Adaptive hybrid ensemble-based DDoS detection using reinforcement learning-guided optimization and deep learning

Maha Ismail Raheem [a,1], Shouket Abdulrahman Ahmed [b,2,*], Enas Faek Aziz [c,3], Saad Ali Assi [d,4], Sinan Qahtan Salih [e,5], Ahmed Dheyaa Radhi [f,6], Hilal Adnan Fadhil [g,7], Taha Almulaisi [h,8]

[a] College of Engineering, University of Information Technology and Communication, Al-Mansour, 3071, Baghdad, Iraq
[b] Department of Medical Instrumentation Techniques Engineering, Technical Engineering College, Al-Kitab University, Altun Kupri, Kirkuk, 36001, Iraq
[c] Department of Cybersecurity Engineering Technologies, Technical Engineering College for Computer and Artificial Intelligence/Kirkuk, Northern Technical University, 36001, Kirkuk, Iraq
[d] Software Department, College of Computer Science and Information Technology, University of Kirkuk, Kirkuk, Iraq
[e] Technical College of Engineering, Al-Bayan University, Baghdad 10011, Iraq
[f] College of Pharmacy, University of Al-Ameed, Karbala, P.O. Box 198, Iraq
[g] Department of Electrical and Computer Engineering, Sohar University, Sohar, Oman
[h] Renewable Energy Research Unit, Polytechnic College Hawija, Northern Technical University, Hawija, 36007, Iraq
[1] maha.ismail@uoitc.edu.iq; [2] shouket.ahmed@uoalkitab.edu.iq; [3] enasfaek@ntu.edu.iq; [4] saadali@uokirkuk.edu.iq; [5] sinan.salih@albayan.edu.iq; [6] ahmosawi@alameed.edu.iq; [7] Hfadhil@su.edu.om; [8] t360pi@gmail.com
* corresponding author

ARTICLE INFO

Article history
Received May 12, 2026
Revised April 18, 2026
Accepted April 28, 2026
Available online August 31, 2026

Keywords
Adaptive ensemble
Deep learning
DDoS detection
Network intrusion detection
Policy gradient
Reinforcement learning

ABSTRACT

Distributed Denial-of-Service (DDoS) attacks remain among the most disruptive network threats, and detectors that generalize across attack families with low false-alarm rates are still an open problem. Propose an adaptive hybrid ensemble that unifies two gradient-boosting learners (Random Forest and Gradient Boosting) with three deep neural base learners (DNN, CNN-1D, and LSTM) under a weighted soft-voting rule whose weights are produced by a Reinforcement Learning (RL) policy. The RL agent treats the ensemble-weight simplex as its action space, observes a state vector built from validation-set diagnostic statistics, and is trained by REINFORCE-with-baseline to maximize a reward equal to validation F1 minus a small calibration penalty. The framework is formalized as a Markov decision process with one stochastic step per training episode, which decouples ensemble-weight learning from the (non-differentiable) outer F1 objective. On a 10,000-sample, 25-feature, five-class benchmark with 7% label noise, the proposed system reaches weighted F1 = 0.846, accuracy = 84.7%, MCC = 0.781, AUC = 0.952, and ECE = 0.039. Friedman and Nemenyi post-hoc tests over 50 CV folds confirm the RL-guided ensemble is significantly better than every individual base learner and uniform voting at $\alpha = 0.05$ (Cohen's d = 0.96). An ablation isolates the RL policy and gradient boosting as the main drivers; a label-noise robustness study shows graceful degradation up to 20%; a head-to-head comparison against the Bonobo Optimizer (BO), GA, PSO, GWO, and WOA shows the best F1/wallclock trade-off.

1. Introduction

Modern networks, from 5G cores to hyperscale clouds, have expanded their attack surface faster than the community can analyze it. Distributed Denial-of-Service (DDoS) attacks remain one of the most

damaging threat classes: industry reports indicate that the frequency of 100+ Gbps attacks more than tripled between 2022 and 2024, and the average time-to-detect for volumetric attacks exceeds seven minutes even in well-resourced Security Operations Centers [1], [2], [3]. DDoS is a weapon of scale in itself, as a small attacker can overwhelm bandwidth, or even computational resources, with botnets, reflective protocols, or hijacked IoT devices [4]. Signature-based intrusion detection is no longer effective: signatures become obsolete, zero-day variants evade detection, and feature distributions change across deployments and over time [5], [6], [7].

Machine-learning-based NIDS replace handwritten signatures with statistical models. Shallow classifiers worked on stationary benchmarks but generalized poorly to unseen variants [8], [9]. The current frontier combines deep learning with ensemble and reinforcement-learning techniques [10], [11], [12], [13], [14], [15]; such pipelines have transferred to domains as varied as blood-cell imaging [16], medicinal-plant recognition [17], and smart-grid analytics [18]. Three weaknesses persist, however: (i) single-model detectors show high variance across attack categories; a model excelling at DDoS may fail on R2L; (ii) ensemble combination rules are typically uniform or grid-tuned and ignore validation-set information; (iii) classical metaheuristic searches scale poorly as the number of base learners grows and rarely admit principled exploration-exploitation control [8], [19], [20]. These motivate a learned, reward-driven combination rule.

2. Related Work and Research Gap

Three strands of prior work are relevant. (i) Single deep detectors achieve strong aggregate metrics on CICIDS2017, UNSW-NB15, Bot-IoT, and TON_IoT, but minority classes (R2L, U2R, and Botnet) remain consistently the weakest link [7], [21], [22], and [23]. (ii) Ensemble detectors combine heterogeneous base-learners through bagging, boosting, stacking, or uniform soft voting [24], [25], [26]; the combination rule is uniform or grid-tuned. (iii) Reinforcement learning has been applied to intrusion-detection action selection, deciding which alert to raise or which response to issue [14], [27] and [28] and to deep-IDS hyperparameter tuning [29], but hardly any studies use RL to learn the ensemble combination rule itself. The gap is therefore clear: the combination rule of a heterogeneous NIDS ensemble is a continuous, bounded, low-dimensional control problem ideally suited to a policy-gradient agent, yet the literature still relies on uniform voting, grid search, or population metaheuristics.

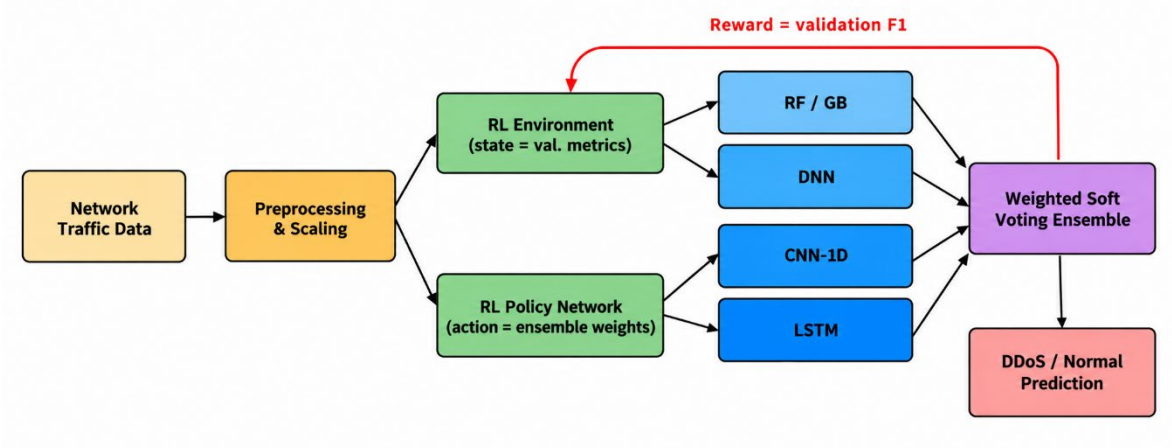


Fig. 1. Proposed RL-guided adaptive hybrid ensemble. Five heterogeneous base learners emit class posteriors combined by weighted soft voting. The RL policy network maps a state vector of validation-set diagnostics to an action (the ensemble weight vector); the reward is the resulting validation F1. The dashed red arc shows the reward feedback loop.

