Anomaly-based Intrusion Detection in IoT using Enhanced Kepler Optimization Algorithm for Feature Selection

Lulu Zhang
Department of Information Engineering, Hebei Chemical & Pharmaceutical College, Shijiazhuang 050026, China
E-mail: zh_lulu1114@163.com

Keywords: intrusion detection, internet of things, botnet, feature selection, optimization

Received: March 25, 2025

The proliferation of Internet of Things (IoT) devices has increased the risk of botnet attacks due to the inherent vulnerabilities of IoT networks. To mitigate this threat, this study presents an anomaly-based intrusion detection framework that incorporates the Enhanced Kepler Optimization Algorithm (EKOA) for feature selection. EKOA integrates adaptive processes, such as dynamic adaptation, oscillatory chaotic force, crosswise solution formation, and optimization based on elites, in an effort to balance exploitation and exploration in favor of enhancing convergence speed alongside solution diversity. The selected features are evaluated using K-Nearest Neighbor (KNN) and Decision Tree (DT) classifiers. Experiments were conducted on typical IoT datasets, i.e., Mirai and Gafgyt. Accuracy, AUC, G-mean, and precision were also used for performance evaluation. The new system achieved detection accuracy greater than 99% and reduced the list of features by 35%. The new system exhibits good generalization capability, botnet attack resistance, and applicability in high-dimensional applications. The results show a good future for practical application in real-time intrusion detection on IoTs.

Povzetek: EKOA (Enhanced Kepler Optimization Algorithm) za izbiro značilnic izboljša detekcijo vdorov (botnet) v IoT omrežjih. Dosega visoko odpornost proti napadom in deluje v realnem času.

1 Introduction

The Internet of Things (IoT) has revolutionized modern technology by connecting billions of devices across various domains, including healthcare, smart cities, and manufacturing [1]. This rapid growth in IoT has also led to serious vulnerabilities, particularly in botnet attacks [2]. Botnets are networks of compromised IoT devices under attacker control conducting large-scale malicious activities, including Distributed Denial of Service (DDoS), phishing, and stealing information [3]. As a general rule, low computational power, default configurations, and weak security protocols make IoT devices an easy target for attackers, posing a significant threat to network integrity and user privacy [4, 5].

Intrusion Detection Systems (IDSs) contribute to security issues in IoT networks by detecting hostile behavior and protecting against cyberattacks [6]. It contrasts with the traditional concept of security based on encryption techniques and authenticity, and this method analyzes network flow traffic and all flow patterns belonging to botnets for other types of cyberattacks [7]. Their nature being adaptive during evolution regarding attack pattern variations makes it essential regarding security in the case of IoTs [8]. In direct relation to this, the performance of the IDS framework heavily relies on selecting features that are both relevant and non-redundant. Utilized features enhance detection performance by distinguishing patterns that distinguish normal and malicious traffic, while removing noisy or irrelevant features reduces the computational cost and protects against overfitting. Efficient feature selection is thus crucial for obtaining a detection performance-resources utilization balance in IoT applications [9].

However, the field of IDS still faces numerous challenges. Traditional feature selection approaches often exhibit significant drawbacks when an IoT high-dimensional dataset contains many irrelevant and redundant features, resulting in increased computational overhead and reduced detection accuracy [10]. Typical traditional brute-force methods, which select the most static subset features beforehand without any update strategy, tend to suffer from insufficient adaptability due to inefficiency when applied to general datasets or several diverse attack scenarios [11]. While some meta-heuristics proposed for optimizing feature subsets present state-of-the-art performances, many face significant weaknesses, including imbalance in exploration and exploitation, which can converge at low speeds to the most optimized feature subsets [12].

The Enhanced Kepler Optimization Algorithm (EKOA) addresses these challenges by building on the foundations of the Kepler Optimization Algorithm (KOA). By incorporating Kepler's laws for the motion of planets, EKOA applies advanced techniques such as dynamic adaptation, oscillatory chaotic force, cross-sectional solution generation, and elite-guided optimization. Such enhancements fine-tune the algorithm's balancing between exploration and exploitation, keeping the population diversity intact, and accelerating convergence.

EKOA's architecture comprises adaptive exploration-exploitation control, chaotic force allocation to enhance population diversity improvement, and elite-based search for fine-tuning good solutions. These processes make EKOA capable of effectively handling IDS's high-dimensional feature space problems, such as redundancy, irrelevance, and the risk of premature convergence. As a result, EKOA offers a strong foundation for selecting compact yet effective feature subsets that enhance detection performance.

The primary objective of this study is to enhance intrusion detection performance in IoT networks by employing an efficient feature selection approach using an improved metaheuristic strategy. EKOA is employed to reduce dimensionality while preserving relevant indicators of malicious behavior. Based on this goal, the research is guided by the following questions:

- RQ1: Can a multi-objective binary variant of the EKOA effectively reduce irrelevant features in IoT intrusion datasets while maintaining high detection accuracy?
- RQ2: How does EKOA compare to other recent metaheuristic feature selection methods in terms of convergence speed, robustness to unseen attacks, and computational efficiency?
- RQ3: Is the proposed IDS framework with EKOA suitable for real-time deployment in IoT environments with constrained resources and high-volume traffic?

Thanks to the inclusion of several adaptive processes, the proposed technique is the first to use a multi-objective binary realization of the EKOA in an IoT-based intrusion detection system. The proposed system differs from previous methods. It includes a binary-encoded, chaos-driven optimization model coupled with an elite solution-based directional guidance for choosing features, aiming mainly at problems posed by high-dimensional IoT data and adaptive attack behaviors.

2 Related work

This section presents outstanding works on IoT botnet detection and feature selection, focusing on metaheuristic optimization methods. They were selected under their appropriateness for the problems addressed in the present paper: dealing with high-dimensional feature spaces, improving detection accuracy, and reducing computational overhead.

Haddadpajouh, et al. [13] proposed an integrated Support Vector Machine (SVM) for malware detection against IoT threats in cloud-edge gateways. The method utilized Gray Wolves Optimization (GWO) for the optimal selection of features based on Opcode and Bytecode datasets, which provided 99.72% accuracy at a lesser computational expense when compared to Deep Neural Networks (DNNs). Abu Khurma, et al. [14] proposed a hybrid method for the selection of features, which combines Ant Lion Optimization (ALO) and Salp Swarm Algorithm (SSA). Based on the N-BaIoT dataset, the method reported a 99.9% actual positive rate, besides resolving the issue of high-dimensional feature space.

Hosseini, et al. [15] suggested botnet detection with a hybrid Slime Mold Algorithm (SMA) and SSA for choosing features. The algorithm utilized chaos theory to balance exploration and exploitation, achieving higher detection in UCI datasets. Gharehchopogh, et al. [16] suggested a binary Multi-Objective Dynamic Harris Hawks Optimization (MODHHO) algorithm for choosing features. The algorithm utilized mutation operators and different classifiers (KNN, SVM, MLP, DT), which showed higher speed and accuracy on five datasets.

