL-KDD [8] and UNSW-NB15 [9] contain numerous redundant and irrelevant features that degrade classifier performance and increase computational overhead.

Khammassi and Krichen [10] proposed a GA-based logistic regression wrapper for feature selection in IDS, achieving 99.0% accuracy on the KDD Cup 99 dataset with a 60% feature reduction. However, their study was limited to a single classifier and the deprecated KDD Cup 99 dataset. Elmasry et al. [11] combined PSO with Deep Neural Networks (DNN) for intrusion detection on CICIDS-2017, reporting 98.8% accuracy but with prohibitive training times exceeding 12 hours. Asghari Varzaneh and Hosseini [12] introduced a Lévy-opposition equilibrium optimization algorithm for feature selection in IDS, achieving 97.6% accuracy on UNSW-NB15 with only 10.8 features selected on average. Almomani et al. [13] evaluated multiple metaheuristic algorithms for IDS feature selection but did not include recently proposed algorithms such as GWO and HHO. Kasongo and Sun [14] utilized a filter-wrapper hybrid with GA for feature selection on UNSW-NB15, achieving 87.4% accuracy with a feed-forward neural network. More recently, Abdel-Basset et al. [15] proposed a binary WOA for IDS feature selection, achieving 97.2% accuracy on NSL-KDD with 15 selected features. Thaher et al. [16] evaluated binary variants of swarm intelligence algorithms for wrapper-based feature selection in IDS. Despite these advances, no prior study has simultaneously compared GA, PSO, GWO, WOA, and HHO for IDS feature selection under uniform experimental conditions with identical classifiers and evaluation protocols.

2.2 | Metaheuristics in Cryptography

Substitution boxes are fundamental nonlinear components in symmetric block ciphers, and the quality of S-box design directly impacts cipher resistance to linear and differential cryptanalysis. The AES Rijndael S-box [17]— constructed through algebraic methods over GF(2^8) — achieves the maximum possible nonlinearity of 112 for 8×8 bijective mappings and serves as the gold standard for S-box quality evaluation.

Metaheuristic approaches to S-box generation have been explored as alternatives to algebraic construction, offering the advantage of producing diverse S-boxes without relying on a single mathematical structure. Farah et al. [4] proposed a chaos-based S-box design using a Lorenz attractor combined with a GA post-optimization step, achieving nonlinearity scores between 104 and 108. Alzaidi et al. [18] applied a GA to optimize the affine transformation parameters in AES-like S-box construction, producing S-boxes with an average nonlinearity of 106.5. Kuznetsov et al. [19] recently demonstrated that evolutionary approaches with Walsh-Hadamard Spectrum cost functions can generate S-boxes with nonlinearity of 104, with a 100% success rate. Zamli et al. [20] utilized the Jaya algorithm for S-box generation, achieving nonlinearity values up to 108. Ahmad et al. [21] proposed a hybrid whale optimization approach for S-box construction, reporting nonlinearity of 109.25. However, recently proposed algorithms with strong exploitation capabilities — particularly HHO — remain largely unexplored for S-box optimization, despite their demonstrated superiority in combinatorial optimization benchmarks.

2.3 | Metaheuristics in Malware Analysis

Malware classification has evolved from signature-based detection to sophisticated machine learning and deep learning paradigms. Nataraj et al. [22] pioneered malware visualization by converting malware binaries to grayscale images and applying image classification techniques, introducing the Malimg dataset of 9,339

samples across 25 malware families. This dataset has since become a standard benchmark for malware classification research. Ucci et al. [5] presented a comprehensive survey of machine learning techniques for malware analysis, identifying feature selection and classifier optimization as key challenges. Panda et al. [23] proposed an ensemble 1D-CNN approach for imbalanced multiclass malware classification, achieving 0.90 F1-score on the Mal-API-2019 dataset. Ensemble methods have shown particular promise, as they combine diverse base classifiers to improve robustness and generalization [24]. However, the optimization of ensemble architectures — including base learner selection, weighting, and hyperparameter tuning — represents a combinatorial optimization problem well-suited to metaheuristic approaches, yet this intersection remains underexplored in the literature.

Table 1 summarizes key works at the intersection of metaheuristic algorithms and cybersecurity.

Table 1. Summary of related work on metaheuristic algorithms in cybersecurity.

Author(s)	Year	Algorithm	Security Domain	Dataset/Problem	Key Result
Khammassi and Krichen [10]	2017	GA	IDS	KDD Cup 99	99.0% accuracy, 60% feature reduction
Moustafa and Slay [9]	2015	—	IDS	UNSW-NB15	Benchmark dataset introduction
Elmasry et al. [11]	2020	PSO-DNN	IDS	CICIDS-2017	98.8% accuracy
Kasongo and Sun [14]	2020	GA-FFNN	IDS	UNSW-NB15	87.4% accuracy
Abdel-Basset et al. [15]	2023	Binary WOA	IDS	NSL-KDD	97.2% accuracy, 15 features
Thaher et al. [16]	2023	SI variants	IDS	NSL-KDD	Wrapper-based comparison
Asghari Varzaneh and Hosseini [12]	2024	BLOEO	IDS	UNSW-NB15	97.6% accuracy, 10.8 features
Ambusaidi et al. [3]	2016	MI-based filter	IDS	NSL-KDD, UNSW	Mutual-info feature selection
Almomani et al. [13]	2019	GA, PSO, ACO	IDS	NSL-KDD	Multi-algorithm comparison
Farah et al. [4]	2020	Chaos + GA	Cryptography	S-box generation	NL 104–108
Alzaidi et al. [18]	2018	GA-AES	Cryptography	S-box optimization	NL 106.5
Kuznetsov et al. [19]	2024	GA + WHS	Cryptography	S-box generation	NL 104, 100% success rate
Zamli et al. [20]	2021	Jaya	Cryptography	S-box generation	NL 108
Ahmad et al. [21]	2022	Hybrid WOA	Cryptography	S-box construction	NL 109.25
Daemen and Rijmen [17]	2002	Algebraic	Cryptography	AES Rijndael	NL 112 (gold standard)
Nataraj et al. [22]	2011	kNN	Malware	Malimg	Malware visualization pioneering
Ucci et al. [5]	2019	Survey	Malware	Multiple	ML malware analysis survey
Panda et al. [23]	2023	Ensemble 1D-CNN	Malware	Mal-API-2019	0.90 F1-score

