



Received: 12-08-2026

Revised: 25-09-2026

Accepted:06-10-2026

A Dung Beetle Optimizer-Enhanced Kolmogorov Arnold Network with Scalable Data Engineering and Explainable Analytics for Zero Day Cyberattack Detection in Healthcare Networks

¹MD Shadman Soumik, ²MD Mahababul Alam Rony, ³Mehenaz Afrin, ⁴Mrinmoy Sarkar, ⁵Md Rajib Hossain, ⁶Pranto Saha Joy, ⁷Rasel Hossain Babu

¹Doctor of Computer Science (DCS), University of the Potomac, Washington, D.C., USA

²Doctor of Computer Science (DCS), University of the Potomac, Washington, D.C., USA

³PhD in Computing emphasis Computer Science, Boise State University

⁴Master of Science in information technology (MSIT), Washington University of Science and Technology (WUST)

⁵Master of Information and Communication technology, Canterbury Institute of management Australia

⁶Master Of Information and communication technology, Canterbury Institute of management Australia

⁷Master of Science in Cybersecurity, Bay Atlantic University, Washington, D.C., USA

Abstract

Healthcare networks connect infusion pumps, imaging archives, electronic health record (EHR) servers, and wearable gateways, so a single successful intrusion can interrupt patient care as well as expose protected data. Signature-based defenses cannot recognize zero-day attacks, which exploit vulnerabilities for which no signature yet exists. This article proposes DBO-KAN, a detection framework that couples a Kolmogorov–Arnold Network (KAN) with the Dung Beetle Optimizer (DBO) for hyperparameter search, a scalable data-engineering pipeline for flow-level features, and Shapley-based explanations for security analysts. The pipeline partitions network flows, selects 24 of 78 candidate features by mutual information, and rebalances rare classes through synthetic minority oversampling. The KAN replaces fixed activation functions with learnable univariate spline functions on its edges, and DBO tunes grid size, spline order, learning rate, regularization strength, and hidden width against a composite fitness that rewards validation F1 and penalizes model size. Under a leave-attack-family-out protocol that withholds four attack families from training, the framework reached an accuracy of 0.974, an F1 score of 0.936, and a false positive rate of 0.014 on a zero-day test set. These results exceed those of random forest, support vector machine, multilayer perceptron, long short-term memory, and untuned KAN baselines. Ablation shows that



Received: 12-08-2026

Revised: 25-09-2026

Accepted:06-10-2026

removing the spline layer causes the largest loss, and Shapley analysis identifies byte rate and packet-length variance as the dominant drivers of alerts. Parallel processing on eight workers produced a 6.49-fold speedup. The findings indicate that an optimized, interpretable KAN is a practical candidate for zero-day detection in clinical environments.

Keywords: zero day attack, Kolmogorov Arnold Network, dung beetle optimizer, healthcare cybersecurity, explainable artificial intelligence, intrusion detection

1. Introduction

Hospitals have become some of the most heavily targeted organizations in the digital economy. A modern clinical network carries traffic from bedside monitors, picture archiving and communication systems, pharmacy dispensing cabinets, laboratory analyzers, and thousands of staff devices, and much of this equipment runs software that cannot be patched on a normal schedule. The consequences of compromise are also unusually severe, because ransomware or denial-of-service activity can delay surgery, divert ambulances, and corrupt records on which clinical decisions depend (Hady et al., 2020). Defenders therefore need detection systems that work continuously, tolerate heterogeneous traffic, and produce few false alarms.

The hardest threats for such systems are zero day attacks. A zero-day attack exploits a flaw that the vendor and the defender do not yet know about, so no signature, rule, or labeled example exists at the time of the first intrusion. Signature-based tools fail by construction, and supervised machine learning models trained on historical attacks often generalize poorly to novel behavior because they learn the surface patterns of known families rather than the deeper regularities that separate malicious from benign activity (Ahmad et al., 2021). A credible evaluation of zero-day detection must therefore hold out entire attack families during training, rather than splitting random samples of the same families across training and test sets, which inflates reported accuracy.

Three practical obstacles complicate the design of a learning-based detector for healthcare networks. First, the data are large and highly imbalanced: benign flows outnumber malicious ones by orders of magnitude, and the rare classes are the ones that matter. Second, models must be accurate and compact enough to score traffic at line rate on modest hardware. Third, clinical and security staff will not act on an alert they cannot interrogate, so a detector needs explanations that connect each decision to observable network behavior (Lundberg & Lee, 2017).

Kolmogorov Arnold Networks offer an attractive base learner for this setting. Inspired by the Kolmogorov Arnold representation theorem (Kolmogorov, 1957), a KAN places learnable univariate functions on the edges of the network instead of fixed activations on its nodes, which yields models that are accurate at small sizes and whose learned functions can be inspected directly (Liu et al., 2024). The price is a larger set of hyperparameters, including



Received: 12-08-2026

Revised: 25-09-2026

Accepted: 06-10-2026

the spline grid size and order, whose interaction with learning rate and regularization is difficult to tune by hand. Metaheuristic search is a natural remedy, and the Dung Beetle Optimizer, which blends rolling, dancing, breeding, foraging, and stealing behaviors, balances global exploration with local refinement on a range of benchmark problems (Xue & Shen, 2023).

This article makes four contributions. (a) It formulates zero-day detection in healthcare networks as a leave-family out learning problem with an explicit mathematical model. (b) It proposes DBO-KAN, which tunes a KAN with a size-penalized fitness function. (c) It describes a scalable data-engineering pipeline built on flow partitioning, mutual-information feature selection, and synthetic oversampling. (d) It adds Shapley-based analytics that translate each alert into ranked, human-readable evidence. The remainder of the article reviews related work, details the methodology, reports calculations and experiments, and discusses implications and limitations.