Alkhammash [17] offered a metaheuristic-based, blockchain-integrated model for DDoS attack detection (MHADMA-BCIDL). The model utilized Arctic Tern Optimization (ATO) for attribute selection and CNN-BiLSTM for classification, achieving 99.32% accuracy on the BoT-IoT dataset. Maghrabi, et al. [18] proposed a hybrid deep learning-based model (BESO-HDLBD) that incorporated Bald Eagle Search Optimization (BESO) for selecting attributes and a CNN-BiLSTM-Attention for bot identification. The model worked best on benchmarked datasets, outperforming existing methods in speed and accuracy.

Maazalahi and Hosseini [19] proposed a hybrid algorithm as a fusion between Whale Optimization Algorithm (WOA), Particle Swarm Optimization (PSO), and Sailfish Optimizer (SFO). The algorithm was tested on BoT-IoT and UNSW-NB15 datasets and achieved a detection accuracy of 99.8% in less execution time. Elsedimy and AboHashish [20] proposed FCM-SWA, an integration between fuzzy C-means clustering and Sperm Whale Algorithm (SWA), for IoT-driven innovative systems. The algorithm outperformed existing methods on BoT-IoT, NSL-KDD, and AWID datasets using adaptive threshold techniques in accuracy and precision.

Despite the good performance encompassed in existing feature selection and classification techniques, as indicated in Table 1, typical weaknesses still hold. Some models incur an inefficient exploration-exploitation balance, leading to convergence in the latter parts or locally optimal sets of features. Others attain good detection accuracy but are computationally costly, especially on high-dimensional or IoT streaming datasets. Most research works provide limited assessment on the impact of the selected feature on the robustness and generalizability of models to unknown attacks. The paper bridges these gaps by introducing EKOA for feature selection, aiming at achieving computational effectiveness, convergence speed, and high detection accuracy.

3 Materials

3.1 Multi-objective optimization

Multi-objective optimization aims to optimize multiple conflicting objectives, often resulting in trade-offs among them. Improving one objective can result in the deterioration of another, requiring solutions that balance these trade-offs.

Table 1: An overview of related works

Reference	Contribution	Accuracy	TPR	Shortcoming
[13]	Suggested a multi-kernel SVM using GWO for feature selection, achieving 99.72% accuracy with reduced training time.	99.72%	-	Limited evaluation datasets and focus on specific malware types (Cortex A9 samples).
[14]	Developed SSA–ALO hybrid for feature selection, achieving 99.9% TPR on N-BaIoT datasets with superior efficiency.	-	99.9%	High computational complexity for large-scale datasets.
[15]	Introduced SMA + SSA with chaos theory for balanced exploration and exploitation in feature selection.	-	-	Results lack comprehensive comparison with advanced optimization algorithms.
[16]	Presented MODHHO for multi-objective feature selection and versatile classification across multiple datasets.	98.1%	-	Moderate accuracy improvement compared to existing approaches.
[17]	Proposed MHADMA-BCIDL with blockchain integration and CNN-BiLSTM for DDoS detection, achieving 99.32% accuracy.	99.32%	-	Dependence on blockchain may introduce overhead in real-time systems.
[18]	Designed BESO-HDLBD with hybrid deep learning for spatial-temporal feature extraction and botnet detection.	99.4%	-	The computational cost is due to the BiLSTM and attention mechanisms in large datasets.
[19]	Proposed SFO-WOA-PSO-K-means hybrid with 99.8% accuracy and low execution time for botnet detection.	99.8%	-	Limited scalability for highly dynamic IoT environments.
[20]	Introduced FCM-SWA with enhanced clustering and global optimization for IoT-based innovative systems.	98.9%	-	Lacks evaluation on diverse IoT network scenarios and attack types.

The solutions to such problems are termed Pareto-optimal solutions, also known as the Pareto front, in which no objective can be enhanced without compromising at least one other objective. The mathematical formulation of a multi-objective optimization problem can be stated as follows:

$$min\, F = \{f_1(X), f_2(X), \ldots, f_M(X)\} \quad (1)$$

Subject to:

$$\begin{aligned} & g_i(X) \le 0, \;\; i = 1,2,\ldots,q \\ & h_i(X) \le 0, \;\; j = 1,2,\ldots,p \\ & X \epsilon \Omega \end{aligned} \quad (2)$$

Where $X = \{x_1, x_2, \ldots, x_D\}$ represents a decision vector in a D-dimensional space, $g_i(X)$ and $h_i(X)$ represent constraints of inequality and equality, respectively, and F is the set of M objective functions to optimize, Ω defines the feasible decision space.

In multi-objective optimization, a solution $U = \{u_1, u_2, \ldots, u_D\}$ is supposed to dominate another solution $V = \{v_1, v_2, \ldots, v_D\}$, denoted as $U \prec V$, if the following conditions are satisfied:

$$\begin{aligned} & f_i(U) \le f_i(V), \;\; \forall i \in \{1,2,\ldots,M\} \\ & f_i(U) < f_i(V), \;\; \exists i \in \{1,2,\ldots,M\} \end{aligned} \quad (3)$$

Non-dominated or Pareto-optimal solutions are the ones not dominated by another. They constitute the Pareto front of the problem, which is said to be the best set of conflicting objective trade-off solutions. The feasible solution is said to be satisfying all the constraints and is in the set of non-dominated solutions if and only if it qualifies for the criteria outlined above. The Pareto front, thus, shows all the Pareto-optimal solutions for a problem.

3.2 Feature selection

Feature selection is a crucial step in data classification, where the target is to select a subset of features from the total feature set *Fet*, consisting of *D* features and *N* samples, to maximize classification performance while minimizing computational cost [21]. The process can be formulated mathematically as follows:

A feature subset X is represented as a binary vector $X = (x_1, x_2, \ldots, x_D)$, where $x_j \in \{0,1\}$ specifies whether the j^{th} feature is selected $(x_j = 1)$ or not $(x_j = 0)$. The following equation can then describe the task of feature selection:

$$max\, H(X) \quad (4)$$

H(X) represents the objective function that evaluates the classification accuracy of the selected feature subset X.

The classifier's running time is directly proportional to the selected number of features. With a larger set of features, the classifier is computationally costlier and runs slower, but classification accuracy is potentially lower for a smaller set. The compromise between optimizing accuracy and keeping the selected number of features small is thus required. The compromise can be framed as a bi-objective optimization problem:

$$min\, F = (ERR(X), |X|) \quad (5)$$

Where *ERR(X)=1−H(X)* is the classification error for the selected feature set X and $|X|$ is the number of selected features.

