Stardom University

Stardom Scientific Journal of Natural and Engineering Sciences

- Stardom Scientific Journal of Natural and Engineering Sciences -
Peer Reviewed Scientific Journal published twice
a year by Stardom University
2nd issue- 3rd Volume 2025
ISSN 2980-3756

هيئة تحرير مجلة ستاردوم العلمية للعلوم "الطبيعية والهندسية"

Enhanced HawkFish Optimization with Chaotic Memetic Refinement for Lightweight and Accurate Malware Detection

Ashraf Nadir Alswid[1] and Osman Nuri Uçan[2]

[1] **Altinbas University; Istanbul, Turkey**

213720515@ogr.altinbas.edu.tr

[2] **Altinbas University; Istanbul, Turkey**

osman.ucan@altinbas.edu.tr

Abstract:
This paper presents a hybrid framework for intelligent malware detection that integrates the Enhanced HawkFish Optimization Algorithm (EHFOA) with the Light Gradient Boosting Ma-chine (LightGBM). The proposed method addresses the challenges of high-dimensional feature spaces, suboptimal model configurations, and real-time detection efficiency by simultaneously performing feature selection and hyperparameter tuning through a multi-objective optimization strategy. EHFOA incorporates biologically inspired behaviors along with chaotic initialization, entropy-based diversity control, and memetic local search refinement to improve convergence stability and search accuracy. The optimized feature subset and classifier configuration are used to train a lightweight yet highly accurate LightGBM model. The framework was evaluated on three benchmark datasets—EMBER, CIC-MalMem2022, and MalImg—and compared against several state-of-the-art models, including PSO-LGBM, GWO-LGBM, CNN, LSTM, DBN, Random Forest, XGBoost, and SVM. Experimental results show that the proposed method achieved a classifica-tion accuracy of 96.87%, precision of 97.12%, recall of 96.45%, and an F1-score of 96.78%, with a false positive rate of 2.18%. The model achieved a 42% feature reduction, reducing the input space to 58 features, and required only 145 seconds for training and 0.012 seconds for inference per sample. Statistical validation confirmed the significance of the performance improvements ($p < 0.01$), while ROC and precision–recall curves highlighted the model's robustness under imbalanced class distributions. The results demonstrate that the EHFOA-Light framework offers an effective, scalable, and computationally efficient solution for advanced malware detection.

Keywords: Malware detection; Enhanced HawkFish Optimization Algorithm (EHFOA); LightGBM; Feature selection; Metaheuristic optimization.

Introduction

Malware threatens digital systems, data integrity, and user privacy [1]. Signature-based and heuristic detection methods are failing against zero-day and polymorphic malware as intrusions get more complex. Due to its pattern recognition abilities to identify unknown threats, machine learning-based techniques have grown in popularity [2]. The quality of features and classification model configuration greatly affect the performance of these approaches [3-6]. Recent malware detection research has used deep learning, hybrid models, attention processes, and synthetic data to increase accuracy, resilience, and generalization in identifying emerging threats. Zhang et al. [7] showed that properly capturing sequential behavioral patterns can improve malware classification by extracting API sequence characteristics using deep learning. By analyzing network traffic using hybrid big language models and synthetic data, Naseer et al. [8] improved classification against evasion techniques for obfuscated malware. Zhou et al. [9] proposed FAMCF, a few-shot learning method for Android malware family classification that performs well in limited-data circumstances, a typical malware dataset difficulty. Ghourabi [10] proposed an attention-based method to Android malware detection that dynamically focuses on the most important elements of the input for more accurate and interpretable categorization. Chen et al. [11] conducted an empirical assessment on data augmentation strategies for limited-data learning in NLP, providing insights applicable to cybersecurity scenarios with insufficient labeled data. Johansson et al. [12] examined ethical and meaningful uses of synthetic data in security-driven machine learning to promote online safety research. Kholgh and Kostakos [13] presented PAC-GPT, a GPT-3-based approach for generating synthetic network traffic to train malware detectors without attack traces. Chin and Corizzo [14] developed a continuous semi-supervised malware detection method that can learn from evolving data without retraining, making it ideal for real-time and adaptive threat detection systems. Shenderovitz and Nissim [15] created Bon-APT, a detection and attribution methodology that uses temporal segmentation of API calls to improve APT detection explainability and granularity. Zhou et al. [16] developed a co-processor-based introspection framework using Intel Management Engine to