3 | Methodology

3.1 | Problem Formulations

3.1.1 | Problem 1: Intrusion detection feature selection and classifier optimization

The intrusion detection problem is formulated as a binary feature selection optimization combined with classifier hyperparameter tuning. Let $F = \{f_1, f_2, \ldots, fn\}$ denote the full feature set with $|F| = n$ features,

and let S ⊆ F denote a selected feature subset. Each candidate solution is encoded as a binary vector x = (x_1, x_2, …, xn), where xi ∈ {0, 1} indicates whether feature fi is selected. The fitness function balances classification accuracy against feature parsimony:

$$\text{Fitness}(\mathbf{x}) = \alpha \cdot \text{Accuracy}(\mathbf{S}) + (1 - \alpha) \cdot (1 - |\mathbf{S}| / |\mathbf{F}|), \quad (1)$$

where Accuracy(S) is the 10-fold cross-validation accuracy of the classifier trained on the selected subset S, |S| = Σxi is the number of selected features, and α = 0.9 is a weighting coefficient that prioritizes classification performance while penalizing feature redundancy. The continuous positions generated by each metaheuristic are converted to binary via the sigmoid transfer function:

$$T(x) = 1 / (1 + e - x), \quad (2)$$

such that xi = 1 if rand() < T(xi), and xi = 0 otherwise.

3.1.2 | Problem 2: Cryptographic s-box optimization

An 8×8 S-box is a bijective mapping S: GF(2^8) → GF(2^8) represented as a permutation of integers {0, 1, …, 255}. Each candidate solution in the metaheuristic population encodes a complete permutation. The fitness function for S-box optimization is a multi-objective formulation combining four cryptographic quality metrics:

$$\text{Fitness}(S) = w_1 \cdot \text{NL}(S) / 120 + w_2 \cdot \text{SAC}(S) + w_3 \cdot \text{BIC}(S) + w_4 \cdot (1 - \text{DU}(S) / 256), \quad (3)$$

where NL(S) is the minimum nonlinearity across all component Boolean functions of the S-box, computed via the Walsh-Hadamard transform; SAC(S) is the strict avalanche criterion measuring the deviation from the ideal value of 0.5; BIC(S) is the bit independence criterion combining BIC-nonlinearity and BIC-SAC; and DU(S) is the differential uniformity measuring resistance to differential cryptanalysis. The weights are set as w_1 = 0.4, w_2 = 0.2, w_3 = 0.2, w_4 = 0.2 to emphasize nonlinearity. Permutation integrity is maintained through order-based encoding and Partially Mapped Crossover (PMX) for GA, while swap-based perturbation operators are employed for swarm-based algorithms.

3.1.3 | Problem 3: Malware ensemble classification optimization

The malware classification problem involves optimizing both the selection of base learners for an ensemble and their voting weights. Let C = {c_1, c_2, …, cm} denote a pool of m candidate base classifiers. A solution vector z = (s_1, …, sm, w_1, …, wm) encodes binary selection indicators si ∈ {0, 1} and continuous weights wi ∈ [0, 1] for each classifier. The ensemble prediction for a sample is determined by weighted majority voting:

$$\hat{y} = \text{argmax}_k \Sigma i: si = 1 \, wi \cdot \mathbb{I}(ci(\mathbf{x}) = k). \quad (4)$$

The fitness function maximizes the weighted F1-score under a complexity constraint:

$$\text{Fitness}(\mathbf{z}) = \text{F1weighted}(\mathbf{z}) - \lambda \cdot \Sigma si \cdot ti / Tmax, \quad (5)$$

where ti is the inference time of classifier ci, Tmax is the maximum allowable ensemble inference time, and λ = 0.05 is a complexity penalty coefficient. The candidate classifier pool consists of: RF, Support Vector Machine (SVM), k-Nearest Neighbors (kNN), XGBoost (XGB), and LightGBM (LGBM).

3.2 | Algorithms Implemented

Five metaheuristic algorithms were implemented with standardized population sizes and iteration counts to ensure fair comparison. All algorithms were implemented in Python 3.10 using NumPy for vectorized operations. The key parameter settings are detailed below and summarized in *Table 2*.

GA: the GA implementation follows Holland's [25] canonical framework with real-valued encoding. Simulated Binary Crossover (SBX) with distribution index ηc = 20 is employed for recombination, and

polynomial mutation with distribution index $\eta m = 20$ governs perturbation. An elitism rate of 5% preserves the best individuals across generations, and tournament selection with tournament size 3 determines parent selection.

PSO: the PSO variant follows the standard formulation of Kennedy and Eberhart [26] with linearly decreasing inertia weight from wmax = 0.9 to wmin = 0.4. Cognitive and social acceleration coefficients are set to $c_1 = c_2 = 2.0$. Velocity clamping is applied to prevent divergence.

GWO: the GWO implementation follows Mirjalili et al. [7], with the convergence parameter a linearly decreasing from 2 to 0 over the course of iterations, governing the transition from exploration to exploitation. The positions of alpha (α), beta (β), and delta (δ) wolves guide the search.

WOA: following Mirjalili and Lewis [27], the WOA employs a spiral coefficient b = 1 and a switching probability p = 0.5 to alternate between shrinking encirclement and spiral update mechanisms.

HHO: the HHO algorithm, proposed by Heidari et al. [6], models cooperative hunting behavior of Harris hawks. The escaping energy E_0 is drawn uniformly from [−1, 1], and the energy $E = 2E_0(1 - t/T)$ governs the transition between exploration ($|E| \geq 1$) and exploitation ($|E| < 1$) phases, with four exploitation strategies based on the prey's escape behavior.

Table 2. Algorithm parameter settings.