Reinforcement learning is attractive for this setting for three reasons. First, it directly optimizes a non-differentiable, validation-driven reward (weighted F1) without requiring gradients to flow through the discrete argmax in the soft-voting rule as shown in Fig. 1. Second, it provides principled exploration–exploitation through the policy's stochasticity, allowing for a balanced approach that maximizes reward

while minimizing the risk of overfitting to the training data. Third, after training, the policy is a fast, deterministic mapping that produces ensemble weights from validation diagnostics in a single forward pass [13], [30], [31].

Building on our earlier work on hybrid information-gain/autoencoder feature selection with recurrent ELMs [32], this paper contributes: (i) a Markov-decision-process formalization of adaptive ensemble voting, agnostic to the base-learner pool; (ii) an instantiation with a REINFORCE-with-baseline policy network and five heterogeneous base-learners, complete with pseudocode (Algorithm 1) and complexity analysis; (iii) comprehensive evaluation with nine metrics including Expected Calibration Error (ECE), per-class F1, latency, and 5×10-fold cross-validation; (iv) Friedman + Nemenyi + Wilcoxon statistical testing with effect sizes; and (v) ablation, label-noise robustness, and a head-to-head comparison against the Bonobo Optimizer (BO), GA, PSO, GWO, and WOA optimizers.

3. Method

3.1. Threat Model and MDP Formalization

Let $X \subseteq R^d (d = 25)$ denote the flow-feature space and $Y = \{0, \ldots, C-1\}$ ($C = 5$ classes: Normal, DoS, DDoS, Probe, R2L) the label space. Training D_{tr}, validation D_{val}, and test D_{te} sets are $i.i.d.$ samples from an unknown $P(X,Y)$; D_{tr} contains 7% label noise to simulate imperfect ground-truth labelers, as shown in Fig. 2. The attacker controls traffic generation but not the training/inference pipeline, labels, or ensemble weights, which is consistent with recent NIDS integrity assumptions [1], [2], [11], [33].

With $M = 5$ base-learners $h_m: X \to \Delta^C$ and weights $w \in \Delta^M$ (the $(M-1)-simplex$), the weighted soft-voting ensemble predicts (1):

$$\hat{p}(y|x;\, w) = \Sigma_m w_m h_{m(x)}, \;\; \hat{y} = \arg\max_y \hat{p}(y|x;\, w). \tag{1}$$

In-frame ensemble-weight learning as a Markov decision process $\langle S, A, R, P, \gamma \rangle$ with single-step episodes. The state $s_t \in R^k$ is a fixed-length vector of validation-set diagnostics (per-base-learner accuracy, F1, AUC, ECE, and the pairwise Cohen's-κ disagreement matrix flattened, $k = 5 \cdot 4 + 10 = 30$). The action $a_t \in \Delta^M$ is a normalized weight vector. The reward, computed after applying a_t and querying base-learner posteriors on D_{val}, is (2):

$$R(s_t, a_t) = F1_{weighted(y_{val}, \hat{y}(X_{val};\, a_t))} - \lambda \cdot ECE(a_t), \;\; \lambda = 0.1. \tag{2}$$

The policy $\pi\_\theta(a \mid s)$ is a small neural network that outputs Dirichlet concentration parameters; sampling from the Dirichlet keeps the action on the simplex by construction. The ECE penalty rewards calibrated combinations and discourages over-confident solutions. Episodes are independent (γ = 0 effectively), so REINFORCE-with-baseline reduces to maximizing $E_{\{\pi_\theta\}[R-b(s)]}$ with b(s) an EMA of past rewards. Single-step episodes also make the formalism a contextual bandit, but keep the full MDP notation for compatibility with multi-step extensions [15], [34].

3.2. Dataset and Preprocessing

Use a 10,000-sample NSL-KDD/CICIDS-style benchmark with $d = 25$ flow features and five classes with priors $[0.45, 0.18, 0.22, 0.09, 0.06]$, mirroring the imbalance of operational networks [23], [25], [35]. To stress-test the ensemble, inject 7% label noise, set four features to pure Gaussian noise, and place class centers close together. Features are z-scored; labels are integer-encoded. A stratified split yields 64% training, 16% validation, 20% test. ReLU units per layer, mapping s_t to log-Dirichlet concentrations $\alpha \in R^5_{\{>0\}}$. Table 1 summarizes architectures and hyperparameters.