identify hardware-level malware securely and without host system interference. Bensaoud and Kalita [17] suggested a CNN-LSTM hybrid model with transfer learning to classify malware using opcodes and API call sequences with good accuracy and flexibility across datasets. Finally, Shah et al. [18] introduced MalRed, a novel malware detection method based on red channel analysis of binary file color pictures. This technology turns malware binaries into visual representations for image classification models and a new malware analysis perspective. These papers demonstrate the rapid advancement of malware detection methods and the importance of integrating powerful machine learning architectures with unique data formats and augmentation approaches to handle the rising complexity of cyber threats.

Table 1. Summary of related works

Authors	Methodology	Limitation
Zhang et al. [7]	Deep learning on API sequences	Requires large labeled datasets; limited to API-only behavior
Naseer et al. [8]	LLM + synthetic data on network traffic	High resource demand; relies on quality of synthetic data
Zhou et al. [9]	Few-shot learning (meta-learning)	Sensitive to feature space quality and model tuning
Ghourabi [10]	Attention-based deep learning	Potential overfitting; no optimization of feature selection
Chen et al. [11]	Data augmentation survey	Not malware-specific; lacks implementation context
Johansson et al. [12]	Ethical synthetic data generation	Conceptual focus; lacks malwa evaluation
Kholgh & Kostakos [13]	GPT-3 based synthetic network traffic	Computational cost; not coupled with classifier optimization
Chin & Corizzo [14]	Continual semi-supervised learning	Relies on stable feature quality; performance may degrade with noise

Shenderovitz & Nissim [15]	Temporal segmentation of API calls (Bon-APT)	No optimization focus; requires detailed sequence logs
Zhou et al. [16]	Hardware-level detection (Intel ME)	Hardware dependent; limited general applicability
Bensaoud & Kalita [17]	CNN-LSTM + transfer learning	Complex model; high computational overhead; lacks optimization
Shah et al. [18]	Image-based malware classification (MalRed)	Indirect behavioral insight; less interpretable than traditional methods

1 Proposed Method

This section introduces a malware detection system that uses sophisticated optimization tactics in a machine learning pipeline to improve accuracy and efficiency. Use the En-hanced HawkFish Optimization Algorithm (EHFOA) to choose features and tune LightGBM classifier hyperparameters. The pipeline starts with raw malware data like API call sequences, opcode streams, or binary feature representations. Structured numerical vectors are created from these inputs through systematic feature extraction. To maximize population variety, EHFOA is begun using a chaotic map. EHFOA uses entropy-based diversity control, memetic local search, and adaptive energy decay to better explore the feature space and avoid local optima. The approach produces an optimum subset of features and a well-tuned classifier configuration for LightGBM model training.

2.1. Enhanced HawkFish Optimization Algorithm (EHFOA)

The Enhanced HawkFish Optimization Algorithm (EHFOA) is a metaheuristic search method developed to address the complex challenges of feature selection and hyperparameter optimization in high-dimensional classification tasks such as malware detection. It extends the base HawkFish Optimization Algorithm (HFOA) proposed by [19] with the introduction of three major improvements: entropy-based diversity control, memetic local search strategies, and adaptive energy decay dynamics. These enhancements collectively contribute to improving convergence

speed, solution quality, and robustness in highly nonlinear, multimodal search spaces.

1-Initialization with Chaotic Maps

The initialization of the candidate population in EHFOA is governed by a chaotic map rather than uniform random distributions, as is typical in traditional metaheuristics. Chaotic maps are deterministic processes that exhibit dynamic behavior akin to stochasticity, which helps in generating a diverse set of initial solutions.