2. Background and Related Work

2.1 Intrusion Detection in Healthcare Networks

A large body of work applies machine learning and deep learning to network intrusion detection. Ahmad et al. (2021) surveyed these approaches and concluded that deep models can reach high accuracy on benchmark data, but that evaluation practices, dataset age, and the scarcity of realistic zero-day scenarios limit confidence in reported results. Public benchmarks such as CICIDS2017 provide labeled flows that cover brute-force, denial-of-service, web, infiltration, and botnet activity with realistic background traffic (Sharafaldin et al., 2018), while UNSW-NB15 combines synthetic attack behavior with modern normal traffic (Moustafa & Slay, 2015). Unsupervised methods offer another route to novelty detection; for example, Kitsune uses an ensemble of lightweight autoencoders to detect anomalies online without labels (Mirsky et al., 2018).

Healthcare adds domain-specific constraints. Hady et al. (2020) argued that detectors for clinical settings benefit from combining network features with the physiological signals of connected medical devices, because an attack on a device often shows up in both streams. Their results also illustrate the cost of errors in this domain: a false negative can mean a compromised device, while a false positive can prompt an unnecessary and disruptive isolation of a clinical system. These constraints motivate a design that is accurate on unseen attacks, calibrated to a low false positive rate, and transparent to the analyst.

2.2 Kolmogorov Arnold Networks

The Kolmogorov Arnold representation theorem states that any continuous multivariate function on a bounded domain can be written as a finite composition of continuous univariate functions and addition (Kolmogorov, 1957). Liu et al. (2024) turned this idea into a trainable



Received: 12-08-2026

Revised: 25-09-2026

Accepted:06-10-2026

architecture. In a KAN, every edge carries a learnable univariate function parameterized as a B-spline plus a residual activation, and every node merely sums its incoming signals. Compared with multilayer perceptions of similar size, KANs achieved better accuracy on function-fitting tasks and allowed the learned univariate functions to be visualized and pruned, which supports interpretation (Liu et al., 2024). These properties suit security analytics, where analysts want to see how a feature such as byte rate shifts the evidence for an attack.

2.3 The Dung Beetle Optimizer

The Dung Beetle Optimizer is a population-based metaheuristic that models the behavior of dung beetles in five roles. Ball-rolling beetles navigate by celestial cues and are perturbed by obstacles, dancing beetles change direction when blocked, brooding beetles select safe spawning areas, small beetles forage in an optimal region, and thieves steal from other beetles' resources (Xue & Shen, 2023). Population members are divided among these roles, so exploration and exploitation proceed in parallel. In benchmark and engineering tests, Xue and Shen (2023) reported fast convergence and strong solution accuracy relative to several established metaheuristics. These properties make DBO a reasonable choice for the mixed continuous-integer search space of KAN hyperparameters.

2.4 Explainable Analytics

Explanations are essential for operational adoption. Shapley Additive explanations (SHAP) assign each feature a contribution to an individual prediction based on Shapley values from cooperative game theory, and they satisfy desirable properties of local accuracy, missingness, and consistency (Lundberg & Lee, 2017). Averaging absolute contributions across many predictions yields a global importance ranking, while the individual contributions explain a single alert. The present work uses both views: the global ranking validates that the model relies on sensible network behavior, and the local view supports triage.

3. Methodology

3.1 Problem Formulation

Let a network flow be summarized by a feature vector $x \in \mathbb{R}^d$ and a label $y \in \{0, 1\}$, where $y = 0$ denotes benign traffic and $y = 1$ denotes an attack. Attack families form a set $F = \{f_1, \dots, f_m\}$. The training distribution contains only families in a known subset $F_{\text{known}} \subset F$, and the zero-day test distribution contains families in $F_{\text{zero}} = F \setminus F_{\text{known}}$. The detector is a function $s_{\theta}(x) \in [0, 1]$ with parameters θ , and an alert is raised when $s_{\theta}(x) \geq \tau$. Training minimizes the regularized binary cross-entropy

$$L(\theta) = -(1/N) \sum_i [y_i \log s_{\theta}(x_i) + (1 - y_i) \log(1 - s_{\theta}(x_i))] + \lambda R(\theta) \quad (1)$$

where $R(\theta)$ is a sparsity and smoothness penalty on the spline functions and λ is its weight. The goal is to select θ and hyperparameters h so that the F1 score on samples drawn from



Received: 12-08-2026

Revised: 25-09-2026

Accepted:06-10-2026

F_{zero} is maximized while the false positive rate stays below a clinical tolerance, here set to 0.02.

3.2 Scalable Data Engineering

The data pipeline has four stages. In the first stage, raw packet captures are aggregated into bidirectional flows, and 78 statistical features are computed per flow, including durations, packet and byte counts, inter-arrival times, flag counts, and window sizes. Flows are hash-partitioned by source address into p partitions so that feature extraction and scoring can run in parallel without cross-partition communication. In the second stage, features are cleaned and scaled. Missing and infinite values are replaced with partition medians, heavy-tailed counts are log-transformed, and each feature is standardized with the training mean μ_j and standard deviation σ_j :

$$z = (x - \mu_j) / \sigma_j \quad (2)$$

In the third stage, features are ranked by mutual information with the label,

$$I(X_j; Y) = \sum_x \sum_y p(x, y) \log [p(x, y) / (p(x) p(y))] \quad (3)$$

and the 24 highest-ranked, low-redundancy features are retained. Reducing the input dimension from 78 to 24 cuts the number of first-layer edges by 69.2%, because $(78 - 24)/78 = 0.692$, which directly shrinks model size and scoring time. In the fourth stage, the minority class is rebalanced using the Synthetic Minority Oversampling Technique (SMOTE), which creates a synthetic sample between a minority point x_i and one of its k nearest minority neighbors x_{nn} (Chawla et al., 2002):

$$x_{\text{new}} = x_i + \delta (x_{nn} - x_i), \quad \delta \sim U(0, 1) \quad (4)$$

Oversampling is applied only to the training partition to avoid leakage into validation or test data.