3.3 Disruption operator

The disruption operator is taken from astrophysical phenomena, which tries to enhance population diversity for optimization methods. Including variation in the population expands the search area and manages exploration and exploitation effectively. The disruption operator successfully enhances the performance of optimization methods to avoid premature convergence. The disruption operator is mathematically represented as:

$$D_{op} = \begin{cases} D_{i,j} \times \delta(-2,2), \;\; if\; D_{i,j,best} \ge 1 \\ 1 + D_{i,j,best} \times \delta\left(-\frac{10^{-4}}{2}, \frac{10^{-4}}{2}\right), otherwise \end{cases} \quad (6)$$

Where $D_{i,j}$ signifies the Euclidean distance between the i^{th} and j^{th} solutions in the population, $D_{i,j,best}$ denotes the Euclidean distance between the i^{th} solution and the best solution identified so far, and $\delta(x, y)$ is a random value generated within the interval $[x, y]$.

The operator dynamically adjusts its impact based on the proximity of solutions to the best-known solution. If $D_{i,j,best} \geq 1$, a larger variation is introduced, allowing for greater exploration in the search space. Otherwise, a minor variation is applied, encouraging fine-tuned exploitation around the best solution. This design ensures that the algorithm strikes a balance between discovering new areas in the search space and refining existing solutions, thereby enhancing overall optimization performance.

4 Methodology

KOA is a physics-inspired metaheuristic algorithm based on Kepler's laws of planetary motion. These laws define the motion of planets around the sun in elliptical orbits, the relationship between areas swept by the planets, and the proportionality between the square of their orbital period and the cube of their semi-major axis [22]. KOA applies these ideas to mimic optimization such that the planets are treated as potential solutions, while the sun is treated as the best. The algorithm starts by using an initial population of planets characterized by a given orbital eccentricity and spin period. The initialization is specified as below.

$$X_i^j = X_{i,lb}^j + rand \times \left(X_{i,ub}^j - X_{i,lb}^j\right), \quad i=1, 2, \ldots, N; \ \ j=1, 2, \ldots, D \tag{7}$$

$$e_i = rand, i=1, 2, \ldots, N \tag{8}$$

$$OP_i = |rand|, i=1, 2, \ldots, N \tag{9}$$

Where D represents the problem's dimensionality, N is the population size, $X_{i,lb}^j$ and $X_{i,ub}^j$ are the lower and upper bounds for the j^{th} variable, and rand is a random number in the interval [0, 1]. This reflects the binary nature of the feature selection problem, where each feature can either be included (1) or excluded (0) from the subset. These normalized bounds ensure that the optimization begins within a valid real-valued range before binary conversion via the sigmoid-based transformation."

The planets rotate around the sun in elliptical orbits, undergoing two phases: moving closer to the sun and moving away. The gravitational force between the sun and a planet, which governs the planet's motion, is calculated as:

$$F_{gi}(t) = e_i \cdot \mu(t) \cdot \frac{M_s \cdot m_i}{R_i^2 + \epsilon} + r_1 \tag{10}$$

Where M_s and m_i represent the normalized masses of the sun and the planet, calculated as follows:

$$M_s = r_2 \cdot \frac{\text{fit}_s(t) - \text{worst}(t)}{\sum_{k=1}^{N}\left(\text{fit}_k(t) - \text{worst}(t)\right)} \tag{11}$$

$$m_i = \frac{fit_i(t) - worst(t)}{\sum_{k=1}^{N}\left(fit_k(t) - worst(t)\right)} \tag{12}$$

Where $\mu(t) = \mu_0 . exp(-\gamma . t/T)$ is the gravitational constant, $R_i = \sqrt{\sum_{j=1}^{d}\left(X_{sj}(t) - X_{ij}(t)\right)^2}$ is the distance between the planet and the sun, and r_1 and r_2 are random values, and ϵ ϵ is a small constant.

The velocity of a planet, influenced by its distance from the sun, is updated as follows:

$$\vec{v}_i(t) \begin{cases} \delta \cdot \left(2r_4 \cdot \overrightarrow{X_l} - \overrightarrow{X_b}\right) + \delta' \cdot \left(\overrightarrow{X_a} - \overrightarrow{X_b}\right) + \left(1 - R_{\text{norm}}(t)\right). \\ \quad \sigma \cdot \overrightarrow{U_1} \cdot r_5 \cdot \left(\overrightarrow{X_{i,ub}} - \overrightarrow{X_{i,lb}}\right), \quad R_{\text{norm}}(t) \leq 0.5 \\ \quad r_4 \cdot \kappa \cdot \left(\overrightarrow{X_a} - \overrightarrow{X_l}\right) + \left(1 - R_{\text{norm}}(t)\right) \cdot \sigma \cdot \\ \quad \overrightarrow{U_2} \cdot r_5 \cdot \left(r_3 \cdot \overrightarrow{X_{i,ub}} - \overrightarrow{X_{i,lb}}\right), \quad \text{otherwise} \end{cases} \tag{13}$$

The position is then updated as follows:

$$\overrightarrow{X_l}(t+1) = \overrightarrow{X_l}(t) + \sigma \cdot \overrightarrow{v_l}(t) + \left(F_{gi}(t) + |r|\right) \cdot \vec{U} \cdot \left(\overrightarrow{X_s}(t) - \overrightarrow{X_l}(t)\right) \tag{14}$$

In the second stage, KOA refines the planetary positions around the sun using an adaptive factor h and the exploration formula:

$$\overrightarrow{X_l}(t+1) = \overrightarrow{X_l}(t) \cdot \overrightarrow{U_1} + \left(1 - \overrightarrow{U_1}\right) \cdot \left(\frac{\overrightarrow{X_l}(t) + \overrightarrow{X_j}(t) + \overrightarrow{X_a}(t)}{3} + h \cdot \left(\frac{\overrightarrow{X_l}(t) + \overrightarrow{X_j}(t) + \overrightarrow{X_a}(t)}{3} - \overrightarrow{X_b}(t) \right) \right) \tag{15}$$

$$h = \frac{1}{e^{\eta r}}, \ \ \eta = (l-1).r_4 + 1, \ \ l = -1 - 1.\left(\frac{t\%T}{T} . \frac{T}{T}\right)$$

Balancing exploration (broad search) and exploitation (fine-grained tuning), KOA effectively finds global optimum solutions in high-dimensional search spaces. Its physical inspiration maintains a balanced optimization process that can be applied to various applications.

EKOA is an improvement on the traditional KOA. EKOA addresses the weakness in the first algorithm, i.e., poor convergence for high-dimensional issues, an insufficient balance between exploration and exploitation, and sub-standard handling of complex solution spaces. EKOA achieves higher accuracy, rapid convergence, and solution diversity by utilizing new strategies, i.e., dynamic fine-tuning, oscillatory chaotic force, cross-direction solution creation, and elite-based optimization.

The adaptive adjusting policy dynamically updates the weight between exploration and exploitation in every iteration. At the initial phases, EKOA emphasizes exploration to thoroughly explore the search area. With increasing iterations, the algorithm transfers the focus step by step toward exploitation, adjusting the promising area for the optimum solution. The weighing is mathematically represented as:

$$w = w_{min} + (w_{max} - w_{min}) \cdot \frac{t}{T} \tag{16}$$

Where w_{min} and $w_{\max}$ are the minimum and maximum weights, respectively, t stands for the ongoing iteration, and T is the total number of iterations. This adaptability prevents the algorithm from prematurely converging to

local optima, ensuring a more robust search across the solution space.