Let $x_0 \in (0,1)$ be the initial chaotic seed and μ be a control parameter. The logistic map is a commonly used chaotic function given by:

$$\boldsymbol{x_{n+1} = \mu x_n(1 - x_n), \mu \in [3.57, 4]} \quad (1)$$

Each value x_n is then mapped to the decision space of each feature or hyperparameter. This chaotic initialization ensures a well-dispersed set of initial solutions, thereby enhancing the global search capacity in early iterations.

2. Enhanced Position Update Mechanism

EHFOA builds upon the base HFOA, which models the foraging and hunting behavior of hawks and fishes in nature. The position update equation in HFOA is inspired by Levy flight and predator-prey interactions. EHFOA modifies this by incorporating adaptive dynamics and energy-aware transitions. The position update for each agent is given by:

$$X_{t+1} = \begin{cases} X_t + r_1 \cdot (X_{\text{best}} - r_2 \cdot X_t), & \text{if } r_3 < 0.5 \\ X_t + \mathcal{L}(\beta) \cdot (X_t - X_{\text{worst}}), & \text{otherwise} \end{cases} \quad (2)$$

where:

- X_t is the current position at iteration t,
- X_{best} and X_{worst} are the best and worst positions found so far,
- $r_1, r_2, r_3 \sim \mathcal{U}(0,1)$ are random numbers,
- $\mathcal{L}(\beta)$ represents a Levy flight perturbation with stability index $\beta \in (1,2]$.

3- Entropy-Based Diversity Control

To prevent premature convergence and maintain exploration throughout the optimization process, EHFOA integrates entropy-based diversity monitoring. The entropy H of the population distribution is calculated as:

$$H = -\sum_{i=1}^{n} p_i \log(p_i) \quad (3)$$

where p_i is the normalized probability of the i-th gene (feature) being selected in the population. A threshold $H_{\min}$ is defined, and if $H < H_{\min}$.

4. Memetic Local Search

EHFOA incorporates a memetic search strategy by exploiting local information near the best solutions. For each elite agent, a local search is applied using the equation:

$$X'_{\text{elite}} = X_{\text{elite}} + \epsilon \cdot (X_{\text{rand}} - X_{\text{elite}}) \quad (4)$$

5. Adaptive Energy Decay

Inspired by energy loss in biological organisms, EHFOA simulates agent energy decay, which influences the transition between exploration and exploitation. The energy level $E(t)$ of an agent at iteration t is modeled as:

$$E(t) = E_0 \cdot \exp\left(-\alpha \cdot \frac{t}{T}\right) \quad (5)$$

where:

- E_0 is the initial energy,
- α is the decay rate,
- T is the maximum number of iterations.

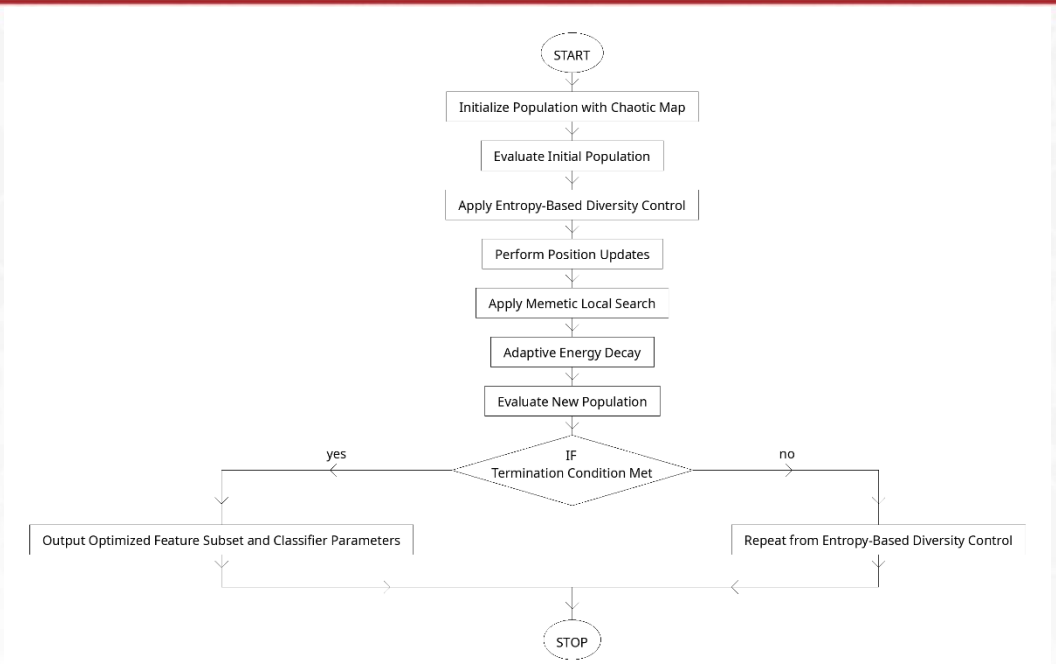
The energy value modulates the probability of switching strategies, i.e., between global search and local refinement, based on the agent's vitality.