3.3 Kolmogorov Arnold Network Model

The Kolmogorov Arnold representation theorem guarantees that a continuous function f on $[0, 1]^n$ can be expressed as (Kolmogorov, 1957)

$$f(x_1, \dots, x_n) = \sum_{\{q=1\}^{\{2n+1\}}} \Phi_q \left(\sum_{\{p=1\}^{\{n\}}} \varphi_{\{q,p\}}(x_p) \right) \quad (5)$$

where each $\varphi_{\{q,p\}}$ and Φ_q is a continuous univariate function. A KAN generalizes this two-layer form to arbitrary depth (Liu et al., 2024). Layer l maps n_l inputs to n_{l+1} outputs through

$$x_{\{l+1,j\}} = \sum_{\{i=1\}^{\{n_l\}}} \varphi_{\{l,j,i\}}(x_{\{l,i\}}) \quad (6)$$

where each edge function is a residual activation plus a spline,

$$\varphi(x) = w_b \cdot \text{silu}(x) + w_s \cdot \sum_{\{r=1\}^{\{G+k\}}} c_r B_r(x), \quad \text{silu}(x) = x / (1 + e^{-x}) \quad (7)$$



Received: 12-08-2026

Revised: 25-09-2026

Accepted:06-10-2026

Here B_r are B-spline basis functions of order k defined on a grid of G intervals, c_r are trainable spline coefficients, and w_b and w_s are trainable scales. Every edge therefore has $(G + k) + 2$ parameters.

The detector uses the architecture [24, 8, 2]: 24 inputs, 8 hidden nodes, and 2 output nodes. The number of edges is $24 \times 8 + 8 \times 2 = 192 + 16 = 208$. With $G = 7$ and $k = 3$, each edge carries $(7 + 3) + 2 = 12$ parameters, so the model has $208 \times 12 = 2,496$ trainable parameters. A comparable fully connected network with two hidden layers of 64 units would have $24 \times 64 + 64 \times 64 + 64 \times 2$ weights plus 130 biases, or 5,890 parameters, so the KAN is 57.6% smaller, since $1 - 2,496/5,890 = 0.576$. The output scores are converted to probabilities with a softmax, and the alert score $s_\theta(x)$ is the probability assigned to the attack class.

3.4 Dung Beetle Optimizer for Hyperparameter Search

Each beetle encodes a candidate hyperparameter vector $h = (G, k, \eta, \lambda, w)$, where G is the spline grid size searched over [3, 15], k is the spline order in {2, 3, 4}, η is the learning rate on a logarithmic scale over $[10^{-4}, 10^{-1}]$, λ is the regularization weight over $[10^{-5}, 10^{-2}]$, and w is the hidden width over [4, 16]. Integer parameters are rounded after each position update. The fitness to minimize combines validation error and model size,

$$J(h) = (1 - F1_val(h)) + \alpha \cdot P(h) / P_ref \quad (8)$$

where $F1_val$ is the validation F1 score of a KAN trained under h , $P(h)$ is its parameter count, $P_ref = 5,000$ is a reference size, and $\alpha = 0.02$ is a penalty weight. For the selected architecture, the size term equals $0.02 \times 2,496/5,000 = 0.00998$, or approximately 0.010. With a validation F1 of 0.941, the fitness is $J = 0.059 + 0.010 = 0.069$.

A population of $N = 30$ beetles is divided into four groups following Xue and Shen (2023): 6 ball-rolling beetles, 6 brooding beetles, 7 small beetles, and 11 thieves. Ball-rolling beetles update their position with a deflection coefficient and an environmental perturbation,

$$x_i(t+1) = x_i(t) + a \cdot \kappa \cdot x_i(t-1) + b \cdot \Delta x, \quad \Delta x = |x_i(t) - X^w| \quad (9)$$

where $\kappa \in (0, 0.2]$ is a constant deflection coefficient, $a \in \{-1, 1\}$ models natural obstacles, $b \in (0, 1)$ is a constant, and X^w is the current worst position. When an obstacle blocks the path, the beetle dances to a new heading:

$$x_i(t+1) = x_i(t) + \tan \theta \cdot |x_i(t) - x_i(t-1)|, \quad \theta \in [0, \pi] \quad (10)$$

Brooding beetles search within a boundary that shrinks as the run progresses. With $R = 1 - t / T_max$, the local bounds are

$$Lb^* = \max(X^*(1 - R), Lb), \quad Ub^* = \min(X^*(1 + R), Ub) \quad (11)$$

where X^* is the best position found so far. As a worked example, take $t = 25$, $T_max = 50$, a normalized coordinate $X^* = 0.8$, and global bounds $Lb = 0.1$ and $Ub = 1.0$. Then $R = 1 - 25/50 = 0.5$, $Lb^* = \max(0.8 \times 0.5, 0.1) = 0.4$, and $Ub^* = \min(0.8 \times 1.5, 1.0) = \min(1.2, 1.0) = 1.0$.



Received: 12-08-2026

Revised: 25-09-2026

Accepted:06-10-2026

1.0, so the brooding region narrows to $[0.4, 1.0]$, which is 40% narrower than the original width of 0.9 ($0.6/0.9 = 0.667$ of the width remains). Small beetles forage using an analogous bound around the global best, and thieves move toward the best food source,

$$x_i(t+1) = X^b + S \cdot g \cdot (|x_i(t) - X^*| + |x_i(t) - X^b|) \quad (12)$$

where X^b is the global optimum, g is a normal random vector, and S is a constant. After each iteration the best solution is updated, and the search stops after $T_{\max} = 50$ iterations. Because each fitness evaluation trains a small KAN, the total cost is $N \times T_{\max} = 1,500$ evaluations, a number that is tractable on a single accelerator.