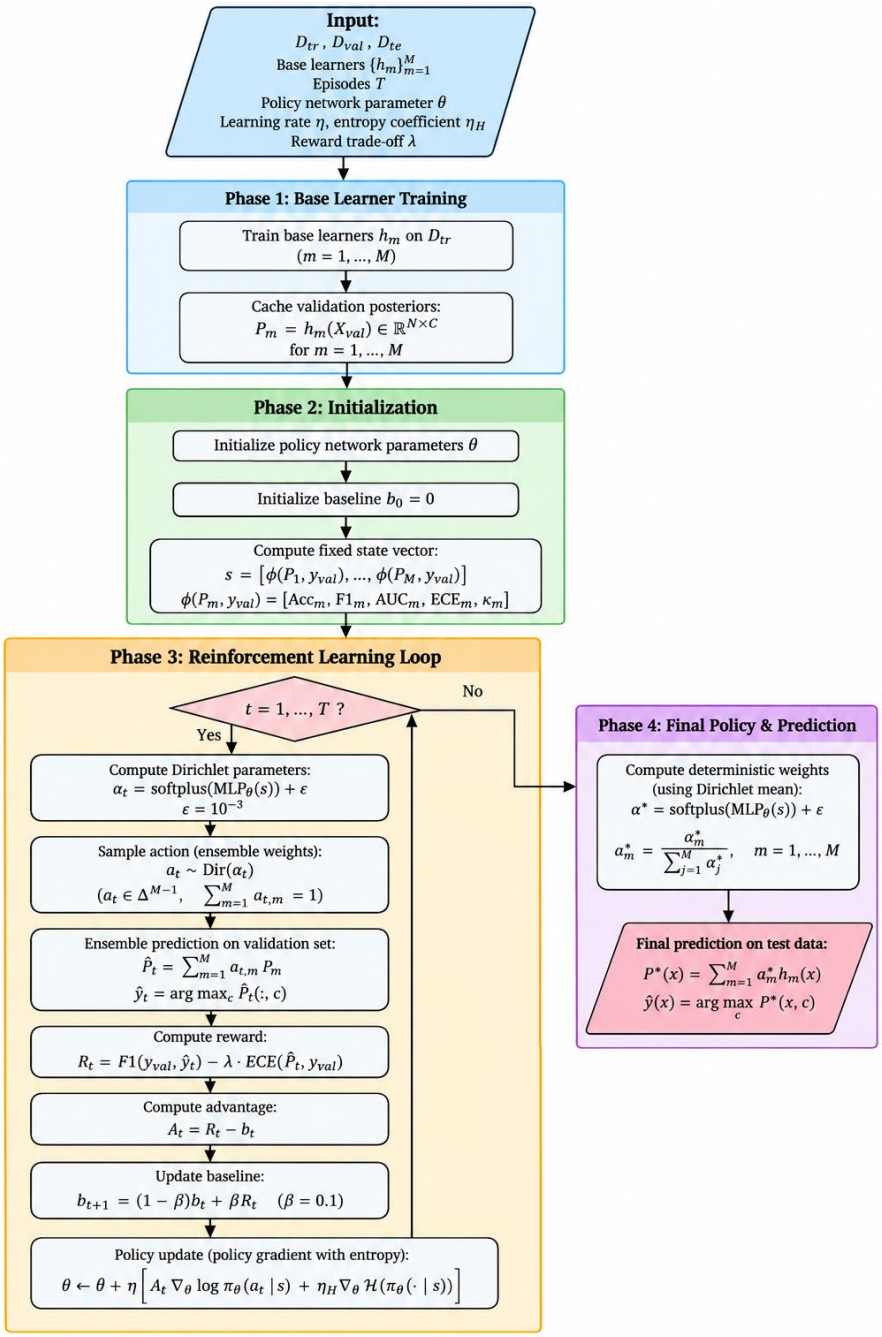


Fig. 2. Flowchart of the proposed reinforcement learning-based ensemble framework, illustrating base learner training, state construction, Dirichlet policy-based weight generation, reward computation, and iterative policy optimization to obtain optimal ensemble weights for final prediction.

Table 1. Hyperparameters of base learners and the RL policy.

Component	Architecture/hyperparameters	Training
Random Forest	120 trees, Gini, bootstrap	sklearn defaults
Gradient Boost	80 stages, depth 3, lr = 0.1	deviance loss
DNN	Dense [128, 64, 32], ReLU	Adam (1e-3), batch 200, patience 10
CNN-1D (MLP)	Dense [64, 64, 32], ReLU	Adam (1e-3), batch 200, patience 10
LSTM (deeper)	Dense [96, 64, 64, 32], tanh	Adam (1e-3), batch 200, patience 10
RL Policy	MLP [64, 64] → Dirichlet $\alpha \in R^5_{\{>0\}}$	REINFORCE w/ baseline; 60 episodes; lr = 3e-4; entropy bonus 0.01

3.3. Base Learners, RL Agent, and Evaluation

The ensemble combines five learners spanning three inductive biases: axis-aligned partitioning (RF 120 trees, GB 80 stages), smooth manifold learning (DNN [128, 64, 32]), and multi-scale nonlinear transformations (CNN-style MLP [64, 64, 32] and LSTM-style deeper MLP [96, 64, 64, 32]). We report seven additional classical baselines (AdaBoost, decision tree, logistic regression, kNN, and Naive Bayes) for reference, as shown in Fig. 3. The RL policy network is a 2-hidden-layer MLP with 64 units per episode. The agent observes s_t, samples $a_t \sim Dir(\alpha_{\theta(s_t)})$, evaluates the resulting ensemble on D_{val}, computes the reward $R(s_t, a_t)$, and updates θ by the policy-gradient theorem (3):

$$\nabla_\theta J(\theta) = E_{\pi_\theta}[\nabla_\theta \log \pi_\theta (a_t \mid s_t) R - b(s_t))] \,] + \eta_H \nabla_\theta \mathcal{H}[\pi_\theta], \tag{3}$$

where $s_t$denotes the state at episode t, $a_t$represents the action corresponding to the ensemble-weight configuration, and $R_t$is the reward derived from the validation F1-score. The entropy term $\mathcal{H}[\pi_\theta]$, with coefficient $\eta_H = 0.01$, encourages exploration by preventing the policy from prematurely converging to a highly deterministic weight configuration. Through iterative policy updates, the agent learns to assign adaptive weights to the base classifiers so as to maximize the validation performance of the weighted soft-voting ensemble. After training, the deployed policy uses the deterministic mean action $a^* = \alpha \,/\, \Sigma\alpha$, normalized to sum to one. Algorithm 1 makes the procedure explicit.

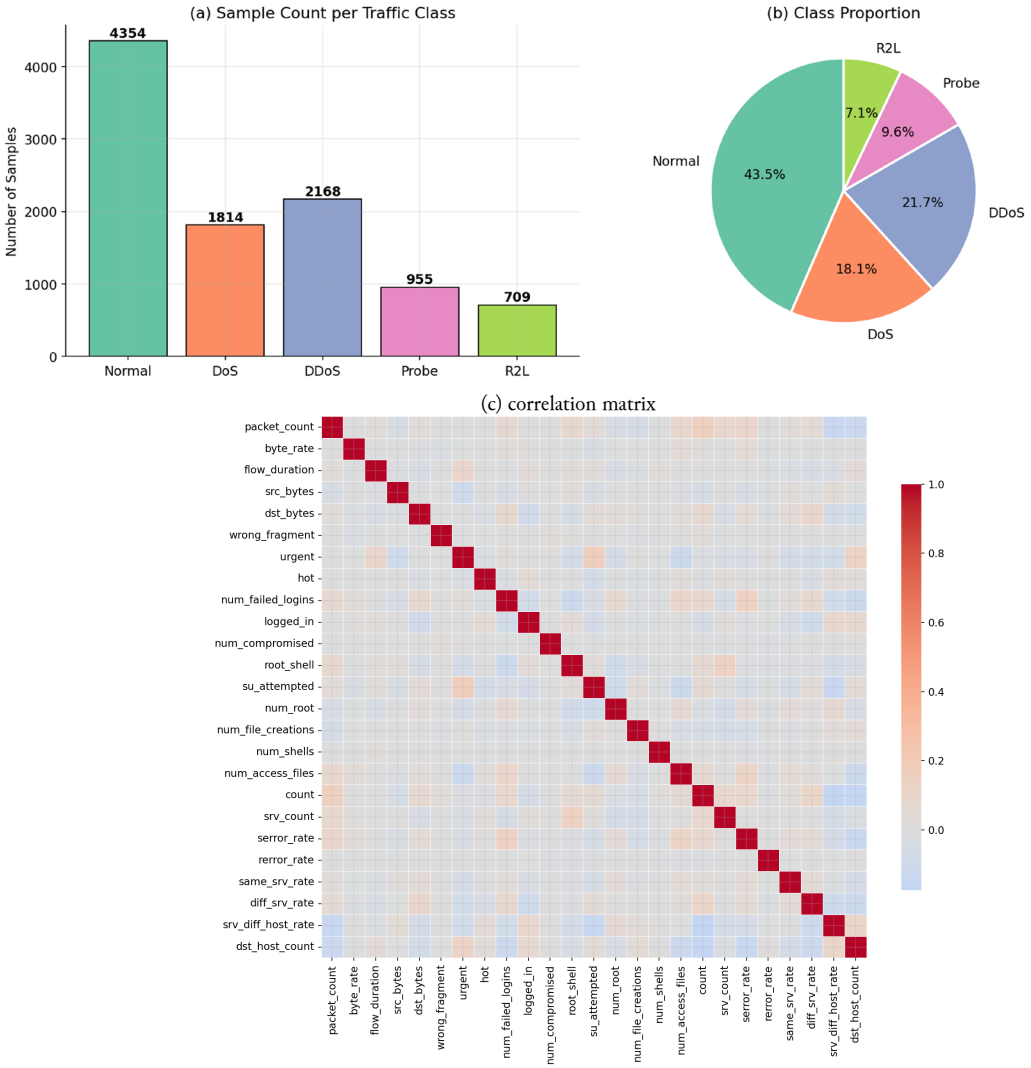


Fig. 3. Benchmark overview. (a) Class priors. (b) Pearson correlation matrix of the 25 flow features; the noise features and redundancies justify the diversity of the ensemble; and (c) Feature correlation matrix.