Parameter	GA	PSO	GWO	WOA	HHO
Population/swarm size	100	100	100	100	100
Maximum iterations	300	300	300	300	300
Crossover type	SBX (η_c=20)	—	—	—	—
Mutation type	Polynomial (η_m=20)	—	—	—	—
Elitism rate	5%	—	—	—	—
Inertia weight (w)	—	0.9 → 0.4	—	—	—
c_1, c_2	—	2.0, 2.0	—	—	—
Convergence parameter (a)	—	—	2 → 0	—	—
Spiral coefficient (b)	—	—	—	1	—
Switching probability (p)	—	—	—	0.5	—
Escaping energy (E_0)	—	—	—	—	U[−1, 1]
Selection type	Tournament (k=3)	—	—	—	—
Transfer function (binary)	Sigmoid	Sigmoid	Sigmoid	Sigmoid	Sigmoid

3.3 | Datasets

Four benchmark datasets spanning two cybersecurity domains were employed in this study. *Table 3* presents their characteristics.

Table 3. Dataset characteristics.

Dataset	Records	Features	Classes	Class Distribution	Year	Source
NSL-KDD	125,973 (train)/22,544 (test)	41	5	Normal: 53.5%, DoS: 36.5%, Probe: 9.2%, R2L: 0.6%, U2R: 0.04%	2009	Tavallaee et al. [8]
UNSW-NB15	175,341 (train)/82,332 (test)	49	10	Normal: 37.0%, Generic: 18.9%, Exploits: 16.1%, Fuzzers: 10.9%, Other: 17.1%	2015	Moustafa and Slay [9]
CICIDS-2017	2,830,743	78	15	Benign: 80.3%, DDoS: 5.1%, PortScan: 5.6%, Other attacks: 9.0%	2017	Sharafaldin et al. [28]
Malimg	9,339	—	25	Largest family: 2,949 (Allaple.A), Smallest: 80 (Skintrim.N)	2011	Nataraj et al. [22]

NSL-KDD is a refined version of the original KDD Cup 99 dataset with duplicate records removed, containing 41 features across numeric, categorical, and binary types. The five classes comprise Normal traffic and four attack categories: Denial of Service (DoS), Probe, Remote-to-Local (R2L), and User-to-Root (U2R). The dataset exhibits significant class imbalance, with U2R attacks constituting only 0.04% of the training data.

UNSW-NB15, developed by Moustafa and Slay [9] at the university of New South Wales, contains 49 features representing modern network traffic with 9 attack categories including Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode, and Worms. This dataset is considered more representative of contemporary network traffic than NSL-KDD.

CICIDS-2017, generated by the Canadian Institute for Cybersecurity, comprises over 2.8 million flow records with 78 bidirectional flow features. It includes benign traffic and 14 attack types representing modern threat scenarios including DDoS, heartbleed, brute force, web attacks, infiltration, and botnet activity.

Malimg consists of 9,339 malware binary images across 25 families. Following Nataraj et al. [22], each malware binary is converted to a grayscale image, resized to a fixed dimension, and features are extracted using GIST descriptors and Histogram of Oriented Gradients (HOG), yielding a 320-dimensional feature vector per sample.

For all IDS datasets, categorical features were one-hot encoded, and all numeric features were standardized to zero mean and unit variance using the training set statistics. Missing values (present in CICIDS-2017) were imputed using median values. Infinite values resulting from division-by-zero in flow feature extraction were replaced with the column maximum.

3.4 | Experimental Setup

All experiments were conducted under the following protocols to ensure statistical rigor and reproducibility:

Cross-validation: 10-fold stratified cross-validation was employed for fitness evaluation during the optimization process. Final reported metrics are computed on held-out test sets not used during optimization.

Independent runs: each algorithm–dataset configuration was executed 30 independent times with different random seeds to account for stochastic variability. Results are reported as mean ± standard deviation.

Iterations: all algorithms were run for 300 iterations per optimization run, with fitness values recorded at each iteration for convergence analysis.

Classifier: RF with 200 estimators was used as the base classifier for IDS experiments. The ensemble classifier pool for malware classification comprised RF (200 trees), SVM (RBF kernel), kNN (k=5), XGBoost (100 estimators), and LightGBM (100 estimators).

Hardware: all experiments were executed on a workstation equipped with dual Intel Xeon E5-2680 v4 processors (28 cores total), 128 GB DDR4 RAM, and an NVIDIA Tesla V100 GPU (32 GB HBM2). GPU acceleration was used only for XGBoost and LightGBM training.

Software: Python 3.10.12, scikit-learn 1.3.0, XGBoost 1.7.6, LightGBM 4.0.0, TensorFlow 2.12, NumPy 1.24.3, SciPy 1.11.1.

Statistical tests: the Friedman test was used for omnibus comparison across algorithms, followed by Nemenyi post-hoc tests for pairwise comparisons. Pairwise Wilcoxon signed-rank tests with Bonferroni correction were additionally conducted. Significance level was set at $\alpha_{stat} = 0.05$.

4 | Results and Analysis

4.1 | Intrusion Detection Results

4.1.1 | NSL-KDD dataset

Table 4 presents the classification results on the NSL-KDD dataset. All five metaheuristic-optimized configurations outperformed the full-feature baseline, confirming the efficacy of feature selection for IDS. The GWO-RF configuration achieved the highest accuracy of 99.41% ± 0.12% while selecting only 9 of 41 features (78.0% reduction), followed by HHO-RF (99.28% ± 0.15%, 10 features) and PSO-RF (99.12% ±

0.18%, 11 features). The full-feature RF baseline achieved 97.84% accuracy using all 41 features, representing a 1.57 percentage point deficit compared to GWO-RF.

Table 4. Classification results on NSL-KDD dataset (mean ± std over 30 runs).