In traditional KOA, the gravitational constant $\mu(t)$ gradually decreases to focus the search around promising areas. EKOA improves this process by introducing an oscillatory chaotic force constant, which dynamically modulates gravitational force to increase diversity in solutions and prevent stagnation. The gravitational constant is updated as:

$$\mu(t) = s_{map}(t) + \mu_0 \cdot exp\left(-\frac{\gamma t}{T}\right) \quad (17)$$

Where $s_{map}(t)$ is an oscillatory chaotic function calculated as follows.

$$s_{map}(t+1) = \alpha \cdot sin\left(\pi \cdot s_{map}(t)\right) \quad (18)$$

Where μ_0 is the initial gravitational constant, γ is a decay factor, and T represents the total cycle count. This chaotic mechanism ensures greater randomness in gravitational influence, allowing EKOA to escape local optima and maintain diverse solutions.

The crosswise solution generation strategy accelerates convergence by improving population diversity and generating new candidate solutions. Based on their current positions, two "satellite" solutions are created around existing solutions. The crossover equations are:

$$KX_{a,j}(t) = r_1 \cdot X_{a,j}(t) + (1 - r_1) \cdot X_{b,j}(t) + c_1 \cdot \left(X_{a,j}(t) - X_{b,j}(t)\right) \quad (19)$$

$$KX_{b,j}(t) = r_1 \cdot X_{b,j}(t) + (1 - r_2) \cdot X_{a,j}(t) + c_2 \cdot \left(X_{b,j}(t) - X_{a,j}(t)\right) \quad (20)$$

Where r_1 and r_2 are random values between [0, 1], and c_1 and c_2 are constants controlling the influence of each parent solution. If the satellite solutions improve the fitness value, they replace their parent solutions, ensuring that the population progressively improves over iterations.

The elite-driven optimization strategy focuses on refining the best solutions to enhance convergence accuracy and precision. It combines three sub-strategies: elite movement, elite cooperation, and elite-driven optimization. Elite movement refines elite solutions by adding perturbations based on their distance from non-elite solutions:

$$GX_i(t) = X_i(t) + A_1 \cdot D_1 + \Delta h$$
$$A_1 = l_1.(r_1 - 0.5) + 1 \quad (21)$$
$$D_1 = 2 \cdot r_2 \cdot X_{\text{best}}(t) - X_i(t)$$

Where Δh is a refinement term derived from the Levy flight mechanism.

Elite solutions collaborate by sharing information to improve population diversity:

$$GX_i(t) = X_r(t) + r_1 \cdot D_3 + \Delta h \quad (22)$$

Where $D_3 = X_a(t) - X_b(t)$ and $X_r(t)$ is a randomly selected elite solution.

Elite-driven optimization focuses on aggressively refining elite solutions:

$$GX_i(t) = X_r(t) + A_2 \cdot D_2 + \Delta h \quad (23)$$

Where $D_2 = X_i(t) - X_r(t)$ and A_2 is a randomly selected elite solution.

As shown in Figures 1 and 2, the EKOA workflow begins with the initialization of the population, where the positions, velocities, and orbital parameters of the solutions are set within the defined bounds. The algorithm evaluates the fitness of each solution in light of the optimization objective. In subsequent iterations, the lateral crossover generates new candidate solutions, the oscillatory chaotic force changes the search region, and the adaptive weight allows for smooth movement between exploration and exploitation. Finally, the elite-driven optimization refines the top-performing solutions, consistently enhancing the population's quality. The loop terminates when the terminating criteria are met, e.g., a fixed number of iterations or a convergence point.

The enhanced algorithm is a multi-target anomaly-based IDS for IoT. The algorithm employs Pareto dominance to attain a practical compromise between conflicting objectives, e.g., enhancing classification accuracy but reducing the number of features chosen to be analyzed. Non-dominated solutions are preserved in each iteration through a repository-based structure, which presents different Pareto-optimal solutions. EKOA optimizes leader solutions in each iteration, chosen from the repository through a roulette-wheel selection algorithm in conjunction with hypercube scores and the Boltzmann function. The leader is therefore guaranteed to be a good choice for optimizing the process.

The repository is divided into two components: the grid and the controller. The grid organizes solutions for easier assessment and diversity, and the controller decides whether new solutions are to be added to the repository. To improve the repository's quality, dominated solutions are purged at fixed intervals so that high-quality solutions are retained. This architecture promotes a well-distributed Pareto front for the opposing accuracy and feature reduction objectives.

The algorithm initializes a randomly started population set of solutions and their positions, speeds, and gravity constants. Non-dominated solutions are moved apart in a repository, while dominated members are preserved in the base population. The leader solution is chosen in the initial step of every iteration from the repository through the roulette-wheel technique, under the governance of the Boltzmann function. The EKOA framework applies its mechanisms, including adaptive weight adjustment, oscillatory chaotic force, crosswise solution generation, and elite-driven optimization strategies, to effectively explore and exploit the solution space. Since feature selection is a binary problem, solutions are converted from the discrete domain to the binary domain using Eq. 24.

$$y_i^{j+1} = f(x)\begin{cases} 1, & if\ w\left(x_i^{j+1}\right) \geq r_{rand} \\ 0, & otherwise \end{cases}$$
$$w(a) = \frac{1}{1 + e^{10(a-0.5)}} \quad (24)$$

This change ensures the algorithm yields a binary code for the feature subset. The optimization step is followed by adding new non-dominated points to the