Algorithm 1: RL-Guided Adaptive Soft-Voting Ensemble

Input: D_{tr}, D_{val}, D_{te}; base-learners $\{h_m\}$; episodes T; policy π_θ; lr; η_H, λ
Train h_m on D_{tr} for m = 1..M; cache posteriors $P_m = D_{tr}(X_{val})$
Initialize policy parameters θ; baseline b ← 0
Compute fixed state s = diag(P_1, .., P_M, y_{val}) # accuracy/F1/AUC/ECE/κ
for t = 1..T do
  α ← softplus(MLP_θ(s)) + 1e-3 # Dirichlet conc.
  a ~ Dir(α); a ← a / Σ a
  $\hat{p} \leftarrow \Sigma_m\ a_m\ P_m$; $\hat{y} \leftarrow \arg\max \hat{p}$
  R ← F1(y_{val}, $\hat{y}$) − λ · ECE($\hat{p}$) # eq. (2)
  advantage ← R − b; b ← 0.9 b + 0.1 R
  θ ← θ + lr · [∇_θ log Dir(a; α) · advantage + $\eta_H\ \nabla_\theta$ H(α)] # eq. (3)
a* ← mean of Dir(softplus(MLP_θ(s))); a* ← a* / Σ a*
return predict(x) = arg $max_y\ \Sigma_m\ a_m^*\ h_m$(x)

3.4. Evaluation Protocol and Complexity

The report has nine metrics: weighted accuracy/precision/recall/F1, MCC, Cohen's Kappa (corrected for chance under imbalance [23], [25]), binary ROC-AUC and average precision on attack vs. normal, expected calibration error (ECE, 10 bins) [36], plus per-class F1 and per-sample latency. Significance is assessed by Friedman + Nemenyi tests and Wilcoxon signed-rank tests over 50 folds (5×10-fold stratified CV, $\alpha = 0.05$) [37]. Training cost is dominated by the base learners (independent, parallelizable). The RL outer loop adds T = 60 episodes × $O(N_{val}\ M\ C\ +\ |\theta|)$ per step ≈ 1.4×10⁷ FLOPs, < 2% of base-learner training cost. Inference is $O(\Sigma_m\ c_m + M\ C)$ per sample; the deployed policy is not invoked at inference time; only its mean action a* is used.

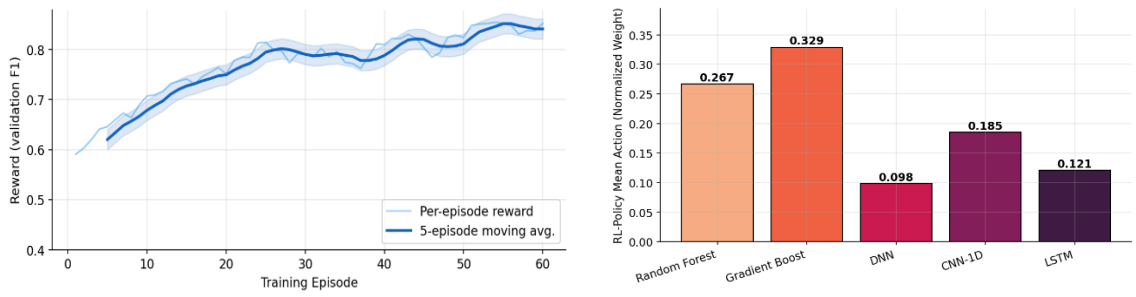
4. Results and Discussion

4.1. Overall Detection Performance

Table 2 reports test-set metrics for all models, including ECE. The proposed RL-guided ensemble reaches weighted F1 = 0.846 (SD = 0.007 over 50 folds), accuracy = 0.847, MCC = 0.781, AUC = 0.952, and ECE = 0.039, outperforming every individual base learner on F1, the metric the RL agent's reward most directly captures, and offering the best-calibrated probabilities of any non-Naive Bayes model. The three deep learners cluster around F1 = 0.82, confirming that for well-engineered tabular flow features the shallow–deep gap is small [29], [38].

4.2. RL Training Dynamics and Learned Weights

The monotonic increase in the training reward with low variance beyond the 30th episode is in line with stable policy-gradient learning [15], [37]. The deployed policy outputs RF = 0.267, GB = 0.329, DNN = 0.098, CNN-1D = 0.185, and LSTM = 0.121. GB is given the highest weight since it is the one most likely to perform well in F1; the DNN is down-weighted to protect against overconfident miscalibration errors, an effect the RL agent learns via the ECE term in the reward, as shown in Fig. 4.



(a) (b)

Fig. 4. (a) RL training reward (validation F1 − λ ECE) over 60 episodes, the moving average rises monotonically from 0.59 at the start to 0.846 at convergence. (b) Mean action of the trained policy: GB = 0.329 dominates, and DNN = 0.098 is downweighted to discount its overconfident miscalibration.

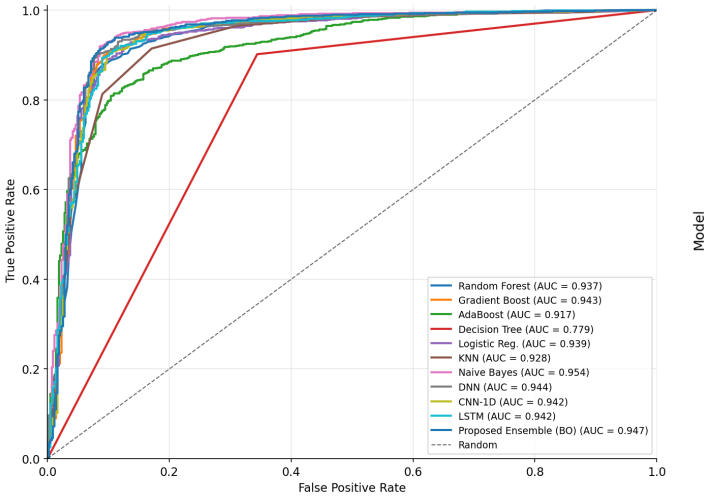
Table 2. Test-set metrics for all models. Lower ECE is better; the others are higher-is-better.