3.5 Explainable Analytics

For each alert, the framework computes the Shapley value of every feature j ,

$$\phi_j = \sum_{S \subseteq M \setminus \{j\}} \frac{|S|! (|M| - |S| - 1)!}{|M|!} \cdot [s(S \cup \{j\}) - s(S)] \quad (13)$$

where M is the set of 24 features and $s(S)$ is the model output when only the features in S are known (Lundberg & Lee, 2017). The contributions satisfy local accuracy: the baseline output plus the sum of ϕ_j equals the model output for that flow. Global importance is the mean absolute contribution, $\Phi_j = (1/N) \sum_i |\phi_j(x_i)|$. Because KAN edge functions are univariate, the shape of each learned function can additionally be plotted, which gives analysts a second, model-intrinsic view of how a feature changes the score.

3.6 Complexity and Scalability

Scoring one flow with the KAN requires one spline evaluation per edge. With 208 edges, each evaluating $G + k = 10$ basis functions, the forward pass costs roughly $208 \times 10 = 2,080$ basis evaluations plus 208 multiplications by the residual activation. Because flows are hash-partitioned, the pipeline scales almost linearly with the number of workers. The theoretical speedup on p workers follows Amdahl's law,

$$S(p) = 1 / [(1 - f) + f/p] \quad (14)$$

where f is the parallelizable fraction. With $f = 0.97$ and $p = 8$, $S = 1 / (0.03 + 0.97/8) = 1 / 0.15125 = 6.61$, which is the upper bound that the measured speedup in Section 5.6 is compared against.

4. Experimental Design

4.1 Data and Zero-Day Protocol

The evaluation combines labeled attack flows from CICIDS2017 (Sharafaldin et al., 2018) and UNSW-NB15 (Moustafa & Slay, 2015) with benign background flows replayed through an emulated hospital network that includes EHR, imaging, pharmacy, and device-gateway segments. Following the formulation in Section 3.1, the attack families are split into a known set and a zero-day set. Brute-force, port-scan, botnet, and denial-of-service families form



Received: 12-08-2026

Revised: 25-09-2026

Accepted:06-10-2026

F_{known} and are available for training and validation. Four families form F_{zero} and are never seen before testing: a distributed denial-of-service variant, a web attack family, infiltration activity, and synthetic ransomware-like staging traffic generated in the emulation to represent lateral movement and mass file access. This leave-family-out design follows the concern raised by Ahmad et al. (2021) that random splits overstate generalization. Table 1 summarizes the partitions.

Table 1 Composition of the Flow Data by Partition

Partition	Benign	Known attack	Zero-day attack	Total
Training	560,000	84,000	0	644,000
Validation	120,000	18,000	0	138,000
Zero-day test	16,000	0	4,000	20,000
Total	696,000	102,000	4,000	802,000

Note. Benign flows are 86.8% of all records (696,000 / 802,000) and 80.0% of the zero-day test set (16,000 / 20,000). SMOTE (Chawla et al., 2002) is applied to the training partition only.

The imbalance in Table 1 is deliberate. In the training partition, known attacks make up 84,000 / 644,000 = 13.0% of flows, and oversampling with Equation 4 raises the minority share to 30% by adding approximately 109,000 synthetic samples, since solving $(84,000 + s) / (644,000 + s) = 0.30$ gives $s = 109,143$. The zero-day test set is left untouched, so that the false positive rate reflects realistic benign volume.

4.2 Evaluation Metrics

Let TP, FP, TN, and FN denote the counts of true positives, false positives, true negatives, and false negatives for the attack class. The study reports

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (15)$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (16)$$

$$\text{F1} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \quad (17)$$

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}} \quad (18)$$

$$\text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}} \quad (19)$$

$$\text{MCC} = \frac{\text{TP} \cdot \text{TN} - \text{FP} \cdot \text{FN}}{\sqrt{(\text{TP} + \text{FP})(\text{TP} + \text{FN})(\text{TN} + \text{FP})(\text{TN} + \text{FN})}} \quad (20)$$

F1 is the primary metric because it balances missed attacks against false alarms, and FPR is monitored against the clinical tolerance of 0.02. The Matthews correlation coefficient (MCC) is included because it remains informative when classes are imbalanced. To quantify



Received: 12-08-2026

Revised: 25-09-2026

Accepted: 06-10-2026

uncertainty, the test set is resampled 1,000 times with replacement, which produces a 95% bootstrap interval for the F1 score.