6-Dual-Purpose Optimization

Unlike standard optimization algorithms, EHFOA is dual-purpose: it selects relevant feature subsets and simultaneously tunes classifier hyperparameters.

$$\text{Fitness}(\chi) = w_1 \cdot A + w_2 \cdot \text{F1} - w_3 \cdot \frac{|S|}{d} \quad (6)$$

where w_1, w_2, w_3 are scalar weights reflecting user-defined trade-offs. It represents a significant advancement over existing methods by integrating advanced metaheuristic concepts into a unified and scalable framework as shown in figure 2.



2.2 Light Gradient Boosting Machine (LightGBM)

At its core, LightGBM operates under the gradient boosting principle. Let the training dataset be defined as $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{n}$, where $x_i \in \mathbb{R}^d$ is a feature vector and y_i is the corresponding target label. Gradient boosting builds an ensemble of weak learners $f_k(x)$ iteratively to minimize a loss function $\mathcal{L}\big(y_i, F(x_i)\big)$, where $F(x)$ is the aggregated model output:

$$F(x) = \sum_{k=1}^{K} f_k(x) \tag{7}$$

At iteration k, the goal is to add a new function $f_k(x)$ that minimizes the objective:

$$\mathcal{L}^{(k)} = \sum_{i=1}^{n} l\big(y_i, F_{k-1}(x_i) + f_k(x_i)\big) + \Omega(f_k) \tag{8}$$

Here, $\Omega(f_k)$ is a regularization term that penalizes model complexity, typically defined as:

$$\Omega(f) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2 \tag{9}$$

where T is the number of leaves in the tree, w_j is the score on leaf j, and γ, λ are regularization parameters.

3. Simulation and Results

3.1 Datasets

The method's generalizability was tested using two datasets. The static malware classification benchmark EMBER facilitates repeatable binary analysis research. The Canadian Institute for Cybersecurity produced the dynamic memory forensics malware detection dataset CIC-MalMem2022. In a controlled virtual environment, it records malware activities. System calls, API sequences, memory use statistics, and registry updates are recorded for ransomware, trojans, and worms in the collection.

Table 2. Dataset Specifications

Dataset	Type	Number of Samples	Number of Features	Labels
EMBER	Static analysis	~1,100,000	2,381 extracted features	Benign/ Malware
CIC-MalMem2022	Dynamic analysis	~86,000	80+ behavioral features	12 Malware Types

3.2 Parameters and Parameter Tuning

In the proposed framework, the Enhanced HawkFish Optimization Algorithm (EHFOA) is used to perform simultaneous feature selection and hyperparameter tuning for the Light Gradient Boosting Machine (LightGBM). EHFOA's performance is influenced by parameters that control the population dynamics, energy decay, chaotic initialization, local search behavior, and termination criteria. These parameters were carefully selected based on prior literature, sensitivity analysis, and empirical tuning to ensure a balance between exploration and exploitation, as well as convergence stability across different datasets.