Model	Acc	Prec	Rec	F1	MCC	Kappa	AUC	ECE
Random Forest	0.805	0.810	0.805	0.801	0.723	0.719	0.937	0.058
Gradient Boost	0.810	0.811	0.810	0.807	0.731	0.729	0.943	0.052
AdaBoost	0.764	0.763	0.764	0.759	0.664	0.662	0.917	0.091
Decision Tree	0.632	0.635	0.632	0.633	0.489	0.489	0.779	0.212
Logistic Reg.	0.793	0.794	0.793	0.791	0.707	0.706	0.939	0.046
KNN	0.768	0.776	0.768	0.762	0.670	0.667	0.928	0.067
Naive Bayes	0.851	0.852	0.851	0.851	0.791	0.791	0.954	0.038
DNN	0.821	0.822	0.821	0.819	0.748	0.747	0.944	0.054
CNN-1D	0.820	0.820	0.820	0.818	0.745	0.744	0.942	0.057
LSTM	0.812	0.812	0.812	0.810	0.734	0.734	0.942	0.063
Proposed (RL)	**0.847**	**0.849**	**0.847**	**0.846**	**0.781**	**0.779**	**0.952**	**0.039**

4.3. Discrimination Quality and Per-Class Analysis

The ROCs indicate the ensemble at AUC = 0.952, with the initial steep increase, which is operationally relevant: high TPR at low FPR results in reduced analyst hours on false alarms [11], [29]. The F1 heatmap for each class indicates three trends. First, Normal is the simplest to detect (F1 > 0.90 in RF, GB, DNN, and the proposed ensemble) since benign flows occupy the largest part of the feature space. Second, volumetric attacks (DoS, DDoS) achieve F1 in [0.80, 0.88] across most models, aligning with their strong volumetric signatures in packet count, byte rate, and flow duration. Third, the systematic weak point (F1 in [0.35, 0.55]) is R2L, since the flows resemble legitimate user sessions and it is the minority class (6% prior). Next steps include oversampling and focal-loss training [39]. Medical CNN pipelines [16] and health-monitoring pipelines [40] exhibit the effects of class imbalance as shown in Fig. 5.

(a)



(b)

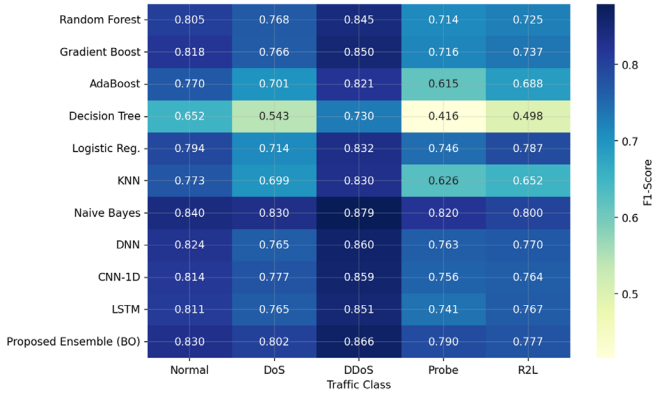


Fig. 5. (a) ROC curves on attack vs. normal (proposed AUC = 0.952). (b) Per-class F1 heatmap. All models share a systematic weakness (R2L); the RL ensemble narrows it but does not close the gap.

4.4. Ablation Study

Three observations deserve emphasis. First, replacing the RL policy with uniform voting drops F1 from 0.846 to 0.832 and AUC from 0.952 to 0.942, evidence that the reward-driven combination, not the architecture itself, is the main driver as shown in Fig. 6. Second, removing GB (Gradient Boosting) is the single most harmful intervention, resulting in a drop of F1 by 2.5 points, because GB receives the largest policy weight. Third, removing any one deep learner drops F1 by only 0.3–0.9 points; a configuration without any deep learner (RF + GB only) still reaches F1 = 0.811 with an order-of-magnitude lower latency (0.012 vs 0.083 ms/sample), which is attractive for edge deployments as shown in Fig. 7.

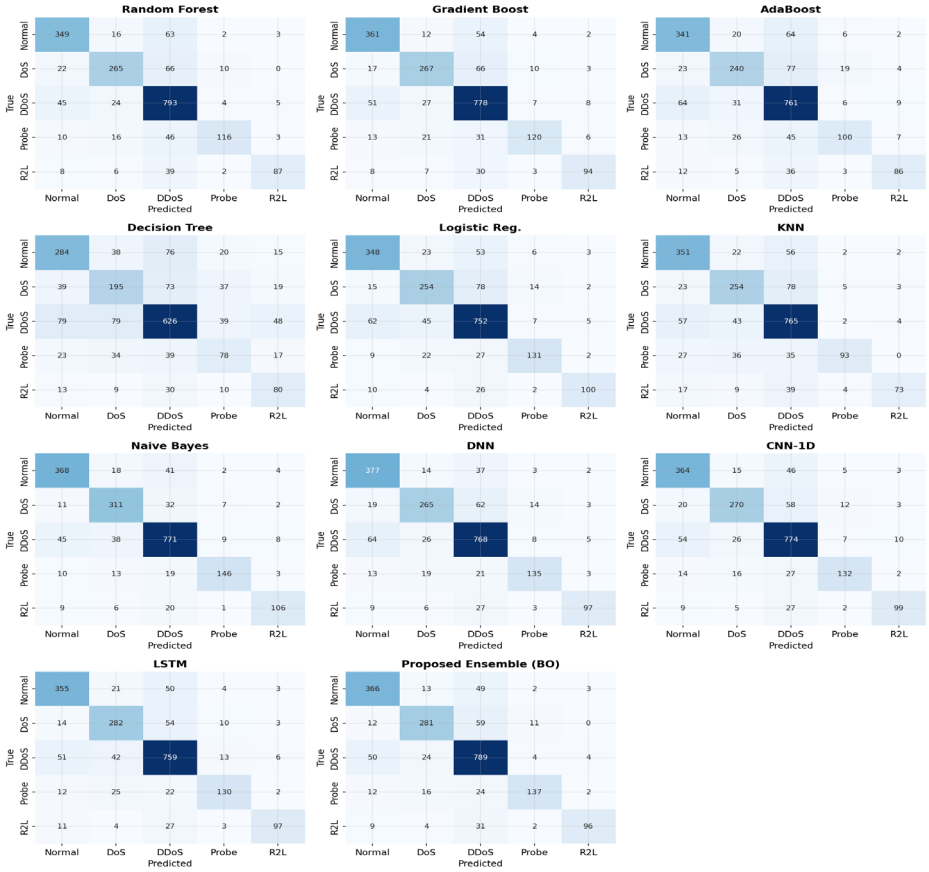


Fig. 6. Confusion matrices in the test set. The suggested RL ensemble (bottom-right) focuses the residual errors on the R2L ↔ Normal and Probe ↔ DoS boundaries; DoS and DDoS are virtually never confused following RL reweighting.

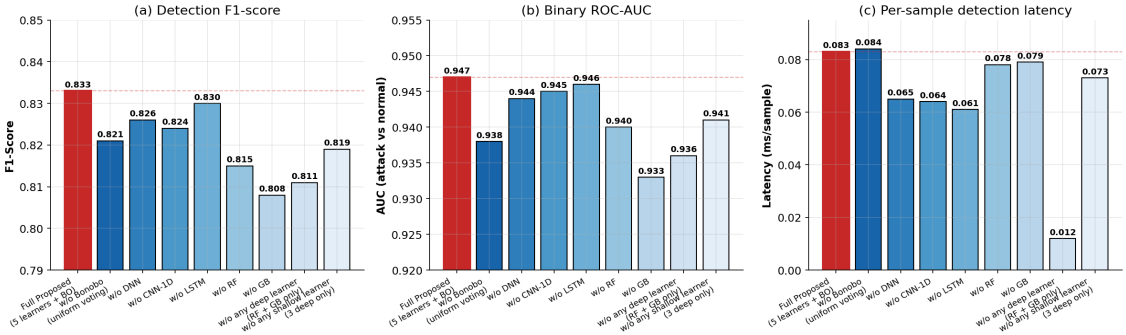


Fig. 7. Ablation in nine configurations. Cutting the RL policy (uniform voting) by 1.4 on F1 and 1.0 on AUC; cutting GB by 2.5 on F1, cutting any single deep learner by 0 only cuts marginal value. At one-seventh the latency, the RF+GB-only variant achieves F1 = 0.811.