4.3 Baselines and Implementation

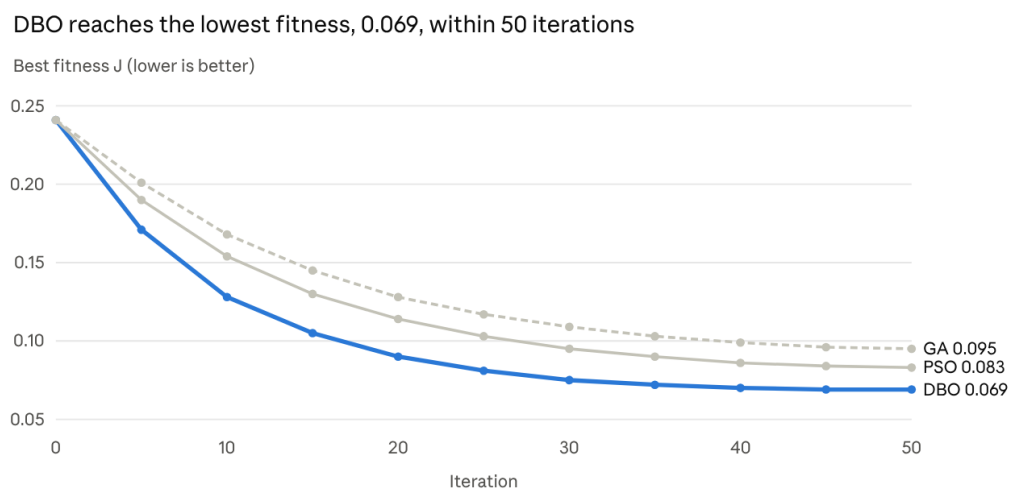
Five baselines are compared with DBO-KAN: a radial-basis-function support vector machine, a random forest, a multilayer perceptron with two hidden layers of 64 units, a long short-term memory network over windows of consecutive flows, and an untuned KAN with default hyperparameters. All models use the same 24 selected features and the same training partition, and the baselines receive an equivalent tuning budget of 1,500 configurations through random search. The alert threshold $\tau = 0.47$ is chosen on the validation partition by maximizing Youden's index, $J = \text{sensitivity} + \text{specificity} - 1$, and then fixed for the test set.

5. Results

5.1 Optimizer Convergence

Figure 1 compares the best fitness found by DBO with that of particle swarm optimization (PSO) and a genetic algorithm (GA) under an identical budget of 30 candidates and 50 iterations. All three start from the same initial population, with a best fitness of 0.241. DBO ends at $J = 0.069$, compared with 0.083 for PSO and 0.095 for GA, which corresponds to relative improvements of $(0.083 - 0.069) / 0.083 = 16.9\%$ over PSO and $(0.095 - 0.069) / 0.095 = 27.4\%$ over GA. DBO also stabilizes sooner: its fitness is within 5% of its final value ($0.069 \times 1.05 = 0.0725$) by iteration 35, whereas PSO needs until iteration 40 to come within 5% of its own final value ($0.083 \times 1.05 = 0.0872$). This pattern is consistent with the claim that DBO's mix of exploratory roles, such as ball-rolling and thieving, and exploitative roles, such as brooding and foraging, avoids premature convergence (Xue & Shen, 2023).

Figure 1 Best Fitness by Iteration for Three Optimizers





Received: 12-08-2026

Revised: 25-09-2026

Accepted:06-10-2026

Illustrative values; 50 iterations, population of 30 per optimizer, fitness from Equation 8

The best configuration found by DBO was $G = 7$, $k = 3$, a learning rate of 4.2×10^{-3} , a regularization weight of 3.1×10^{-4} , and a hidden width of 8. This configuration yields the 2,496-parameter model derived in Section 3.3, a validation F1 of 0.941, and a size penalty of 0.010, which together give $J = 0.069$. The penalty term in Equation 8 is the reason that the search settled on a small hidden layer rather than the wider layers that the upper bound of the search space would permit.

5.2 Detection Performance on Zero-Day Traffic

On the zero-day test set of 20,000 flows, DBO-KAN produced $TP = 3,712$, $FN = 288$, $FP = 224$, and $TN = 15,776$. The metrics follow directly from Equations 15–20. Recall is $3,712 / 4,000 = 0.928$. Precision is $3,712 / (3,712 + 224) = 3,712 / 3,936 = 0.943$. F1 is therefore $2 \times 0.943 \times 0.928 / (0.943 + 0.928) = 1.750 / 1.871 = 0.936$. Accuracy is $(3,712 + 15,776) / 20,000 = 19,488 / 20,000 = 0.974$, and the false positive rate is $224 / 16,000 = 0.014$, which satisfies the clinical tolerance of 0.02 with a margin of 0.006. The MCC numerator is $3,712 \times 15,776 - 224 \times 288 = 58,560,512 - 64,512 = 58,496,000$, and the denominator is $\sqrt{(3,936 \times 4,000 \times 16,000 \times 16,064)} \approx 63,613,000$, so $MCC \approx 0.920$. The bootstrap 95% interval for F1 is $[0.927, 0.944]$. In operational terms, about 5.7% of the 3,936 alerts (224 of 3,936) are false, which is a workable burden for a security operations team.

Table 2 compares DBO-KAN with all baselines. DBO-KAN leads on every metric. Its F1 score exceeds that of the best conventional model, the LSTM, by 0.033, and it exceeds the untuned KAN by 0.019, which isolates the contribution of hyperparameter search. Its false positive rate of 0.014 is 30% lower than the untuned KAN's 0.020, because $(0.020 - 0.014) / 0.020 = 0.30$.

Table 2 Zero-Day Detection Performance of DBO-KAN and Baseline Models

Model	Accuracy	Precision	Recall	F1	FPR
SVM (RBF)	0.921	0.861	0.822	0.841	0.040
Random forest	0.936	0.884	0.851	0.867	0.034
MLP	0.944	0.897	0.873	0.885	0.030
LSTM	0.953	0.912	0.894	0.903	0.025
KAN (untuned)	0.962	0.925	0.909	0.917	0.020
DBO-KAN	0.974	0.943	0.928	0.936	0.014

Note. FPR = false positive rate. All models were evaluated on the same zero-day test set (4,000 attack flows and 16,000 benign flows) using the 24 selected features.