Algorithm–Classifier	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	Features	Reduction (%)	Time (s)
GA-RF	98.73 ± 0.21	98.45 ± 0.28	98.62 ± 0.24	98.53 ± 0.25	14 ± 2.1	65.9	8.7
PSO-RF	99.12 ± 0.18	98.89 ± 0.22	99.01 ± 0.19	98.95 ± 0.20	11 ± 1.8	73.2	7.2
GWO-RF	99.41 ± 0.12	99.23 ± 0.14	99.38 ± 0.13	99.30 ± 0.13	9 ± 1.3	78.0	6.1
WOA-RF	99.05 ± 0.20	98.71 ± 0.26	98.92 ± 0.22	98.81 ± 0.23	12 ± 1.9	70.7	7.8
HHO-RF	99.28 ± 0.15	99.05 ± 0.18	99.17 ± 0.16	99.11 ± 0.17	10 ± 1.5	75.6	6.8
Full Features-RF	97.84 ± 0.31	97.12 ± 0.38	97.56 ± 0.33	97.34 ± 0.35	41	0.0	14.3

The feature reduction achieved by GWO is particularly noteworthy. Analysis of the most frequently selected features across 30 runs revealed that GWO consistently identified a compact, informative subset dominated by temporal and connection-based features: src_bytes (selected in 100% of runs), dst_bytes (100%), service (96.7%), flag (93.3%), logged_in (90.0%), count (86.7%), serror_rate (83.3%), dst_host_srv_count (80.0%), and dst_host_same_src_port_rate (76.7%). Content-based features such as num_shells, num_outbound_cmds, and is_host_login were rarely selected (≤ 6.7%), confirming their low discriminative value for network traffic classification.

4.1.2 | UNSW-NB15 dataset

Table 5. Classification results on UNSW-NB15 dataset (mean ± std over 30 runs).

Algorithm–Classifier	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	Features	Reduction (%)	Time (s)
GA-RF	95.34 ± 0.38	94.87 ± 0.45	95.12 ± 0.40	94.99 ± 0.42	18 ± 2.8	63.3	15.4
PSO-RF	96.21 ± 0.29	95.78 ± 0.34	96.05 ± 0.31	95.91 ± 0.32	15 ± 2.3	69.4	12.8
GWO-RF	97.18 ± 0.22	96.85 ± 0.27	97.03 ± 0.24	96.94 ± 0.25	12 ± 1.7	75.5	10.3
WOA-RF	95.89 ± 0.33	95.41 ± 0.39	95.72 ± 0.35	95.56 ± 0.37	16 ± 2.5	67.3	13.5
HHO-RF	96.82 ± 0.25	96.45 ± 0.30	96.68 ± 0.27	96.56 ± 0.28	13 ± 1.9	73.5	11.1
Full Features-RF	93.67 ± 0.45	92.89 ± 0.52	93.34 ± 0.48	93.11 ± 0.49	49	0.0	22.6

The UNSW-NB15 dataset, which is considered more challenging due to its modern traffic patterns and diverse attack categories, exhibited generally lower accuracy values compared to NSL-KDD. Nevertheless, the relative performance ranking among algorithms remained consistent, with GWO-RF leading at 97.18% accuracy and 75.5% feature reduction. The performance gap between metaheuristic-optimized and full-feature baselines was more pronounced on this dataset (3.51 percentage points for GWO), underscoring the importance of feature selection for complex, high-dimensional intrusion detection tasks.

4.1.3 | CICIDS-2017 dataset

Table 6. Classification results on CICIDS-2017 dataset (mean ± std over 30 runs).

Algorithm–Classifier	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	Features	Reduction (%)	Time (s)
GA-RF	98.45 ± 0.19	98.12 ± 0.24	98.31 ± 0.21	98.21 ± 0.22	22 ± 3.4	71.8	42.7
PSO-RF	99.01 ± 0.14	98.78 ± 0.18	98.89 ± 0.15	98.83 ± 0.16	18 ± 2.7	76.9	35.1
GWO-RF	99.34 ± 0.10	99.15 ± 0.13	99.27 ± 0.11	99.21 ± 0.12	14 ± 2.0	82.1	28.3
WOA-RF	98.78 ± 0.17	98.49 ± 0.22	98.65 ± 0.19	98.57 ± 0.20	19 ± 2.9	75.6	37.4
HHO-RF	99.19 ± 0.12	98.97 ± 0.15	99.10 ± 0.13	99.03 ± 0.14	16 ± 2.3	79.5	31.2
Full Features-RF	97.23 ± 0.28	96.78 ± 0.34	97.01 ± 0.30	96.89 ± 0.31	78	0.0	68.9

On the large-scale CICIDS-2017 dataset (2.8 million records, 78 features), GWO-RF achieved the most aggressive feature reduction (82.1%, selecting only 14 of 78 features) while maintaining the highest accuracy

(99.34%). The computational efficiency gains were substantial: GWO-RF required 28.3 seconds for training compared to 68.9 seconds for the full-feature baseline, representing a 58.9% reduction in training time. This is particularly relevant for real-time IDS deployment scenarios where model retraining frequency is an operational consideration.

4.1.4 | Per-class detection rates

Table 7 presents per-class detection rates (recall) on the NSL-KDD dataset, revealing the impact of metaheuristic-guided feature selection on minority class detection.

Table 7. Per-class detection rate (%) on NSL-KDD dataset.

Attack Type	GA	PSO	GWO	WOA	HHO
Normal	99.1 ± 0.2	99.5 ± 0.1	99.7 ± 0.1	99.3 ± 0.2	99.6 ± 0.1
DoS	99.4 ± 0.1	99.7 ± 0.1	99.9 ± 0.0	99.6 ± 0.1	99.8 ± 0.1
Probe	98.2 ± 0.4	98.8 ± 0.3	99.3 ± 0.2	98.5 ± 0.3	99.1 ± 0.2
R2L	91.3 ± 1.2	93.7 ± 1.0	95.8 ± 0.8	92.9 ± 1.1	94.6 ± 0.9
U2R	72.5 ± 3.8	78.3 ± 3.2	84.1 ± 2.7	76.8 ± 3.5	81.2 ± 3.0

All algorithms achieved excellent detection rates (>99%) for majority classes (Normal, DoS). However, performance diverged substantially for the severely underrepresented R2L and U2R attack classes. GWO exhibited the highest U2R detection rate of 84.1%, representing an 11.6 percentage point improvement over GA (72.5%). This advantage is attributable to GWO's hierarchical leadership structure, which maintains a diverse feature subset that captures the subtle behavioral patterns characteristic of privilege escalation attacks. The U2R class, comprising only 52 training samples (0.04% of training data), remains the most challenging category for all algorithms, consistent with findings in the broader IDS literature.