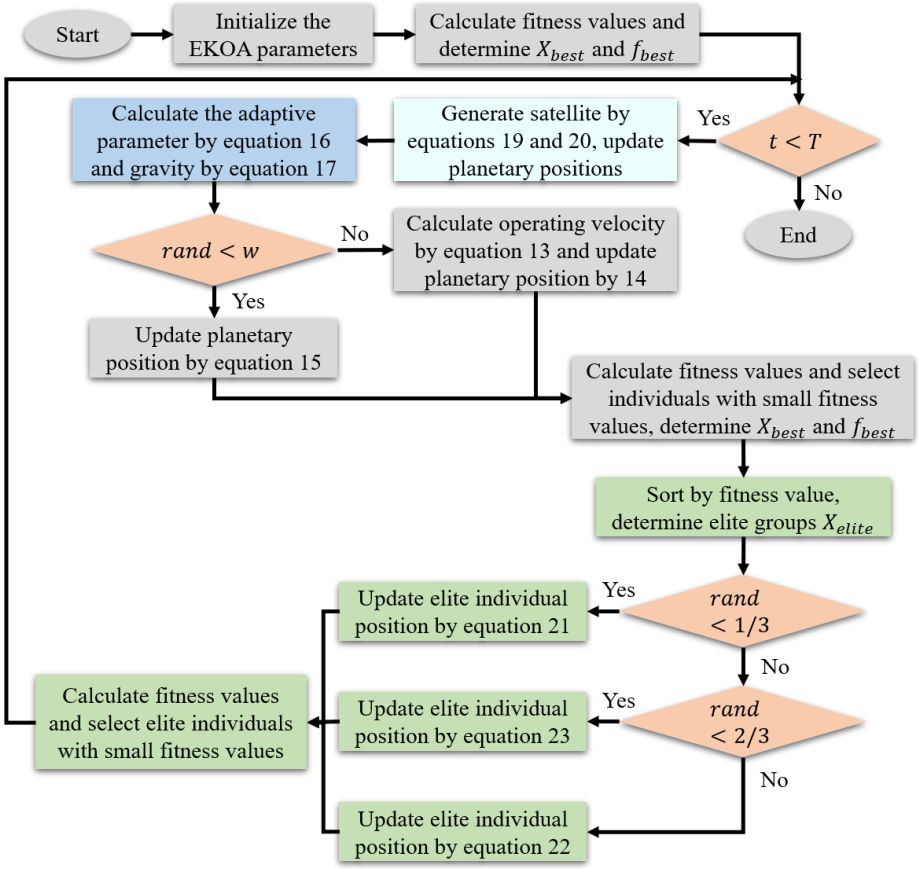

Figure 1: Flowchart of EKOA

repository and eliminating dominated ones. If the repository size exceeds, the lesser-quality points are eliminated in favor of higher-quality points. The disruption operator is called at regular intervals to introduce controlled randomness to prevent stagnation. The algorithm terminates when the maximum iterations are exceeded or a predetermined convergence criteria are met.

5 Results

To evaluate the performance of the proposed EKOA-based intrusion detection system, experiments were

Input:
Population size N
Maximum number of iterations T
Search space dimensionality D
Output:
Optimal solution X_{best}
Best fitness value f_{best}
Initialize parameters: gravitational constant μ_0, decay factor γ, and total iterations T
Generate initial population using Eq. 7; assign orbital eccentricity using Eq. 8, and compute orbital period via Eq. 9
Evaluate the initial fitness for each candidate; identify the best individual, and set its fitness; initialize iteration count
Repeat until $t \geq T$:
For each individual $i = 1$ *to* N:
Compute gravitational attraction using Eq. 10
Measure distance to the current best solution using Eq. 12
Determine velocity using Eq. 13
If a random number < threshold:
Update position with Eq. 14
Else:
Use Eq. 15 for position update
Recalculate fitness; if this new fitness improves upon f_{best}, update both f_{best} and X_{best}
Increment iteration count: $t = t + 1$
End

Figure 2: Pseudo-code

conducted on three datasets: Mirai and Gafgyt. The datasets were normalized and encoded according to standard preprocessing and labeling methods. Feature selection was performed according to the proposed binary multi-objective EKOA algorithm. Table 2 shows the complete set of algorithms and classifier hyperparameters, like population size, number of iterations, and crossover coefficients. The levels for these variables were selected after preliminary tuning for stable convergence and desirable search behavior. The 5-fold cross-validation method was chosen for statistical robustness. The experiments were repeated five times, and the means were calculated across performance metrics.

EKOA was evaluated on ten feature selection datasets and nine botnet detection datasets, as indicated in Tables 3 and 4. The datasets had at least 100,000 samples, and some even exceeded a million. Table 5 provides the nature of the datasets, e.g., normal/abnormal class ratios.

Mirai botnet attacks were utilized for the training set (70%), while Gafgyt botnet attacks comprised the test set (30%). This was to keep the model robust, as we are training on attacks, we are aware of, but testing on the ability to identify new patterns previously unseen. The test was performed in MATLAB on an Intel Core i5-8400 processor computer, running 8 GB of RAM.

Feature selection experiments compared EKOA against five multi-objective algorithms: MOHHOFOA [23], NSGA-IIFS [24], B-MOABCFS [25], and MOPSOFS [26]. The Hyper-Volume (HV) and several feature subsets (FN) metrics were used to evaluate

Table 2: Hyperparameter settings for algorithms and classifiers

Component	Parameter	Range	Description
EKOA	μ_0	0.1	Initial gravitational constant for attraction force
	γ	15	Gravitational decay factor controlling convergence speed
	w_{min}	0.4	Minimum adaptive weight for exploration
	w_{max}	0.9	Maximum adaptive weight for exploitation
	c_1 and c_2	Uniform [–1, 1]	Crossover coefficients in lateral crossover mechanism
	Population size	30	Number of individuals in the population
	Max iterations	100	Maximum number of optimization iterations
KNN	k-value	3	Number of neighbors used for classification
Decision tree	Max depth	None (default)	Tree expansion continues until full purity or constraint
SVM	Kernel	RBF	Radial basis function kernel for non-linear classification

Table 3: Summary of datasets used for feature selection

Dataset	No. of features	No. of classes	No. of samples
Yale_64	1024	15	165
CNAE-9	857	9	540
LSVT	309	2	126
Musk	167	2	476
Urban land cover	148	9	507
Hill-valley	100	2	606
Sonar	60	2	208
Ionosphere	34	2	351
Vehicle	18	4	846
Vowel	10	11	990

performance. Table 6 presents the HV results for the ten datasets, which measure solution convergence and diversity. Table 7 reports the FN values (average and standard deviation) to assess the effectiveness of dimensionality reduction. EKOA's classification relies on K-Nearest Neighbors (KNN) and Leave-One-Out Correlation (LOOCV) scores to measure classification errors. Experiments demonstrated EKOA's ability to effectively optimize high classification accuracy feature subsets and outperform traditional multi-objective algorithms.

EKOA optimized both the anomaly detection and feature selection for botnet detection. Non-dominated solutions with lowest error rates in each iteration were saved in a second external archive. Table 8 is a comparison between EKOA and other algorithms, which indicates that EKOA performs better than all the algorithms in all metrics: True Positive Rate (TPR), True Negative Rate (TNR), False Alarm Rate (FAR), accuracy, Area Under the Curve (AUC), and Geometric Mean (G-mean)

Accuracy measures the proportion of correctly labeled records, combining True Negatives (TNs) and True Positives (TPs) over the total population:

$$A = \frac{TN + TP}{FN + FP + TN + TP} \tag{25}$$

FAR evaluates the proportion of False Positives (FPs) among standard samples:

$$FAR = \frac{FP}{TN + FP} \tag{26}$$

TPR or sensitivity quantifies the percent of true positives identified successfully:

$$TPR = \frac{TP}{TP + FN} \tag{27}$$

TNR or specificity determines the percentage of true negatives recorded:

$$TNR = \frac{TN}{TN + FP} \tag{28}$$

G-mean balances sensitivity and specificity, providing a harmonic mean between TPR and TNR. AUC measures the relationship between TPR and FAR over a range of classification thresholds:

$$AUC = \frac{TPR.FAR}{2} + \frac{(1 + TPR).(1 - FAR)}{2} \tag{29}$$

The consistently superior performance of EKOA across datasets is primarily due to its hybrid optimization