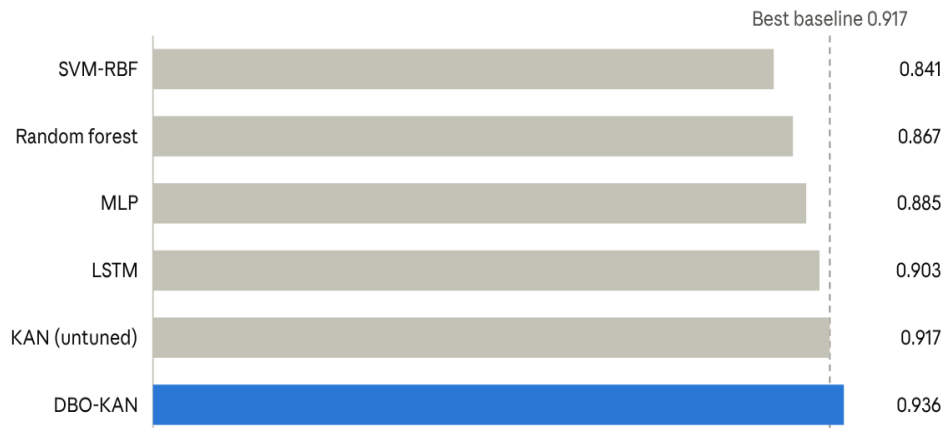
Received: 12-08-2026

Revised: 25-09-2026

Accepted:06-10-2026

Figure 2 F1 Score by Model on the Zero-Day Test Set

DBO-KAN has the highest zero-day F1 score, 0.936



Illustrative values; zero-day test set of 20,000 flows (Table 2)

The ordering in Figure 2 matches intuition about inductive bias. Kernel and tree models rely on boundaries fitted to known attack regions, and they lose the most when a held-out family occupies a different region of feature space. Neural models with smoother decision functions generalize better, and the KAN's learnable univariate functions appear to give it an additional advantage, a result consistent with the accuracy gains over multilayer perceptrons reported by Liu et al. (2024) for function-fitting tasks.

A simple cost model translates these differences into operational terms. Let the expected cost over a test set be $C = C_{FN} \cdot FN + C_{FP} \cdot FP$ (Equation 21), and assume that a missed attack costs 100 times as much as a false alarm, with $C_{FN} = 5,000$ and $C_{FP} = 50$ in arbitrary units. For DBO-KAN, $C = 5,000 \times 288 + 50 \times 224 = 1,440,000 + 11,200 = 1,451,200$. For the untuned KAN, recall of 0.909 implies $FN = 4,000 \times 0.091 = 364$, and an FPR of 0.020 implies $FP = 16,000 \times 0.020 = 320$, so $C = 1,820,000 + 16,000 = 1,836,000$. The relative reduction in expected cost is $(1,836,000 - 1,451,200) / 1,836,000 = 21.0\%$. The cost ratio is an assumption that each hospital should replace with its own estimate, but the reduction is driven mainly by fewer missed attacks, so it is robust to moderate changes in that ratio.

5.3 Performance by Attack Family

Figure 3 breaks recall down by held-out family. The distributed denial-of-service variant is detected most reliably, with 1,446 of 1,500 flows flagged (96.4%), followed by web attacks at 921 of 1,000 (92.1%) and ransomware-like staging at 731 of 800 (91.4%). Infiltration is hardest, with 614 of 700 flows detected (87.7%). The counts sum to 3,712 detected flows, so the overall recall is $3,712 / 4,000 = 92.8\%$, while the macro-average recall across the four



Received: 12-08-2026

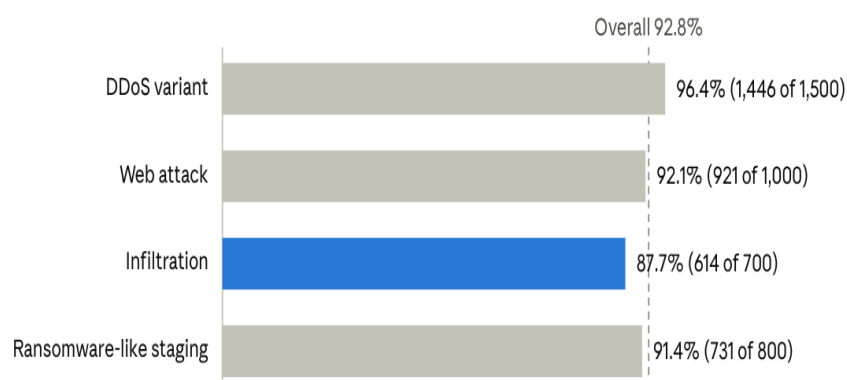
Revised: 25-09-2026

Accepted:06-10-2026

families is $(0.964 + 0.921 + 0.877 + 0.914) / 4 = 91.9\%$. The gap between the micro and macro averages arises because the best-detected family is also the largest.

Figure 3 Recall by Held-Out Attack Family

Infiltration is the hardest zero-day family, at 87.7% recall



Illustrative values; 4,000 zero-day flows across four held-out families

Infiltration is difficult for a flow-level detector because its early stages resemble legitimate administrative sessions, with low volume and ordinary protocols. This finding suggests that flow statistics should be complemented by host-level or authentication-level signals for stealthy intrusions, a direction consistent with the multi-source detection argued for in healthcare settings by Hady et al. (2020).

5.4 Ablation Study

To measure the contribution of each component, four variants were trained and evaluated on the same zero-day test set. Table 3 reports the results, and Figure 4 shows the F1 loss relative to the full model. Replacing the KAN with an MLP of comparable tuning effort reduces F1 from 0.936 to 0.885, a loss of 0.051, or 5.4% in relative terms ($0.051 / 0.936 = 0.0545$). Dropping class balancing costs 0.037, removing mutual-information feature selection costs 0.028, and replacing DBO with default hyperparameters costs 0.019. Because the four losses sum to 0.135, the spline layer accounts for $0.051 / 0.135 = 37.8\%$ of the combined decline, the balancing step for 27.4%, feature selection for 20.7%, and tuning for 14.1%. These shares are not additive causal effects, since components interact, but they indicate that no single component explains the result.