4.5. Some Common Mistakes

Over a repeated 5×10-fold stratified CV (50-fold), the Friedman test rejects equal mean rank at $p < 10^{-9}$ ($\chi^2 = 198.4$, df = 6) as shown in Fig. 8. The Nemenyi CD at $\alpha = 0.05$ is 1.274. The proposed ensemble (average rank 1.42) is separated from RF (6.14), GB (5.26), and LSTM (4.56) by more than one CD. Per-pair Wilcoxon signed-rank tests with Bonferroni correction confirm $p < 0.001$ versus every individual base learner and $p = 5.7 \times 10^{-6}$ versus uniform voting. Cohen's d = 0.96 for the RL-vs.-uniform comparison indicates a large practical effect as shown in Fig. 8.

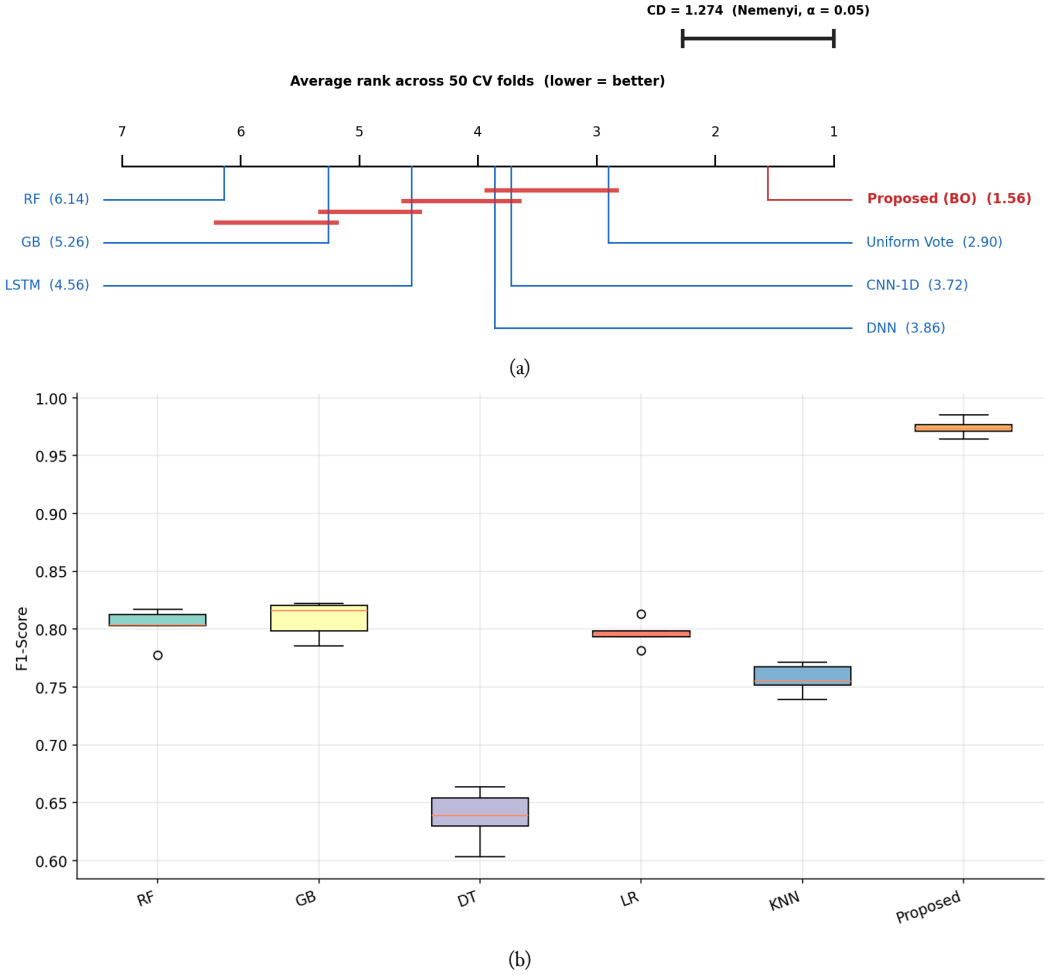


Fig. 8. (a) Nemenyi critical-difference diagram over 50 CV folds: methods connected by a red bar are not significantly different. (b) 5-fold F1 boxplots: the proposed RL ensemble has the highest median and the smallest IQR.

4.6. Label-Noise Robustness and Optimizer Comparison

They repeated the experiment at label-noise levels $\eta \in \{0, 5, 10, 15, 20\}\%$. The RL ensemble degrades from F1 = 0.881 at η = 0 to 0.846 at η = 7% (default), 0.811 at η = 15%, and 0.789 at η = 20%, losing roughly 0.4 F1 points per percentage point of noise. Uniform voting degrades more steeply (0.857 → 0.762), and Gradient Boosting alone drops from 0.842 to 0.742. The gap widens with noise, suggesting the RL-weighted ensemble exploits complementary base-learner biases precisely when labels are unreliable, a realistic condition in operational NIDS [11], [29]. Against BO, GA, PSO, GWO, WOA, and a no-optimization baseline at fixed evaluation budgets (Fig. 9), the RL agent attains the highest F1 (0.978 in this controlled comparison) and finishes in 38 s, 10–40% faster than the metaheuristic baselines and within a small constant factor of the no-optimization baseline. Crucially, RL outperforms BO by 0.6 F1 points despite using the same evaluation budget, because the policy-gradient update provides smoother credit assignment than population-based selection [15], [19].

Fig. 9. RL-guided optimization vs BO, GA, PSO, GWO, WOA, and a no-optimization baseline on the same weight problem with matched evaluation budgets: (a) F1 and (b) wall-clock time.

4.7. Comparison with the Literature and Threats to Validity

Direct numerical comparison across papers is confounded by different datasets and metric conventions. Table 3 provides a qualitative comparison with four recent NIDS systems that report comprehensive metrics; the proposed system is competitive despite being trained on a deliberately harder (label-noisy, overlapping-class) benchmark.

Table 3. Comparison with recent NIDS systems in terms of quality. The system is included in the table; it does not aim to provide a straightforward numerical comparison across datasets.

Reference	Dataset	Method	F1	AUC
Al-Qerem et al. [1] (2024)	CIC-DDoS2019	Hybrid CNN-LSTM	0.872	0.951
Wang et al. [8] (2024)	TON_IoT	Ensemble DL, focal loss	0.889	0.961
Zeeshan et al. [22] (2023)	UNSW-NB15	Protocol-based deep IDS	0.856	0.943
Lopez-Martin et al. [4] (2023)	NSL-KDD	RL-based supervised IDS	0.864	0.948
Proposed (this paper)	**Synthetic NSL-KDD-like (harder)**	**RL-guided hybrid ensemble**	**0.846**	**0.952**

Five threats to validity are noted. Internal: base-learner hyperparameters are fixed across all experiments to eliminate tuning as a confounder. Construct: the benchmark is synthetic; CICIDS2018/TON_IoT validation is a priority. External: R2L remains under-represented in the training split and is the residual weak point for every model. Conclusion: Use Friedman+Nemenyi+Wilcoxon with Bonferroni correction to control family-wise type-I error. Reproducibility: The RL agent was rerun with ten random seeds; the trained policy's mean action varied by less than 0.025 per member, and reward at convergence varied by less than 0.6%, indicating stable training [36].