Table 4: Summary of datasets used for botnet detection

Dataset	Bashlite (%)	Mirai (%)	Anomaly (%)	Normal (%)	No. of records	No. of features
Ennio doorbell	89	0	89	11	3,55,506	115
Samsung webcam	86	0	86	14	3,75,228	115
Monitoring equipment XC1003	39	59	98	3	8,15,237	115
Monitoring equipment XC1002	37	58	94	6	8,29,079	115
Ecobee thermostat	38	61	98	2	8,35,887	115
Monitoring equipment PT838	37	51	88	12	8,36,902	115
Monitoring equipment PT737	40	52	92	7	8,28,271	115
Danmini doorbell	31	64	95	5	10,18,309	115
Baby monitor	28	55	84	16	10,98,688	115

structure, which is well-aligned with the nature of IoT botnet detection. EKOA's adaptive strategy dynamically shifts the focus from exploration to exploitation, improving convergence without overfitting. Sinusoidal chaotic force addition introduces controlled randomness to enhance population diversity, which is crucial in avoiding a local optimum because of redundancy or noise that is common in high-dimensional data from IoT. The elite-guided aspect also introduces localized optimization for possibly good candidates, in such a way that compact and effective sub-sets of features are chosen, leading to higher accuracy in the classifier.

To assess the proposed framework's ability to generalize across unseen botnet types, a cross-family evaluation was conducted. Specifically, two scenarios were tested:

- Scenario 1: Training on Mirai samples and testing on Gafgyt samples
- Scenario 2: Training on Gafgyt samples and testing on Mirai samples

These setups simulate real-world IoT environments where the intrusion detection system must detect novel attack variants without prior exposure during training. The results for both scenarios using KNN and DT classifiers are summarized in Table 9.

These results consolidate that the resultant EKOA-based feature selection method facilitates successful generalizability in new attack patterns. Surprisingly, the performance is marginally higher for the KNN classifier under domain shift scenarios. The reason is that EKOA can weed out noise in the datasets and emphasize behavior-centric patterns usable for different families of botnets.

6 Discussion

The proposed EKOA-based intrusion detection system exhibits clear comparative benefits compared to the diversity of state-of-the-art methods in Table 1. The different methods all contribute to metaheuristic-based feature choice or hybrid detection methods. Nevertheless, EKOA presents clear performance benefits in various dimensions, such as generalizability, convergence speed, and deployability.

In experiments on typical test datasets such as Mirai and Gafgyt, the EKOA framework always achieved detection accuracy greater than 99% and reduced the set of features by 35%. This is on par with methods such as GWO-SVM and MHADMA-BCIDL, which performed with high accuracy in narrow-use cases but were evaluated on less inclusive datasets or a few malware types. EKOA's consistent performance on diverse attack types, e.g., DDoS, data exfiltration, and command-and-control traffic, shows higher generalizability to new threats.

From an algorithmic point of view, EKOA addresses several weaknesses characteristic of metaheuristic-based detection systems. Such approaches as SSA–ALO and

Table 5: Distribution of botnet-related classes in training and testing sets

Dataset	Testing set (%)		Training set (%)	
	First class	Second class	First class	Second class
Monitoring equipment XC1003	Gafgyt (95)	Normal (5)	Mirai (96)	Normal (4)
Monitoring equipment XC1002	Gafgyt (85)	Normal (15)	Mirai (92)	Normal (8)
Ecobee thermostat	Gafgyt (94)	Normal (6)	Mirai (96)	Normal (4)
Monitoring equipment PT838	Gafgyt (76)	Normal (14)	Mirai (82)	Normal (18)
Monitoring equipment PT737	Gafgyt (84)	Normal (16)	Mirai (86)	Normal (14)
Danmini doorbell	Gafgyt (85)	Normal (15)	Mirai (91)	Normal (9)
Baby monitor	Gafgyt (65)	Normal (35)	Mirai (78)	Normal (22)

Table 6: HV results for feature selection experiments

HV	MOHHOFOA	B-MOABCFS	NSGA-IIFS	MOPSOFS	EKOA
	Std/average	Std/average	Std/average	Std/average	Std/average
Yale_64	0.009/0.687	0.003/0.752	0.0135/0.448	0.005/0.645	0.002/0.771
CNAE-9	0.011/0.823	0.011/0.834	0.019/0.487	0.008/0.755	0.006/0.852
LSVT	0.029/0.768	0.072/0.822	0.006/0.404	0.004/0.752	0.025/0.881
Musk	0.005/0.936	0.008/0.942	0.014/0.618	0.022/0.894	0.003/0.957
Urban land cover	0.005/0.884	0.007/0.871	0.024/0.596	0.011/0.836	0.003/0.892
Hill-valley	0.004/0.649	0.021/0.631	0.017/0.531	0.007/0.652	0.002/0.932
Sonar	0.004/0.91	0.005/0.913	0.021/0.699	0.016/0.891	0.003/0.922
Ionosphere	0.002/0.93	0.003/0.927	0.091/0.841	0.003/0.922	0.003/0.944
Vehicle	0.006/0.684	0.006/0.693	0.033/0.613	0.007/0.692	0.003/0.724
Vowel	0.001/0.826	0.001/0.828	0.03/0.815	0.009/0.826	0.009/0.839

Table 7. Feature subset results for feature selection experiments

FN	MOHHOFOA	B-MOABCFS	NSGA-IIFS	MOPSOFS	EKOA
	Std/average	Std/average	Std/average	Std/average	Std/average
Yale_64	1.27/8.45	2.46/12.56	1.91/5.55	1.66/3.45	1.11/10.28
CNAE-9	2.15/8.92	3.66/9.29	2.44/7.41	0.44/5.53	2.66/10.49
LSVT	1.04/5.23	1.25/4.51	0.83/3.39	3.45/5.02	1.26/6.23
Musk	1.73/14.03	2.75/10.86	3.55/6.8	1.42/10.05	1.99/15.74
Urban land cover	2.01/14.05	2.53/11.42	2.56/9.22	1.88/10.55	1.13/14.53
Hill-valley	1.21/9.15	2.76/9.26	1.11/7.02	2.71/8.25	0.45/9.21
Sonar	1.42/11.1	1.76/11.02	1.63/5.81	1.96/10.23	2.28/12.18
Ionosphere	0.71/7.22	0.88/7.26	1.15/5.22	0.71/6.41	0.41/7.56
Vehicle	0.47/5.23	0.36/5.42	0.22/4.13	0.41/5.35	0.31/5.91
Vowel	0/9.01	0/9.01	0.19/8.34	0.44/8.44	0/9

Table 8: Comparative performance analysis of EKOA and other algorithms for botnet detection