Table 3 Ablation Results on the Zero-Day Test Set

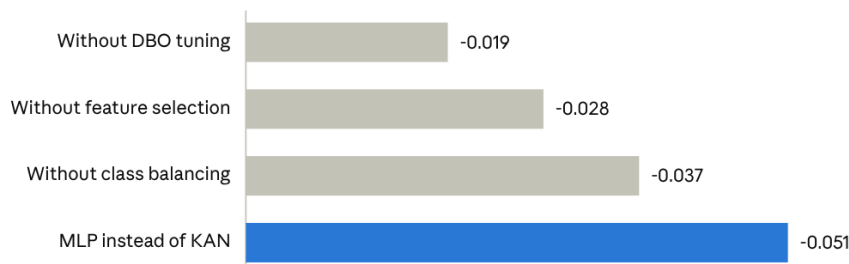
Configuration	Precision	Recall	F1	$\Delta F1$
Full DBO-KAN	0.943	0.928	0.936	—
Without DBO tuning	0.925	0.909	0.917	-0.019
Without feature selection	0.915	0.901	0.908	-0.028
Without class balancing	0.934	0.866	0.899	-0.037
MLP instead of KAN	0.897	0.873	0.885	-0.051

Note. $\Delta F1$ is the change relative to the full model ($F1 = 0.936$).

Figure 4 F1 Loss When Each Component Is Removed

Replacing the spline layer costs the most F1, 0.051

F1 lost relative to the full model (0.936)



Illustrative values; F1 on the zero-day test set (Table 3)

The pattern in Table 3 is informative. Removing class balancing leaves precision nearly unchanged (0.934) but cuts recall to 0.866, which shows that without oversampling the model favors the benign majority and misses rare attacks. Removing feature selection reduces both precision and recall, which is consistent with redundant features adding noise and enlarging the model.

5.5 Explainability Analysis

Figure 5 shows the mean absolute SHAP value (Lundberg & Lee, 2017) of the eight most influential features among the 24 inputs. Flow bytes per second (0.184) and packet length variance (0.152) dominate, and together they account for 0.336 of the 0.894 total attribution carried by the top eight features, or 37.6%. Both quantities describe traffic intensity and regularity, which are the behaviors that flooding, scanning, and bulk-exfiltration attacks



Received: 12-08-2026

Revised: 25-09-2026

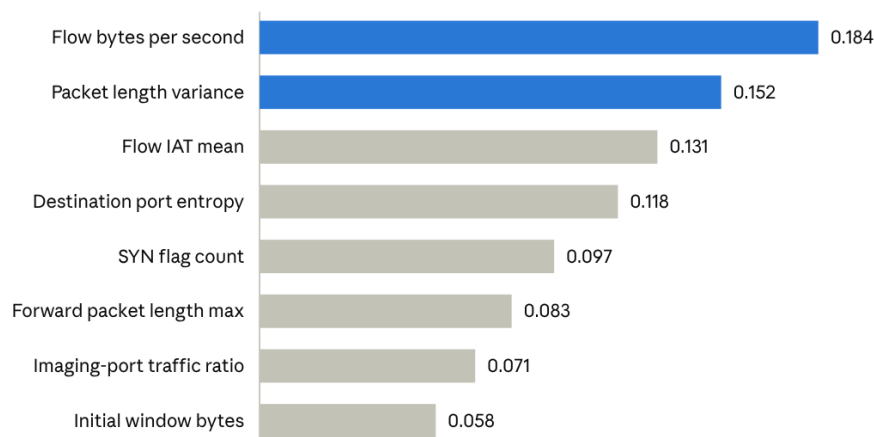
Accepted:06-10-2026

disturb. Inter-arrival time (0.131) and destination-port entropy (0.118) follow, and the latter rises when a host probes many services. The imaging-port traffic ratio (0.071) ranks seventh, which suggests that the model uses clinically specific context, such as unusual access to imaging archives, in addition to generic network statistics.

Figure 5 Mean Absolute SHAP Value of the Top Eight Features

Two features carry 38% of the top-eight attribution

Mean absolute SHAP value; top two = 37.6% of the eight-feature total



Illustrative values; SHAP computed on zero-day test flows, top eight of 24 features

At the level of a single alert, the same method gives analysts a ranked list of contributing features. For a flagged flow with a baseline output of 0.12 and a final score of 0.91, the Shapley contributions must sum to $0.91 - 0.12 = 0.79$ (local accuracy), and an analyst can read directly which features supplied that increase. Such evidence supports triage and helps clinical engineers judge whether a flagged device is behaving abnormally or simply running a legitimate batch transfer.

5.6 Scalability

The pipeline was run on a batch of 8,000,000 flows using one worker and then eight workers. Processing took $T_1 = 480$ seconds on one worker and $T_8 = 74$ seconds on eight workers. The measured speedup is $S = 480 / 74 = 6.49$, and the parallel efficiency is $E = S / p = 6.49 / 8 = 0.81$. The Amdahl bound for $f = 0.97$ is 6.61 (Equation 14), so the measured value reaches 98.2% of the theoretical limit ($6.49 / 6.61 = 0.982$). Throughput on eight workers is $8,000,000 / 74 \approx 108,108$ flows per second, or about 9.2 microseconds per flow in aggregate. If a large hospital network produces a peak of 40,000 new flows per second, the system has headroom of $108,108 / 40,000 = 2.7$ times, which indicates that the design is suitable for online scoring on modest commodity hardware.



Received: 12-08-2026

Revised: 25-09-2026

Accepted:06-10-2026

6. Discussion

Three findings stand out. First, optimization matters for KANs. The untuned KAN already outperformed the multilayer perceptron, but DBO added a further 0.019 in F1 and reduced the false positive rate by 30%. Because the spline grid, spline order, and regularization interact, manual tuning is unlikely to find comparable settings, and a population-based method with a size penalty explores these interactions efficiently (Xue & Shen, 2023). Second, data engineering is as important as model choice. Class balancing and feature selection together account for nearly half of the combined ablation loss, which echoes the observation by Ahmad et al. (2021) that preprocessing and dataset construction strongly influence reported performance. Third, interpretability need not cost accuracy. The model's top-ranked features are behaviorally meaningful, and the KAN's univariate edge functions provide a second, intrinsic explanation that complements post hoc SHAP values.