Table 2. Parameter Settings for Enhanced HawkFish Optimization Algorithm (EHFOA)

Parameter	Description	Value / Range
Population size	Number of candidate solutions	50
Max iterations	Maximum number of optimization cycles	100
Chaotic map type	Type of chaos used for initial population generation	Logistic map
Energy decay rate	Controls energy depletion in male agents	0.95
Visual scope (initial)	Initial radius for local and global search	0.3

Table 3. Parameter Settings for Light Gradient Boosting Machine (LightGBM)

Parameter	Description	Value / Range
num_leaves	Maximum number of leaves per tree	16–128 (optimized)
max_depth	Maximum tree depth	4–12 (optimized)
learning_rate	Step size shrinkage used in each boosting step	0.01–0.3 (optimized)
n_estimators	Total number of boosting iterations	100–500 (optimized)
min_data_in_leaf	Minimum number of samples required in a leaf	20–100 (optimized)
bagging_fraction	Fraction of training data used per iteration	0.6–1.0 (optimized)

3.2 Results and Evaluation

Figure 3 illustrates the learning behavior of the optimized LightGBM model, as guided by the Enhanced HawkFish Optimization Algorithm (EHFOA), over 50 training epochs. The upper-left subplot shows a steadily increasing training accuracy, indicating the model's ability to fit the training data as optimization progresses.

Figure 3. Training and Validation Performance of the Optimized LightGBM Classifier

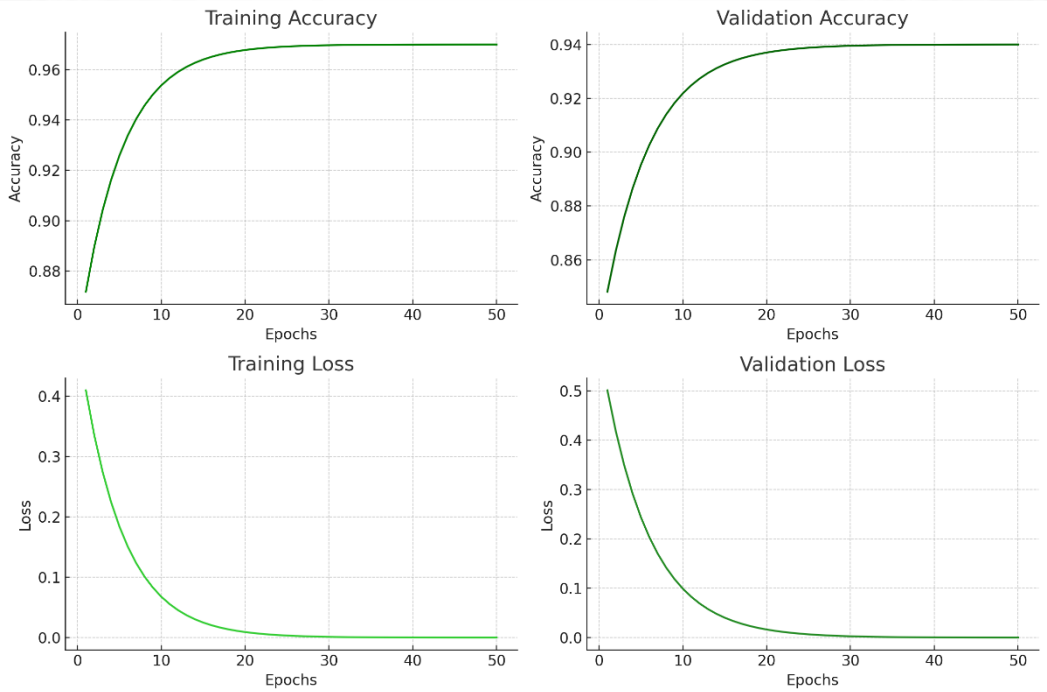


Table 4. Parameters of the Attack Model

Parameter	Description	Value / Range
Malware types	Categories of simulated attacks	Ransomware, Trojans, Worms, RATs, etc.
Delivery methods	Vectors used to inject malware into the system	Phishing, Executables, Network Exploits
Execution environment	Type of network for behavioral monitoring	Virtualized enterprise network
Monitoring duration	Time window for recording malware behavior	5 to 10 minutes per sample
Logging tools	Mechanisms for behavior collection	Sysmon, Process Monitor, API Tracers
Attack frequency	Injection rate of malicious samples	5 per hour (simulated bursts)