Datasets	Algorithms	AUC	G-mean	TPR	TNR	FAR	Accuracy
Monitoring equipment XC1003	MOHHOFOA	0.88	0.87	0.94	0.82	0.18	0.89
	NSGA-IIFS	0.68	0.67	0.84	0.54	0.47	0.69
	B-MOABCFS	0.83	0.82	0.91	0.74	0.26	0.84
	MOPSOFS	0.67	0.65	0.83	0.52	0.48	0.68
	EKOA	0.98	0.98	0.97	0.99	0.08	0.98
Monitoring equipment XC1002	MOHHOFOA	0.89	0.87	0.92	0.86	0.14	0.89
	NSGA-IIFS	0.68	0.67	0.73	0.62	0.39	0.68
	B-MOABCFS	0.74	0.73	0.61	0.87	0.14	0.69
	MOPSOFS	0.62	0.61	0.78	0.48	0.53	0.64
	EKOA	0.98	0.98	0.98	0.98	0.02	0.97
Ecobee thermostat	MOHHOFOA	0.89	0.88	0.94	0.85	0.17	0.9
	NSGA-IIFS	0.72	0.71	0.87	0.57	0.44	0.72
	B-MOABCFS	0.85	0.86	0.91	0.82	0.18	0.87
	MOPSOFS	0.77	0.77	0.78	0.73	0.28	0.78
	EKOA	0.99	0.99	0.99	0.98	0.05	0.98
Monitoring equipment PT838	MOHHOFOA	0.9	0.9	0.95	0.85	0.16	0.91
	NSGA-IIFS	0.76	0.74	0.92	0.58	0.41	0.78
	B-MOABCFS	0.81	0.81	0.9	0.7	0.28	0.82
	MOPSOFS	0.77	0.77	0.86	0.67	0.34	0.78
	EKOA	0.98	0.98	0.96	0.98	0.009	0.98
Monitoring equipment PT737	MOHHOFOA	0.79	0.79	0.92	0.68	0.33	0.82
	NSGA-IIFS	0.65	0.63	0.82	0.49	0.51	0.66
	B-MOABCFS	0.76	0.75	0.88	0.64	0.36	0.78
	MOPSOFS	0.64	0.61	0.84	0.44	0.55	0.65
	EKOA	0.97	0.97	0.98	0.95	0.08	0.97
Danmini doorbell	MOHHOFOA	0.87	0.86	0.93	0.82	0.19	0.88
	NSGA-IIFS	0.66	0.63	0.88	0.44	0.56	0.69
	B-MOABCFS	0.77	0.74	0.93	0.59	0.41	0.84
	MOPSOFS	0.69	0.67	0.87	0.52	0.48	0.71
	EKOA	0.98	0.98	0.97	0.91	0.04	0.98
Baby monitor	MOHHOFOA	0.92	0.92	0.96	0.88	0.12	0.92
	NSGA-IIFS	0.68	0.68	0.76	0.62	0.39	0.67
	B-MOABCFS	0.83	0.83	0.79	0.89	0.11	0.83
	MOPSOFS	0.71	0.71	0.61	0.82	0.18	0.71
	EKOA	0.97	0.97	0.99	0.94	0.06	0.97

MODHHO are typically susceptible to premature convergence or limited diversity in solution space, potentially inhibiting precision or becoming unstable. EKOA integrates four strategic enhancements to overcome such shortcomings:

- Dynamic adjustment strategy: adapts parameters dynamically based on search progress, balancing stably between exploration and exploitation.
- Oscillatory chaotic force: introduces controlled randomness to prevent stagnation and enhance escape from local optima.
- Crosswise Solution Generation: enhances diversity in candidate solutions in later iterations.
- Elite-driven optimization: ensures that high-performance solutions control the evolutionary process, increasing the likelihood of a global optimum.

Such processes cause EKOA to converge faster than classical evolutionary techniques but without a loss in solution quality. For example, in comparison with BESO-HDLBD and SFO-WOA-PSO, which involve the use of high-level neural structures or multi-level optimization steps, EKOA discovers optimal or near-optimal ensembles of features in fewer iterations and with much less computational expenditure.

Feasibility in practical deployments is of concern for IoT-driven intrusion detection systems, which are usually resource-limited and need a low-latency response. EKOA is suitable for such an environment because:

Table 9: Cross-attack generalization results using EKOA-selected features

Scenario	Classifier	Accuracy	TPR (Recall)	FAR	TNR	G-Mean	AUC
Train: Mirai → Test: Gafgyt	KNN	96.7%	95.6%	4.3%	95.7%	95.6%	95.5%
Train: Mirai → Test: Gafgyt	DT	95.4%	93.7%	5.1%	94.9%	94.3%	94.1%
Train: Gafgyt → Test: Mirai	KNN	97.2%	96.0%	3.8%	96.2%	96.1%	95.9%
Train: Gafgyt → Test: Mirai	DT	95.8%	94.3%	4.7%	95.3%	94.8%	94.6%

- Low execution time: Feature selection using EKOA is computationally lightweight and does not depend on deep learning backbones or large ensemble models.
- Classifier compatibility: The system leverages efficient classifiers (KNN and decision tree), which are known for fast inference times and ease of integration on edge devices.
- Scalability: The modular design allows the framework to be deployed on distributed or hierarchical architectures such as cloud-edge systems, IoT gateways, and embedded devices.

These advantages make EKOA a compelling and high-performance substitute for more advanced or specialized intrusion detection methods. It perfectly balances speed, accuracy, and scalability, the essential properties for real-time IoT network security in high-speed applications.

7 Conclusion

This paper proposed an EKOA-driven optimal IoT security feature selection intrusion detection system. EKOA incorporates adaptive control, chaotic force modulation, cross-sectional solution construction, and elite-based fine-tuning to promote convergence and robustness. Experimental verifications demonstrated higher detection accuracy and reduced feature dimensionality against state-of-the-art contemporary multi-objective methods on standard benchmark sets. Future work will extend the system for real-time intrusion detection based on online learning models. Secondly, realization in realistic-edge scenarios and exploring transfer learning methods between IoT applications will be attempted to enhance adaptability and scalability.

Conflict of interest

The authors declare that they have no conflicts of interest.

References

[1] K. Halimi, A. Hadjadj, Z. Kouahla, and B. Farou, "A Fuzzy Logic-Driven Semantic and Binary Tree-Based Indexing Framework for Scalable IoT Data Storage and Retrieval," *Informatica,* vol. 49, no. 24, 2025, doi: https://doi.org/10.31449/inf.v49i24.8039.

[2] R. M. Ghadban, H. Z. Neima, and H. A. Jasim, "A Blockchain-Based Security Framework for IoT Networks: Design, Implementation, and Evaluation," *Informatica,* vol. 49, no. 24, 2025, doi: https://doi.org/10.31449/inf.v49i24.8122.

[3] B. Bala and S. Behal, "AI techniques for IoT-based DDoS attack detection: Taxonomies, comprehensive review and research challenges," *Computer science review,* vol. 52, p. 100631, 2024, doi: https://doi.org/10.1016/j.cosrev.2024.100631.