4.1.5 | Convergence analysis

Convergence curves across the five algorithms on the NSL-KDD dataset revealed distinctive optimization dynamics. GWO achieved rapid initial convergence, reaching 95% of its final fitness value by iteration 80 and stabilizing near its optimum by iteration 120. This rapid convergence is attributed to the alpha-beta-delta leadership hierarchy, which provides strong directional guidance in the early optimization phase. HHO exhibited a characteristic two-phase convergence pattern: moderate progress during the exploration phase (iterations 1–100) followed by accelerated improvement during the exploitation phase (iterations 100–200) as the escaping energy parameter transitioned below the threshold value. PSO demonstrated steady, linear-like convergence throughout the iteration range, reaching its final fitness value around iteration 180. WOA showed oscillatory convergence behavior consistent with its spiral update mechanism, periodically improving in bursts rather than monotonically. GA exhibited the slowest convergence, requiring approximately 220 iterations to reach 95% of its final fitness, though it maintained the greatest diversity in the population throughout the run.

4.2 | Cryptographic Substitution-Box (S-box) Optimization Results

Table 8 presents the S-box quality metrics achieved by each algorithm, compared against the AES Rijndael S-box as a gold standard reference.

Table 8. S-box quality metrics comparison (mean ± std over 30 runs).

Algorithm	NL (avg)	SAC (avg)	BIC-NL	BIC-SAC	DU	LP_{max}	Iterations	Time (s)
GA	106.5 ± 1.8	0.4987 ± 0.0021	103.2 ± 2.1	0.5012 ± 0.0018	12 ± 2	0.1406	300	245 ± 31
PSO	108.3 ± 1.5	0.5001 ± 0.0018	105.1 ± 1.8	0.4998 ± 0.0015	10 ± 2	0.1250	300	198 ± 24
GWO	110.1 ± 1.1	0.5023 ± 0.0014	106.8 ± 1.4	0.4989 ± 0.0012	8 ± 1	0.0938	300	167 ± 18
WOA	109.2 ± 1.3	0.5009 ± 0.0016	105.7 ± 1.6	0.4995 ± 0.0014	10 ± 2	0.1094	300	183 ± 21
HHO	112.0 ± 0.7	0.5034 ± 0.0011	108.4 ± 1.0	0.4982 ± 0.0010	6 ± 1	0.0625	300	152 ± 15
AES Rijndael	112	0.5000	112	0.5000	4	0.0625	—	—

HHO demonstrated clear superiority in S-box optimization, achieving the highest average nonlinearity of 112.0 ± 0.7, matching the AES Rijndael S-box's nonlinearity score. The HHO-generated S-boxes also achieved the lowest differential uniformity (6 ± 1) among the metaheuristic approaches, only marginally higher than the AES S-box's optimal value of 4. The maximum linear probability (LPmax) of 0.0625 matches the AES S-box, indicating comparable resistance to linear cryptanalysis.

The SAC values for all algorithms were close to the ideal value of 0.5000, with HHO achieving 0.5034 ± 0.0011. While the AES S-box achieves precisely 0.5000, the slightly elevated SAC of HHO-generated S-boxes falls within an acceptable cryptographic margin. The BIC-NL scores show the greatest gap relative to AES: the highest BIC-NL achieved was 108.4 (HHO), compared to the AES S-box's 112. This suggests that while metaheuristic-generated S-boxes approach AES quality in individual component function nonlinearity, achieving pairwise independence between component functions remains more challenging for optimization-based approaches.

The computational time for HHO (152 ± 15 seconds for 300 iterations) was the lowest among all algorithms, which is noteworthy given its superior output quality. This efficiency arises from HHO's adaptive phase switching mechanism, which concentrates computational effort on exploitation once a promising region of the permutation space has been identified. GA required the longest computation time (245 ± 31 seconds), largely due to the computational overhead of the PMX crossover operator on 256-element permutations.

A detailed analysis of the best HHO-generated S-box (NL = 112, DU = 6) revealed that its algebraic degree was 7 (matching the AES S-box) and that all eight component Boolean functions were balanced (equal number of 0s and 1s in the truth table). The S-box satisfied the strict avalanche criterion with a mean absolute deviation of 0.0018 from the ideal value of 0.5, and the output bit independence was confirmed via BIC analysis. These results demonstrate that HHO is capable of producing cryptographically viable S-boxes suitable for deployment in lightweight block cipher designs where the algebraic structure of the AES S-box (based on multiplicative inverse in GF(2^8)) may be undesirable due to susceptibility to algebraic attacks.

4.3 | Malware Classification Results

Table 9 presents the malware classification performance on the Malimg dataset using metaheuristic-optimized ensemble classifiers.

Table 9. Malware classification on the Malimg dataset (mean ± std over 30 runs).

Algorithm–Ensemble	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	Base Learners Selected	Inference (ms)
GA-ensemble	97.23 ± 0.34	97.01 ± 0.39	97.15 ± 0.36	97.08 ± 0.37	RF + SVM + kNN + XGB	12.3
PSO-ensemble	98.87 ± 0.18	98.72 ± 0.22	98.81 ± 0.20	98.76 ± 0.21	RF + XGB + LightGBM	8.7
GWO-ensemble	98.45 ± 0.23	98.31 ± 0.27	98.39 ± 0.25	98.35 ± 0.26	RF + SVM + XGB	10.1
WOA-ensemble	97.89 ± 0.29	97.65 ± 0.34	97.78 ± 0.31	97.71 ± 0.32	RF + kNN + XGB + LightGBM	11.5
HHO-ensemble	98.56 ± 0.21	98.42 ± 0.25	98.50 ± 0.23	98.46 ± 0.24	RF + XGB + LightGBM	9.2

PSO-Ensemble achieved the highest accuracy (98.87%) and F1-score (98.76%) while simultaneously selecting the most computationally efficient ensemble configuration (RF + XGB + LightGBM) with an inference time of only 8.7 ms per sample. The PSO-optimized weight distribution assigned 42.3% weight to XGBoost, 35.1% to RF, and 22.6% to LightGBM, reflecting XGBoost's strong performance on the gradient-based features extracted from malware images.

Notably, PSO consistently excluded SVM and kNN from the optimal ensemble across the majority of runs (26/30 and 28/30, respectively), while GA selected all four classifiers including SVM and kNN in 73% of runs. This suggests that PSO's velocity-based exploration more effectively navigated the discrete-continuous search space of the ensemble optimization problem, identifying that the marginal accuracy gain from including SVM and kNN was insufficient to justify their computational cost.