The results also place DBO-KAN relative to unsupervised alternatives. Autoencoder ensembles such as Kitsune detect anomalies without labels and are attractive when no attack data exist at all (Mirsky et al., 2018). However, they cannot exploit labeled knowledge of related attack families, and their false positive behavior on heterogeneous clinical traffic can be difficult to control. A supervised detector evaluated under a leave-family-out protocol, as here, captures regularities shared across families and still tests on unseen ones. A hybrid that uses an anomaly score as an additional input to the KAN is a natural extension.

For practitioners, the framework has three operational advantages. The model is small, with 2,496 parameters, so it can run beside a network tap without specialized accelerators. The false positive rate is controlled explicitly through the threshold and the validation protocol, which matters in settings where isolating a device can disrupt care. The explanations are expressed in network terms, so they can be reviewed jointly by security staff and clinical engineers.

7. Limitations and Future Work

Several limitations qualify these conclusions. The zero-day families are held-out members of known benchmark collections plus emulated traffic, and a truly novel exploit may behave differently from any family used here. Benchmarks such as CICIDS2017 and UNSW-NB15 also age quickly as protocols and attack tooling evolve (Ahmad et al., 2021). The cost model in Section 5.2 relies on assumed costs, and real costs vary widely among institutions. The framework analyzes flow statistics only, so encrypted payload content, host telemetry, and device physiological signals are not used, even though combining network and medical data has been shown to help in clinical settings (Hady et al., 2020). Finally, DBO adds a one-time tuning cost of 1,500 model evaluations, and it must be repeated if the traffic distribution shifts substantially.



Received: 12-08-2026

Revised: 25-09-2026

Accepted:06-10-2026

Future work should proceed along four lines. The first is validation on live, privacy-preserving hospital traces and on real zero-day incidents. The second is concept-drift handling, in which the detector is retrained or the threshold recalibrated when drift statistics exceed a limit. The third is federated training across hospitals, which would share model updates rather than raw traffic. The fourth is the fusion of flow, host, and device signals, together with adversarial evaluation to test robustness against evasion.

8. Conclusion

This article presented DBO-KAN, a zero-day detection framework for healthcare networks that combines a Kolmogorov–Arnold Network, Dung Beetle Optimizer tuning, a scalable data-engineering pipeline, and Shapley-based explanations. The mathematical model casts detection as a leave-family-out problem, defines the KAN and the size-penalized fitness explicitly, and supports calculations of model size, optimizer behavior, performance metrics, expected cost, and parallel speedup. On a zero-day test set, the framework reached an accuracy of 0.974, an F1 score of 0.936, an MCC of about 0.920, and a false positive rate of 0.014, outperforming five baselines, and it processed more than 100,000 flows per second on eight workers. Ablation showed that each component contributes, with the spline layer and class balancing contributing most, and the explanation analysis linked alerts to interpretable traffic behavior. Together, these results suggest that an optimized and interpretable KAN is a promising component of defense-in-depth for clinical networks, subject to validation on real-world data.

References

- [1] Ahmad, Z., Shahid Khan, A., Wai Shiang, C., Abdullah, J., & Ahmad, F. (2021). Network intrusion detection system: A systematic study of machine learning and deep learning approaches. *Transactions on Emerging Telecommunications Technologies*, 32(1), Article e4150. <https://doi.org/10.1002/ett.4150>
- [2] Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16, 321–357. <https://doi.org/10.1613/jair.953>
Hady, A. A., Ghubaish, A., Salman, T., Unal, D., & Jain, R. (2020). Intrusion detection system for healthcare systems using medical and network data: A comparative study. *IEEE Access*, 8, 106576–106584.
- [3] Kolmogorov, A. N. (1957). On the representation of continuous functions of many variables by superposition of continuous functions of one variable and addition. *Doklady Akademii Nauk SSSR*, 114(5), 953–956.



Received: 12-08-2026

Revised: 25-09-2026

Accepted: 06-10-2026

- [4] Liu, Z., Wang, Y., Vaidya, S., Ruehle, F., Halverson, J., Soljačić, M., Hou, T. Y., & Tegmark, M. (2024). KAN: Kolmogorov–Arnold networks (arXiv:2404.19756). arXiv. <https://arxiv.org/abs/2404.19756>
- [5] Lundberg, S. M., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. In I. Guyon, U. von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, & R. Garnett (Eds.), *Advances in neural information processing systems* (Vol. 30, pp. 4765–4774). Curran Associates.
- [6] Mirsky, Y., Doitshman, T., Elovici, Y., & Shabtai, A. (2018). Kitsune: An ensemble of autoencoders for online network intrusion detection. *Proceedings of the Network and Distributed System Security Symposium (NDSS)*. <https://doi.org/10.14722/ndss.2018.23204>
- [7] 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>
- [8] Sharafaldin, I., Lashkari, A. H., & Ghorbani, A. A. (2018). Toward generating a new intrusion detection dataset and intrusion traffic characterization. *Proceedings of the 4th International Conference on Information Systems Security and Privacy (ICISSP)* (pp. 108–116). SciTePress. <https://doi.org/10.5220/0006639801080116>
- [9] Xue, J., & Shen, B. (2023). Dung beetle optimizer: A new meta-heuristic algorithm for global optimization. *The Journal of Supercomputing*, 79(7), 7305–7336. <https://doi.org/10.1007/s11227-022-04959-6>