[4] T. Al-Shurbaji *et al.*, "Deep Learning-Based Intrusion Detection System For Detecting IoT Botnet Attacks: A Review," *IEEE Access,* 2025, doi: https://doi.org/10.1109/ACCESS.2025.3526711.

[5] O. Malkawi, N. Obaid, and W. Almobaideen, "Intrusion Detection System for 5G Device-to-Device Communication Technology in Internet of Things," *Informatica,* vol. 48, no. 15, 2024, doi: https://doi.org/10.31449/inf.v48i15.4646.

[6] E. Rivandi and R. Jamili Oskouie, "A Novel Approach for Developing Intrusion Detection Systems in Mobile Social Networks," *Available at SSRN 5174811,* 2024, doi: https://dx.doi.org/10.2139/ssrn.5174811.

[7] A. Heidari and M. A. Jabraeil Jamali, "Internet of Things intrusion detection systems: a comprehensive review and future directions," *Cluster Computing,* vol. 26, no. 6, pp. 3753-3780, 2023, doi: https://doi.org/10.1007/s10586-022-03776-z.

[8] N. Mohamed, "Artificial intelligence and machine learning in cybersecurity: a deep dive into state-of-the-art techniques and future paradigms," *Knowledge and Information Systems,* pp. 1-87, 2025, doi: https://doi.org/10.1007/s10115-025-02429-y.

[9] J. Azimjonov and T. Kim, "Stochastic gradient descent classifier-based lightweight intrusion detection systems using the efficient feature subsets of datasets," *Expert Systems with Applications,* vol. 237, p. 121493, 2024, doi: https://doi.org/10.1016/j.eswa.2023.121493.

[10] J. Li, M. S. Othman, H. Chen, and L. M. Yusuf, "Optimizing IoT intrusion detection system: feature selection versus feature extraction in machine learning," *Journal of Big Data,* vol. 11, no. 1, p. 36, 2024, doi: https://doi.org/10.1186/s40537-024-00892-y.

[11] K. Harahsheh, R. Al-Naimat, and C.-H. Chen, "Using Feature Selection Enhancement to Evaluate Attack Detection in the Internet of Things Environment," *Electronics,* vol. 13, no. 9, p. 1678, 2024, doi: https://doi.org/10.3390/electronics13091678.

[12] M. Ahmadi *et al.*, "Optimal allocation of EVs parking lots and DG in micro grid using two-stage GA-PSO," *The Journal of Engineering,* vol. 2023, no. 2, p. e12237, 2023, doi: https://doi.org/10.1049/tje2.12237.

[13] H. Haddadpajouh, A. Mohtadi, A. Dehghantanaha, H. Karimipour, X. Lin, and K.-K. R. Choo, "A multikernel and metaheuristic feature selection approach for IoT malware threat hunting in the edge layer," *IEEE Internet of Things Journal,* vol. 8, no. 6, pp. 4540-4547, 2020, doi: https://doi.org/10.1109/JIOT.2020.3026660.

[14] R. Abu Khurma, I. Almomani, and I. Aljarah, "IoT botnet detection using salp swarm and ant lion hybrid optimization model," *Symmetry,* vol. 13, no. 8, p. 1377, 2021, doi: https://doi.org/10.3390/sym13081377.

[15] F. Hosseini, F. S. Gharehchopogh, and M. Masdari, "A botnet detection in IoT using a hybrid multi-objective optimization algorithm," *New Generation Computing,* vol. 40, no. 3, pp. 809-843, 2022, doi: https://doi.org/10.1007/s00354-022-00188-w.

[16] F. S. Gharehchopogh, B. Abdollahzadeh, S. Barshandeh, and B. Arasteh, "A multi-objective mutation-based dynamic Harris Hawks optimization for botnet detection in IoT," *Internet of Things,* vol. 24, p. 100952, 2023, doi: https://doi.org/10.1016/j.iot.2023.100952.

[17] M. Alkhammash, "A Metaheuristic Approach to Detecting and Mitigating DDoS Attacks in Blockchain-Integrated Deep Learning Models for IoT Applications," *IEEE Access,* 2024, doi: https://doi.org/10.1109/ACCESS.2024.3519132.

[18] L. A. Maghrabi *et al.*, "Enhancing cybersecurity in the internet of things environment using bald eagle search optimization with hybrid deep learning," *IEEE Access,* vol. 12, pp. 8337-8345, 2024, doi: https://doi.org/10.1109/ACCESS.2024.3352568.

[19] M. Maazalahi and S. Hosseini, "Machine learning and metaheuristic optimization algorithms for feature selection and botnet attack detection," *Knowledge and Information Systems,* pp. 1-49, 2025, doi: https://doi.org/10.1007/s10115-024-02322-0.

[20] E. Elsedimy and S. M. AboHashish, "An intelligent hybrid approach combining fuzzy C-means and the sperm whale algorithm for cyber attack detection in IoT networks," *Scientific Reports,* vol. 15, no. 1, p. 1005, 2025, doi: https://doi.org/10.1038/s41598-024-79230-4.

[21] M. B. Bagherabad, E. Rivandi, and M. J. Mehr, "Machine Learning for Analyzing Effects of Various Factors on Business Economic," *Authorea Preprints,* 2025, doi: https://doi.org/10.36227/techrxiv.174429010.09842200/v1.

[22] M. Abdel-Basset, R. Mohamed, S. A. A. Azeem, M. Jameel, and M. Abouhawwash, "Kepler optimization algorithm: A new metaheuristic algorithm inspired by Kepler's laws of planetary motion," *Knowledge-based systems,* vol. 268, p. 110454, 2023, doi: https://doi.org/10.1016/j.knosys.2023.110454.

[23] B. Abdollahzadeh and F. S. Gharehchopogh, "A multi-objective optimization algorithm for feature selection problems," *Engineering with Computers,* vol. 38, no. Suppl 3, pp. 1845-1863, 2022, doi: https://doi.org/10.1007/s00366-021-01369-9.

[24] T. M. Hamdani, J.-M. Won, A. M. Alimi, and F. Karray, "Multi-objective feature selection with NSGA II," in *Adaptive and Natural Computing Algorithms: 8th International Conference, ICANNGA 2007, Warsaw, Poland, April 11-14, 2007, Proceedings, Part I 8*, 2007: Springer, pp. 240-247, doi: https://doi.org/10.1007/978-3-540-71618-1_27.

[25] E. Hancer, B. Xue, M. Zhang, D. Karaboga, and B. Akay, "Pareto front feature selection based on artificial bee colony optimization," *Information Sciences,* vol. 422, pp. 462-479, 2018, doi: https://doi.org/10.1016/j.ins.2017.09.028.

[26] B. Xue, M. Zhang, and W. N. Browne, "Particle swarm optimization for feature selection in classification: A multi-objective approach," *IEEE transactions on cybernetics,* vol. 43, no. 6, pp. 1656-1671, 2012, doi: https://doi.org/10.1109/TSMCB.2012.2227469.