Analysis of per-family classification performance revealed that all algorithms achieved perfect (100%) detection rates for the five largest malware families (Allaple.A, Allaple.L, Yuner.A, VB.AT, Fakerean), which together comprise approximately 65% of the dataset. Performance differences emerged primarily for smaller families: PSO-Ensemble achieved a 96.25% F1-score on the challenging Skintrim.N family (80 samples), compared to 91.87% for GA-Ensemble and 94.50% for GWO-Ensemble. The Obfuscator.AD family, which contains heavily obfuscated binaries producing less distinctive visual textures, was the most challenging across all algorithms, with F1-scores ranging from 93.12% (GA) to 97.81% (PSO).

4.4 | Statistical Significance Analysis

To establish the statistical significance of observed performance differences, the Friedman non-parametric test was applied across all three security domains. *Table 10* presents the Friedman rankings.

Table 10. Friedman test rankings across three security domains.

Algorithm	IDS Rank	Crypto Rank	Malware Rank	Average Rank	Overall Position
GWO	1.33	2.00	2.33	1.89	1
HHO	2.00	1.00	2.00	1.67	2
PSO	2.67	2.67	1.00	2.11	3
WOA	3.33	3.33	3.67	3.44	4
GA	4.67	5.00	5.00	4.89	5

The Friedman test yielded a chi-squared statistic of $\chi^2 = 18.93$ with $p < 0.001$, indicating statistically significant differences among the five algorithms across the three cybersecurity domains. The average ranking revealed that while HHO achieved the lowest average rank (1.67) driven by its dominance in the cryptography domain, GWO exhibited the most balanced performance across all three domains (average rank 1.89). PSO ranked third overall (2.11) but achieved the best rank in the malware domain. GA consistently ranked last across all domains, suggesting that its crossover-mutation paradigm is less effective than swarm and predatory intelligence for cybersecurity optimization.

Nemenyi post-hoc analysis with a critical difference (CD) of 2.29 at $\alpha = 0.05$ revealed the following significant pairwise differences: GWO significantly outperformed GA (rank difference = 3.00 > CD), HHO significantly outperformed GA (rank difference = 3.22 > CD), and PSO significantly outperformed GA (rank difference = 2.78 > CD). The differences among GWO, HHO, and PSO were not statistically significant at $\alpha = 0.05$, indicating that these three algorithms form a statistically indistinguishable top tier.

Table 11 presents pairwise Wilcoxon signed-rank test p-values with Bonferroni correction, computed across the combined performance metrics of all three domains.

Table 11. Pairwise Wilcoxon signed-rank test p-values (Bonferroni corrected).

	GA	PSO	GWO	WOA	HHO
GA	—	0.0023*	0.0001*	0.0412*	0.0008*
PSO	0.0023*	—	0.0834	0.0187*	0.2145
GWO	0.0001*	0.0834	—	0.0043*	0.1267
WOA	0.0412*	0.0187*	0.0043*	—	0.0098*
HHO	0.0008*	0.2145	0.1267	0.0098*	—

* denotes statistical significance at $\alpha = 0.05$ after Bonferroni correction.

The Wilcoxon results corroborate the Friedman analysis. GA was significantly outperformed by all other algorithms ($p < 0.05$ in all pairwise comparisons). WOA was significantly outperformed by PSO, GWO, and HHO. The top three algorithms (GWO, HHO, PSO) showed no statistically significant pairwise differences, confirming their comparable overall performance despite domain-specific specializations.

4.5 | Computational Cost Analysis

Table 12 summarizes the average computation time per domain for each algorithm, aggregated over 30 independent runs.

Table 12. Average computation time per domain (minutes).

Algorithm	IDS (min)	Crypto (min)	Malware (min)	Total (min)	Per-Iteration (s)
GA	43.5 ± 4.2	4.08 ± 0.52	18.7 ± 2.1	66.28	4.42
PSO	36.0 ± 3.1	3.30 ± 0.40	15.3 ± 1.7	54.60	3.64
GWO	30.5 ± 2.5	2.78 ± 0.30	13.8 ± 1.4	47.08	3.14
WOA	39.0 ± 3.6	3.05 ± 0.35	16.1 ± 1.8	58.15	3.88
HHO	34.0 ± 2.8	2.53 ± 0.25	14.5 ± 1.5	51.03	3.40

GWO exhibited the lowest total computation time (47.08 minutes across all domains), while GA required the most computational resources (66.28 minutes). In the IDS domain, which dominated computation time due to the size of the CICIDS-2017 dataset (2.8 million records), GWO required 30.5 minutes compared to GA's 43.5 minutes — a 29.9% reduction. In the cryptography domain, HHO was the fastest (2.53 minutes), reflecting its efficient exploitation phase for the permutation-based S-box problem. The per-iteration cost ranged from 3.14 seconds (GWO) to 4.42 seconds (GA), with the differences primarily attributable to the complexity of the update operators rather than fitness evaluation overhead.

5 | Discussion

The experimental results reveal a clear pattern of domain-specific algorithm strengths, providing actionable guidance for cybersecurity practitioners. This section examines the underlying reasons for these specializations, discusses practical deployment considerations, and acknowledges limitations of the study.

GWO's superiority in intrusion detection: the GWO's hierarchical leadership structure — comprising alpha, beta, and delta wolves that collectively guide the pack — provides a balanced exploration-exploitation trade-off that is particularly well-suited to the feature selection problem. In high-dimensional feature spaces (41–78 dimensions for IDS), the three-leader mechanism prevents premature convergence to local optima while maintaining sufficient exploitation pressure to identify compact, discriminative feature subsets. The alpha wolf provides strong directional guidance toward promising feature combinations, while the beta and delta wolves explore adjacent regions, effectively maintaining a diverse search front. This mechanism mirrors the nature of the IDS feature selection problem, where multiple near-optimal feature subsets may exist with different compositions but comparable classification performance.