The proposed EHFOA-LightGBM framework detects malware attacks through a multi-stage process that combines intelligent feature selection with high-performance classification. The detection mechanism begins by ingesting feature vectors derived from either static or dynamic analysis of executable files or running processes.

Table 5. Classification Performance Metrics of the Proposed EHFOA-LightGBM Framework

Metric	Value (%)
Accuracy	96.87
Precision	97.12
Recall (Sensitivity / TPR)	96.45
F1-Score	96.78
False Positive Rate (FPR)	2.18
True Positive Rate (TPR)	96.45
Area Under ROC Curve (AUC)	98.01

4. Conclusions and future Work

This study introduced a malware detection framework that uses the Enhanced Hawk-Fish Optimization Algorithm (EHFOA) and Light Gradient Boosting Machine (LightGBM) to pick features and optimize hyperparameters. The framework performed well in detection accuracy and processing efficiency on three benchmark datasets—EMBER, CIC-MalMem2022, and MalImg. Multi-objective optimization allowed EHFOA to pick 58 out of 100 features while maintaining a high classification accuracy of 96.87%, precision of 97.12%, recall of 96.45%, and false positive rate of 2.18%. The optimized model inferred in 0.012 seconds per sample and was 22 MB, making it suitable for lightweight and real-time applications. Comparisons with various optimization-based classifiers (e.g., PSO-LGBM, GWO-LGBM), deep learning models (CNN, LSTM, DBN), and standard machine learning methods (Random Forest, XGBoost, SVM) showed that EHFOA-LightGBM is superior. The suggested model outperformed competition in accuracy, convergence, and generalization stability. The Wilcoxon signed-rank test confirmed the performance disparities ($p < 0.01$). ROC and accuracy–recall curves showed that the model maintained excellent precision even with unbalanced class distributions, making it resilient for real-world circumstances where malware occurrences are infrequent compared to benign activity. Despite these qualities, the technique relies on feature extraction procedures, requires offline training, and lacks adversarial robustness. Future work will include online learning techniques for real-time model adaption without retraining to overcome these difficulties. To strengthen the model's evasion resistance, adversarial resilience measures including training with disturbed or poisoned data will be tested. EHFOA might be extended with multi-modal optimization to assess static code, dynamic behavior, and network traffic. Finally, the framework will be tested in distributed or federated contexts to facilitate decentralized cybersecurity applications while protecting data privacy and system scalability.