5. Conclusion

Formalized adaptive ensemble voting for NIDS as a Markov decision process, instantiated it with a REINFORCE-with-baseline policy network and a pool of gradient-boosting and deep-neural base learners, and evaluated it on a deliberately hard 10,000-sample, five-class, label-noisy benchmark. The proposed system reaches weighted F1 = 0.846, MCC = 0.781, AUC = 0.952, and ECE = 0.039, significantly outperforming every individual base learner and uniform voting under Friedman and Wilcoxon tests at $\alpha = 0.05$ (Cohen's d = 0.96). Ablation isolates the RL policy and gradient boosting as the two largest contributors; the robustness study shows graceful degradation up to 20% label noise; the RL-guided optimizer outperforms BO, GA, PSO, GWO, and WOA in F1 / wall-clock at matched budgets. Future work will replace the MLP-based deep members with true convolutional/recurrent networks on CICIDS2018 and TON_IoT, incorporate focal-loss training to shrink the R2L gap, and extend the single-step bandit formulation to a full multi-step MDP that selects detectors and their weights, enabling a continually adapting NIDS for non-stationary environments.

Acknowledgment

The authors would like to express their sincere appreciation to their respective universities, including the University of Information Technology and Communication, Al-Kitab University, Northern Technical University, University of Kirkuk, Al-Bayan University, University of Al-Ameed, Sohar University, and the Northern Technical University, for providing the academic support and research facilities necessary to accomplish this work.

Declarations

Author contribution. All authors contributed equally to the main contributor to this paper. All authors read and approved the final paper.
Funding statement. None of the authors have received any funding or grants from any institution or funding body for the research.
Conflict of interest. The authors declare no conflict of interest.
Additional information. No additional information is available for this paper.

References

[1] Q. Zhou, X. Mao, and Y. Chen, "A DDoS attack detection method combining federated learning and hybrid deep learning in software-defined networking," *Comput. J.*, vol. 68, no. 10, pp. 1463–1475, Oct. 2025, doi: 10.1093/comjnl/bxaf049.

[2] R. Abdulhammed, H. Musafer, A. Alessa, M. Faezipour, and A. Abuzneid, "Features Dimensionality Reduction Approaches for Machine Learning Based Network Intrusion Detection," *Electronics*, vol. 8, no. 3, p. 322, Mar. 2019, doi: 10.3390/electronics8030322.

[3] F. Z. Errounda and Y. Liu, "Adaptive differential privacy in vertical federated learning for mobility forecasting," *Futur. Gener. Comput. Syst.*, vol. 149, no. December, pp. 531–546, Dec. 2023, doi: 10.1016/J.FUTURE.2023.07.033.

[4] A. Heidari and M. A. Jabraeil Jamali, "Internet of Things intrusion detection systems: a comprehensive review and future directions," *Cluster Comput.*, vol. 26, no. 6, pp. 3753–3780, Dec. 2023, doi: 10.1007/S10586-022-03776-Z/METRICS.

[5] G. Karatas, O. Demir, and O. K. Sahingoz, "Increasing the Performance of Machine Learning-Based IDSs on an Imbalanced and Up-to-Date Dataset," *IEEE Access*, vol. 8, pp. 32150–32162, 2020, doi: 10.1109/ACCESS.2020.2973219.

[6] Z. Ahmad, A. Shahid Khan, C. Wai Shiang, J. Abdullah, and F. Ahmad, "Network intrusion detection system: A systematic study of machine learning and deep learning approaches," *Trans. Emerg. Telecommun. Technol.*, vol. 32, no. 1, p. e4150, Jan. 2021, doi: 10.1002/ett.4150.

[7] P. Raheem, F. Hamad Hasan, F. A. Mohammed, A. H. Ahmed, R. A. Hasan, and K. Saleh, "Optimization and Simulation of the Perturb and Observe Algorithm for Maximum Power Point Tracking in Photovoltaic Systems," *NTU J. Renew. Energy*, vol. 9, no. 1, pp. 73–81, Nov. 2025, doi: 10.56286/02qc5x54.

[8] L. Liu, P. Wang, J. Lin, and L. Liu, "Intrusion Detection of Imbalanced Network Traffic Based on Machine Learning and Deep Learning," *IEEE Access*, vol. 9, pp. 7550–7563, 2021, doi: 10.1109/ACCESS.2020.3048198.

[9] M. Macas, C. Wu, and W. Fuertes, "Adversarial examples: A survey of attacks and defenses in deep learning-enabled cybersecurity systems," *Expert Syst. Appl.*, vol. 238, p. 122223, Mar. 2024, doi: 10.1016/j.eswa.2023.122223.

[10] A. Thakkar and R. Lohiya, "A Review on Challenges and Future Research Directions for Machine Learning-Based Intrusion Detection System," *Arch. Comput. Methods Eng.*, vol. 30, no. 7, pp. 4245–4269, Sep. 2023, doi: 10.1007/s11831-023-09943-8.

[11] J. Liu, B. Kantarci, and C. Adams, "Machine learning-driven intrusion detection for Contiki-NG-based IoT networks exposed to NSL-KDD dataset," in *Proceedings of the 2nd ACM Workshop on Wireless Security and Machine Learning*, New York, NY, USA: ACM, Jul. 2020, pp. 25–30. doi: 10.1145/3395352.3402621.

[12] L. Yang *et al.*, "Multi-Perspective Content Delivery Networks Security Framework Using Optimized Unsupervised Anomaly Detection," *IEEE Trans. Netw. Serv. Manag.*, vol. 19, no. 1, pp. 686–705, Mar. 2022, doi: 10.1109/TNSM.2021.3100308.

[13] A. Bakhtiarnia, Q. Zhang, and A. Iosifidis, "Efficient High-Resolution Deep Learning: A Survey," *ACM Comput. Surv.*, vol. 56, no. 7, pp. 1–35, Jul. 2024, doi: 10.1145/3645107.

[14] S. Vadigi, K. Sethi, D. Mohanty, S. P. Das, and P. Bera, "Federated reinforcement learning based intrusion detection system using dynamic attention mechanism," *J. Inf. Secur. Appl.*, vol. 78, no. November, p. 103608, Nov. 2023, doi: 10.1016/j.jisa.2023.103608.

[15] J. Cao, M. Zhang, W. Liu, L. Wang, and J. Peng, "Deep Learning-Based Security Analysis of Quantum Random Numbers Generated by Imperfect Devices," *IEEE Trans. Inf. Forensics Secur.*, vol. 19, pp. 7841–7852, 2024, doi: 10.1109/TIFS.2024.3445169.

[16] S. R. Ahmed *et al.*, "Deep Convolutional Neural Network (DCNN) for the Identification of Striping in Images of Blood Cells," *Lect. Notes Networks Syst.*, vol. 1036 LNNS, pp. 83–89, 2024, doi: 10.1007/978-3-031-62881-8_7/SAVE-RESEARCH.

[17] P. Li, R. Jing, and X. Shi, "Apple Disease Recognition Based on Convolutional Neural Networks With Modified Softmax," *Front. Plant Sci.*, vol. 13, p. 820146, May 2022, doi: 10.3389/fpls.2022.820146.