HHO's superiority in cryptography: the HHO algorithm's adaptive energy-based phase switching proved particularly effective for the S-box optimization problem, which requires precise navigation of the 256! permutation space. The four exploitation strategies — soft besiege, hard besiege, soft besiege with progressive rapid dives, and hard besiege with progressive rapid dives — provide nuanced local search capabilities essential for fine-tuning the nonlinearity and differential uniformity of candidate S-boxes. Unlike the IDS problem where coarse feature inclusion/exclusion decisions dominate, S-box optimization requires subtle permutation adjustments where swapping even a single pair of elements can significantly impact cryptographic quality metrics. HHO's Lévy flight-based progressive rapid dives enable precisely this type of fine-grained perturbation.

PSO's superiority in malware ensemble optimization: the PSO algorithm excelled in the ensemble classifier optimization task due to its natural handling of the hybrid discrete-continuous search space. The velocity-

based update mechanism allows PSO particles to smoothly navigate the continuous weight space while the sigmoid transfer function effectively handles the discrete base learner selection variables. The memory mechanisms (personal best and global best) enable PSO to efficiently recall and recombine previously successful ensemble configurations, a property that is particularly advantageous when the fitness landscape has a relatively small number of high-performing ensemble architectures.

Practical deployment considerations: for real-time IDS processing network traffic at line speed, the combination of GWO-based feature selection and RF classification offers a compelling operational profile. The 78% feature reduction directly translates to reduced inference latency, as only 9 features need to be extracted and processed per network flow record. In production environments, the feature selection optimization can be performed offline on historical traffic data, with the resulting feature subset deployed to lightweight inline classifiers. For cryptographic applications, the HHO-generated S-boxes can serve as drop-in replacements for algebraically constructed S-boxes in lightweight block ciphers targeting IoT devices, where diversity of S-box designs may be desirable to mitigate algebraic attack vectors. The PSO-optimized ensemble for malware classification is well-suited for integration into commercial Endpoint Detection and Response (EDR) platforms, where the 8.7 ms inference time per sample enables real-time scanning of executable files.

Comparison with deep learning approaches: while deep learning methods — particularly Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) — have achieved competitive accuracy in intrusion detection and malware classification, they impose significantly higher computational costs for both training and inference. The GWO-RF approach proposed herein achieves 99.41% accuracy on NSL-KDD, which is comparable to or exceeds reported CNN-based IDS results (typically 97–99%), while requiring orders of magnitude less computational resources and offering full model interpretability. The feature subsets identified by metaheuristic optimization are human-readable and can inform security analysts about which traffic characteristics are most indicative of malicious activity, a property absent in deep learning black-box models.

Limitations: several limitations of this study warrant acknowledgment. First, the feature selection is binary (include/exclude), and continuous feature weighting may yield further improvements. Second, the S-box optimization was limited to 8×8 bijective permutations; larger S-boxes (e.g., 16×16) present a combinatorially harder problem that may require algorithm modifications. Third, the Malimg dataset, while widely used, contains only 25 malware families and may not fully represent the diversity of modern malware. Fourth, the experiments were conducted in an offline setting with static datasets; the impact of concept drift in streaming network data was not evaluated. Fifth, adversarial robustness — the susceptibility of metaheuristic-optimized classifiers to adversarial examples crafted to evade detection — remains an open concern not addressed in this work.

6 | Conclusion

This paper presented a comprehensive multi-domain experimental study evaluating five metaheuristic optimization algorithms — GA, PSO, GWO, WOA, and HHO — across three critical cybersecurity applications: network intrusion detection, cryptographic S-box generation, and malware classification. Through rigorous experimentation on four benchmark datasets with 30 independent runs per configuration and full statistical analysis, the following principal conclusions were established.

First, metaheuristic-optimized feature selection consistently and significantly outperformed full-feature baselines across all three IDS datasets, with the best configuration (GWO-RF) achieving 99.41% accuracy on NSL-KDD with a 78.0% feature reduction. This result underscores the importance of intelligent feature selection as a prerequisite for high-performance intrusion detection.

Second, HHO demonstrated exceptional capability for cryptographic S-box generation, producing S-boxes with a nonlinearity of 112 and differential uniformity of 6 — closely approaching the AES Rijndael S-box quality while offering structural diversity not achievable through algebraic construction methods.

Third, PSO-optimized ensemble classifiers achieved the highest malware classification performance (98.76% F1-score) while simultaneously minimizing ensemble complexity and inference time, demonstrating the practical viability of metaheuristic-driven ensemble design for endpoint security.

Fourth, no single algorithm dominated across all three domains, highlighting the importance of domain-specific algorithm selection. The Friedman test confirmed statistically significant performance differences ($p < 0.001$), with GWO, HHO, and PSO forming a top tier that significantly outperformed WOA and GA.

For cybersecurity practitioners, the following recommendations are offered: (i) deploy GWO for feature selection in network IDS, particularly for high-dimensional datasets; (ii) employ HHO for generating diverse, high-quality S-boxes for lightweight cipher designs; and (iii) utilize PSO for optimizing ensemble malware classifiers in endpoint security platforms.

Future research directions include: (a) dynamic feature selection for streaming network data using online metaheuristic variants to address concept drift; (b) optimization of post-quantum cryptographic primitives, including lattice-based and code-based parameters; (c) evaluation of adversarial robustness of metaheuristic-optimized classifiers against evasion attacks; (d) extension to multi-objective optimization frameworks that simultaneously optimize detection accuracy, computational cost, and energy consumption for green cybersecurity; and (e) investigation of hybrid metaheuristics that combine the strengths of GWO's exploration with HHO's exploitation for unified cross-domain security optimization.