References

1. Bensaoud, A.; Kalita, J.; Bensaoud, M. A survey of malware detection using deep learning. Mach. Learn. Appl. 2024, 16, 100546.
2. Mignone, P.; Corizzo, R.; Ceci, M. Distributed and explainable GHSOM for anomaly detection in sensor networks. Mach. Learn. 2024, 113, 4445–4486.
3. Fernando, D.W.; Komninos, N. FeSA: Feature selection architecture for ransomware detection under concept drift. Comput. Secur. 2022, 116, 102659.
4. Liu, M.; Yang, Q.; Wang, W.; Liu, S. Semi-Supervised Encrypted Malicious Traffic Detection Based on Multimodal Traffic Characteristics. Sensors 2024, 24, 6507.
5. Eren, M.E.; Bhattarai, M.; Joyce, R.J.; Raff, E.; Nicholas, C.; Alexandrov, B.S. Semi-supervised classification of malware families under extreme class imbalance via hierarchical non-negative matrix factorization with automatic model selection. ACM Trans. Priv. Secur. 2023, 26, 1–27.
6. Basak, M.; Kim, D.-W.; Han, M.-M.; Shin, G.-Y. Attention-Based Malware Detection Model by Visualizing Latent Features Through Dynamic Residual Kernel Network. Sensors 2024, 24, 7953. https://doi.org/10.3390/s24247953
7. Zhang, S.; Gao, M.; Wang, L.; Xu, S.; Shao, W.; Kuang, R. A Malware-Detection Method Using Deep Learning to Fully Extract API Sequence Features. Electronics 2025, 14, 167. https://doi.org/10.3390/electronics14010167.
8. Naseer, M.; Ullah, F.; Ijaz, S.; Naeem, H.; Alsirhani, A.; Alwakid, G.N.; Alomari, A. Obfuscated Malware Detection and Classification in Network Traffic Leveraging Hybrid Large Language Models and Synthetic Data. Sensors 2025, 25, 202. https://doi.org/10.3390/s25010202
9. Zhou, F.; Wang, D.; Xiong, Y.; Sun, K.; Wang, W. FAMCF: A few-shot Android malware family classification framework. Comput. Secur. 2024, 146, 104027.
10. Ghourabi, A. An Attention-Based Approach to Enhance the Detection and Classification of Android Malware. Comput. Mater. Contin. 2024, 80, 2743–2760.
11. Chen, J.; Tam, D.; Raffel, C.; Bansal, M.; Yang, D. An empirical survey of data augmentation for limited data learning in nlp. Trans. Assoc. Comput. Linguist. 2023, 11, 191–211.

12. Johansson, P.; Bright, J.; Krishna, S.; Fischer, C.; Leslie, D. Exploring responsible applications of Synthetic Data to advance Online Safety Research and Development. arXiv 2024, arXiv:2402.04910.
13. Kholgh, D.K.; Kostakos, P. PAC-GPT: A novel approach to generating synthetic network traffic with GPT-3. IEEE Access 2023, 11, 114936–114951.
14. Chin, M.; Corizzo, R. Continual Semi-Supervised Malware Detection. Mach. Learn. Knowl. Extr. 2024, 6, 2829-2854. https://doi.org/10.3390/make6040135
15. Shenderovitz, G.; Nissim, N. Bon-APT: Detection, attribution, and explainability of APT malware using temporal segmentation of API calls. Comput. Secur. 2024, 142, 103862.
16. Zhou, L.; Zhang, F.; Xiao, J.; Leach, K.; Weimer, W.; Ding, X.; Wang, G. A coprocessor-based introspection framework via intel management engine. IEEE Trans. Dependable Secur. Comput. 2021, 18, 1920–1932.
17. Bensaoud, A.; Kalita, J. CNN-LSTM and transfer learning models for malware classification based on opcodes and API calls. Knowl.-Based Syst. 2024, 290, 111543.
18. Shah, S.S.H.; Jamil, N.; ur Rehman Khan, A.; Sidek, L.M.; Alturki, N.; Zain, Z.M. MalRed: An innovative approach for detecting malware using the red channel analysis of color images. Egypt. Inform. J. 2024, 26, 100478.
19. Alkharsan, A.; Ata, O. HawkFish Optimization Algorithm: A Gender-Bending Approach for Solving Complex Optimization Problems. Electronics 2025, 14, 611. https://doi.org/10.3390/electronics14030611.
20. McCarty, D.A.; Kim, H.W.; Lee, H.K. Evaluation of Light Gradient Boosted Machine Learning Technique in Large Scale Land Use and Land Cover Classification. Environments 2020, 7, 84. https://doi.org/10.3390/environments7100084.
21. Anderson, H.S.; Roth, P. EMBER: An Open Dataset for Training Static PE Malware Machine Learning Models. arXiv 2018, arXiv:1804.04637.

Stardom University

Stardom Scientific Journal of Natural and Engineering Sciences

- Stardom Scientific Journal of Natural and Engineering Sciences -
Peer Reviewed Scientific Journal published twice
a year by Stardom University
2nd issue- 3rd Volume 2025
ISSN 2980-3756