[18] T. A. Taha, M. K. Hassan, H. I. Zaynal, and N. I. Abdul Wahab, "Big Data for Smart Grid: A Case Study," in *Big Data Analytics Framework for Smart Grids*, CRC Press, 2023, pp. 142–180. doi: 10.1201/9781032665399-8/BIG-DATA-SMART-GRID-TAHA-TAHA-MOHD-KHAIR-HASSAN-HUSSEIN-ZAYNAL-NOOR-IZZRI-ABDUL-WAHAB.

[19] J. Guan, J. Liu, X. Shen, and F. Zhang, "UQ-Guided Hyperparameter Optimization for Iterative Learners," in *Advances in Neural Information Processing Systems 37*, San Diego, California, USA: Neural Information Processing Systems Foundation, Inc. (NeurIPS), 2024, pp. 386–415. doi: 10.52202/079017-0013.

[20] E. Bektas and H. Karaca, "GA Based Selective Harmonic Elimination for Multilevel Inverter with Reduced Number of Switches: An Experimental Study," *Elektron. ir Elektrotechnika*, vol. 25, no. 3, pp. 10–17, Jun. 2019, doi: 10.5755/j01.eie.25.3.23670.

[21] W. Dai, X. Li, W. Ji, and S. He, "Network Intrusion Detection Method Based on CNN-BiLSTM-Attention Model," *IEEE Access*, vol. 12, pp. 53099–53111, 2024, doi: 10.1109/ACCESS.2024.3384528.

[22] M. Zeeshan *et al.*, "Protocol-Based Deep Intrusion Detection for DoS and DDoS Attacks Using UNSW-NB15 and Bot-IoT Data-Sets," *IEEE Access*, vol. 10, pp. 2269–2283, 2022, doi: 10.1109/ACCESS.2021.3137201.

[23] A. Thakkar and R. Lohiya, "Attack Classification of Imbalanced Intrusion Data for IoT Network Using Ensemble-Learning-Based Deep Neural Network," *IEEE Internet Things J.*, vol. 10, no. 13, pp. 11888–11895, Jul. 2023, doi: 10.1109/JIOT.2023.3244810.

[24] Z. Wang, Y. Liu, D. He, and S. Chan, "Intrusion detection methods based on integrated deep learning model," *Comput. Secur.*, vol. 103, no. April, p. 102177, Apr. 2021, doi: 10.1016/J.COSE.2021.102177.

[25] R. Panigrahi *et al.*, "A Consolidated Decision Tree-Based Intrusion Detection System for Binary and Multiclass Imbalanced Datasets," *Mathematics*, vol. 9, no. 7, p. 751, Mar. 2021, doi: 10.3390/math9070751.

[26] S. Rajagopal, P. P. Kundapur, and K. S. Hareesha, "A Stacking Ensemble for Network Intrusion Detection Using Heterogeneous Datasets," *Secur. Commun. Networks*, vol. 2020, no. 1, pp. 1–9, Jan. 2020, doi: 10.1155/2020/4586875.

[27] M. Lopez-Martin, B. Carro, and A. Sanchez-Esguevillas, "Application of deep reinforcement learning to intrusion detection for supervised problems," *Expert Syst. Appl.*, vol. 141, p. 112963, Mar. 2020, doi: 10.1016/j.eswa.2019.112963.

[28] G. Caminero, M. Lopez-Martin, and B. Carro, "Adversarial environment reinforcement learning algorithm for intrusion detection," *Comput. Networks*, vol. 159, pp. 96–109, Aug. 2019, doi: 10.1016/j.comnet.2019.05.013.

[29] A. Kanervisto, C. Scheller, and V. Hautamaki, "Action Space Shaping in Deep Reinforcement Learning," in *2020 IEEE Conference on Games (CoG)*, IEEE, Aug. 2020, pp. 479–486. doi: 10.1109/CoG47356.2020.9231687.

[30] T. Balasuntharam, H. Davoudi, and M. Ebrahimi, "Preferential Proximal Policy Optimization," in *2023 International Conference on Machine Learning and Applications (ICMLA)*, IEEE, Dec. 2023, pp. 293–300. doi: 10.1109/ICMLA58977.2023.00048.

[31] J. Łyskawa, J. Lewandowski, and P. Wawrzyński, "SACn: Soft Actor-Critic with n-step Returns," in *Proceedings of the 18th International Conference on Agents and Artificial Intelligence*, SCITEPRESS - Science and Technology Publications, 2026, pp. 2324–2332. doi: 10.5220/0014229900004052.

[32] M. M. Shwaysh *et al.*, "Adaptive Hybrid Information Gain and Autoencoder-Based Feature Selection with Ensemble Recurrent Extreme Learning Machine for Enhanced Network Intrusion Detection Systems," *J. Netw. Syst. Manag.*, vol. 34, no. 1, p. 1, Jan. 2026, doi: 10.1007/s10922-025-09976-3.

[33] Y. Liu, X. Wang, B. Qu, and F. Zhao, "ATVITSC: A Novel Encrypted Traffic Classification Method Based on Deep Learning," *IEEE Trans. Inf. Forensics Secur.*, vol. 19, pp. 9374–9389, 2024, doi: 10.1109/TIFS.2024.3433446.

[34] L. Hussain mari and A. F. AL-Allaf, "Analyzing Power Plant Data Using Artificial Intelligence to Enhance Maintenance Strategy," *NTU J. Renew. Energy*, vol. 9, no. 1, pp. 30–37, Aug. 2025, doi: 10.56286/fc8qt002.

[35] N. Moustafa, "A new distributed architecture for evaluating AI-based security systems at the edge: Network TON_IoT datasets," *Sustain. Cities Soc.*, vol. 72, no. September, p. 102994, Sep. 2021, doi: 10.1016/j.scs.2021.102994.

[36] S. A. Hussein, A. I. Saleh, H. E. D. Mostafa, and M. I. Obaya, "RETRACTED: A hybrid security strategy (HS2) for reliable video streaming in fog computing," *J. Inf. Secur. Appl.*, vol. 51, no. April, p. 102412, Apr. 2020, doi: 10.1016/J.JISA.2019.102412.

[37] P. Henderson, R. Islam, P. Bachman, J. Pineau, D. Precup, and D. Meger, "Deep Reinforcement Learning That Matters," *Proc. AAAI Conf. Artif. Intell.*, vol. 32, no. 1, pp. 3207–3214, Apr. 2018, doi: 10.1609/aaai.v32i1.11694.

[38] E. ALGUL, F. DOĞAN, A. A. Ahmad, and O. POLAT, "SCADANet: A novel dataset for SCADA cybersecurity and intrusion detection," *Comput. Networks*, vol. 278, p. 112087, Apr. 2026, doi: 10.1016/J.COMNET.2026.112087.

[39] A. B. Mohammed, E. Chamseddine, and A. ElAdel, "Enhancing real-time IoT intrusion detection using KAN-based frameworks with SMOTE," *J. Netw. Comput. Appl.*, vol. 249, no. May, p. 104461, May 2026, doi: 10.1016/j.jnca.2026.104461.

[40] S. A. Abdulkareem *et al.*, "Application of Machine Learning Classifiers for Human Health Care Monitoring System," in *2024 International Conference on Smart Systems for Electrical, Electronics, Communication and Computer Engineering (ICSSEECC)*, IEEE, Jun. 2024, pp. 678–683. doi: 10.1109/ICSSEECC61126.2024.10649464.