References

[1] Morgan, S. (2020). Cybercrime to cost the world $10.5 trillion annually by 2025. *Cybercrime magazine*, *13*(11), 2020. https://cybersecurityventures.com/hackerpocalypse-cybercrime-report-2016/

[2] Roesch, M. (1999). Snort: Lightweight intrusion detection for networks. *Lisa* (Vol. 99, No. 1, pp. 229-238). USENIX Association. https://www.usenix.org/legacy/event/lisa99/full_papers/roesch/roesch.pdf

[3] Ambusaidi, M. A., He, X., Nanda, P., & Tan, Z. (2016). Building an intrusion detection system using a filter-based feature selection algorithm. *IEEE transactions on computers*, *65*(10), 2986–2998. https://doi.org/10.1109/TC.2016.2519914

[4] Farah, M. A. Ben, Farah, A., & Farah, T. (2020). An image encryption scheme based on a new hybrid chaotic map and optimized substitution box. *Nonlinear dynamics*, *99*(4), 3041–3064. https://doi.org/10.1007/s11071-019-05413-8

[5] Ucci, D., Aniello, L., & Baldoni, R. (2019). Survey of machine learning techniques for malware analysis. *Computers & security*, *81*, 123–147. https://doi.org/10.1016/j.cose.2018.11.001

[6] Heidari, A. A., Mirjalili, S., Faris, H., Aljarah, I., Mafarja, M., & Chen, H. (2019). Harris Hawks optimization: Algorithm and applications. *Future generation computer systems*, *97*, 849–872. https://doi.org/10.1016/j.future.2019.02.028

[7] Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Grey wolf optimizer. *Advances in engineering software*, *69*, 46–61. https://doi.org/10.1016/j.advengsoft.2013.12.007

[8] Tavallaee, M., Bagheri, E., Lu, W., & Ghorbani, A. A. (2009). A detailed analysis of the kdd cup 99 data set. *2009 IEEE symposium on computational intelligence for security and defense applications* (pp. 1–6). IEEE. https://doi.org/10.1109/CISDA.2009.5356528

[9] Moustafa, N., & Slay, J. (2015). UNSW-nb15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). *2015 military communications and information systems conference (MILCIS)* (pp. 1–6). IEEE. https://doi.org/10.1109/MilCIS.2015.7348942

[10] Khammassi, C., & Krichen, S. (2017). A GA-LR wrapper approach for feature selection in network intrusion detection. *Computers & security*, *70*, 255–277. https://doi.org/10.1016/j.cose.2017.06.005

[11] Elmasry, W., Akbulut, A., & Zaim, A. H. (2020). Evolving deep learning architectures for network intrusion detection using a double PSO metaheuristic. *Computer networks*, *168*, 107042. https://doi.org/10.1016/j.comnet.2019.107042

[12] Varzaneh, Z. A., & Hosseini, S. (2024). An improved equilibrium optimization algorithm for feature selection problem in network intrusion detection. *Scientific reports*, *14*(1), 18696. https://www.nature.com/articles/s41598-024-67488-7

[13] Almomani, A., Alweshah, M., Al Khalayleh, S., Al-Refai, M., & Qashi, R. (2019). Metaheuristic algorithms-based feature selection approach for intrusion detection. In *Machine learning for computer and cyber security* (pp. 184–208). CRC Press. https://doi.org/10.1201/9780429504044-8

[14] Kasongo, S. M., & Sun, Y. (2020). A deep learning method with wrapper based feature extraction for wireless intrusion detection system. *Computers & security*, *92*, 101752. https://doi.org/10.1016/j.cose.2020.101752

[15] Koryshev, N., Hodashinsky, I., & Shelupanov, A. (2021). Building a fuzzy classifier based on whale optimization algorithm to detect network intrusions. *Symmetry*, *13*(7), 1211. https://doi.org/10.3390/sym13071211

[16] Thaher, T., Heidari, A. A., Mafarja, M., Dong, J. S., & Mirjalili, S. (2019). Binary Harris Hawks optimizer for high-dimensional, low sample size feature selection. In *Evolutionary machine learning techniques: Algorithms and applications* (pp. 251–272). Springer. https://doi.org/10.1007/978-981-32-9990-0_12

[17] Daemen, J., & Rijmen, V. (2002). *The design of Rijndael*. Springer. https://doi.org/10.1007/978-3-662-60769-5

[18] Alzaidi, A. A., Ahmad, M., Doja, M. N., Al Solami, E., & Beg, M. M. S. (2018). A new 1D chaotic map and $\beta $-hill climbing for generating substitution-boxes. *IEEE access*, *6*, 55405–55418. https://doi.org/10.1109/ACCESS.2018.2871557

[19] Kuznetsov, O., Poluyanenko, N., Frontoni, E., Arnesano, M., & Smirnov, O. (2024). *Evolutionary approach to s-box generation: Optimizing nonlinear substitutions in symmetric ciphers*. https://arxiv.org/abs/2407.03510

[20] Ali, R. S., Hasoun, R. K., Tayyeh, H. K., & Mohammed, M. Q. (2024). A comprehensive review on s-box generation methods. *AIP conference proceedings* (Vol. 3207, p. 20001). AIP Publishing LLC. https://doi.org/10.1063/5.0234892

[21] Ahmad, M., Khaja, I. A., Baz, A., Alhakami, H., & Alhakami, W. (2020). Particle swarm optimization based highly nonlinear substitution-boxes generation for security applications. *IEEE access*, *8*, 116132–116147. https://doi.org/10.1109/ACCESS.2020.3004449

[22] Nataraj, L., Karthikeyan, S., Jacob, G., & Manjunath, B. S. (2011). Malware images: Visualization and automatic classification. *Proceedings of the 8th international symposium on visualization for cyber security* (pp. 1-7). Association for Computing Machinery (ACM). https://doi.org/10.1145/2016904.2016908

[23] Panda, B., Bisoyi, S. S., & Panigrahy, S. (2023). An ensemble approach for imbalanced multiclass malware classification using 1D-CNN. *PeerJ computer science*, *9*, e1677. https://doi.org/10.7717/peerj-cs.1677

[24] Moujoud, L., Ayache, M., & Belmekki, A. (2024). Ensemble learning for malware detection. *International conference on smart applications and data analysis* (pp. 233–245). Cham: Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-77040-1_17

[25] Holland, J. H. (1992). *Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence*. MIT Press. https://mitpress.mit.edu/9780262581110/adaptation-in-natural-and-artificial-systems/

[26] Eberhart, R., & Kennedy, J. (1995). Particle swarm optimization. *Proceedings of the IEEE international conference on neural networks* (Vol. 4, pp. 1942–1948). IEEE. https://doi.org/10.1109/ICNN.1995.488968

[27] 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

[28] Sharafaldin, I., Lashkari, A. H., & Ghorbani, A. A. (2018). Toward generating a new intrusion detection dataset and intrusion traffic characterization. *ICISSp*, *1*(2018), 108–116. https://doi.org/10.5220/0006639801080116