CARE-GATE: Cost-Aware Uncertainty Gating for Digital Twin based IoT Intrusion Response

1st Nauman Irshad Ali Shah
Department of Computer Science
University of Central Punjab
Lahore, Pakistan
naumanirshadalishah@gmail.com

2nd Syed Qarib Ali Naqvi
Department of Computer Science
University of Central Punjab
Lahore, Pakistan
Contactqaribnaqvi@gmail.com

3rd Muhammad Umer Amir
Department of Computer Science
University of Central Punjab
Lahore, Pakistan
umrramrr@gmail.com

4th Ali Ahmad
Department of Computer Science
University of Central Punjab
Lahore, Pakistan
aliahmaddev61@gmail.com

5th Danish Ali
Department of Computer Science
University of Central Punjab
Lahore, Pakistan
danish.ali02721@gmail.com

6th Ahmad Arsalan
Faculty of Information Technology
and Computer Science
University of Central Punjab
Lahore, Pakistan
ahmad.arslan@ucp.edu.pk

***Abstract*—Machine-learning intrusion detectors usually report predictive performance but do not specify how a detected IoT attack should be handled when response actions have different operational costs. This paper presents CARE-GATE, a cost-aware response policy that combines detector confidence, explanation stability, and a Digital Twin state residual in an uncertainty gate. The gate blocks automatic isolation when evidence is uncertain and uses an explicit cost matrix to calibrate its weights and threshold. We evaluate the framework on a balanced eight-class subset of the real CICIoT2023 dataset, using 5400 training flows and 1800 held-out test flows represented by 44 numerical features. Among six detectors, CatBoost gives the best weighted F1-score (76.09%) with 76.17% accuracy, 96.95% ROC-AUC, and 0.0061 ms/sample model-only inference time. Selective SHAP is invoked for 82.78% of alerts, with a global explanation-stability score of 0.875. CARE-GATE selects monitor, throttle, and isolate for 12.22%, 70.61%, and 17.17% of alerts, respectively. Its mean cost per alert is 0.2331, compared with 0.2336 for a fixed-weight gate. A confidence-only policy is cheaper (0.1661) but isolates fewer alerts (6.17%). These results show how explicit operational costs and uncertainty evidence shape response behavior beyond detector accuracy alone.**

***Index Terms*—Internet of Things, intrusion detection, cost-sensitive response, uncertainty gating, Digital Twin, explainable AI, SHAP, CatBoost**

I. Introduction

Internet of Things (IoT) gateways must identify denial-of-service, reconnaissance, spoofing, brute-force, web, and malware traffic under limited compute and latency budgets [1], [2]. Machine-learning intrusion detection systems (IDS) can assign a label and probability to each flow, but a label alone does not determine a safe response. A false alarm may disconnect a healthy device, while a missed severe attack may allow damage to propagate. The operational problem is therefore not only whether an alert is correct, but whether it should be monitored, throttled, isolated, or sent to an analyst.

Prior IDS work largely emphasizes accuracy, F1-score, and detection latency [3], [4]. Explainable methods attach feature-level evidence to predictions [5], [6], and Digital Twin (DT) security systems maintain virtual state for monitored assets [7], [8]. Cost-sensitive intrusion response provides a separate line of work in which the consequences of an action are part of the decision objective [9]. These components are useful, but they are often evaluated separately. A practical response layer must combine model confidence, supporting evidence, device state, and the asymmetric costs of acting too aggressively or too cautiously.

CARE-GATE addresses this decision layer. A detector supplies class probabilities, a selective SHAP stage supplies explanation evidence, and a lightweight device-state twin supplies a mismatch residual. Their outputs form an uncertainty score. A hard interlock prevents automatic isolation when uncertainty exceeds a calibrated threshold. The calibration stage minimizes the operational cost defined by an explicit cost matrix, rather than selecting the response policy solely on the basis of predictive accuracy.

This work makes four contributions. First, CARE-GATE formulates IoT intrusion response as a cost-aware action-selection problem in which an uncertainty threshold prevents unsafe automatic isolation. Second, it separates offline policy calibration from online response through a compact algorithm that trains the detector, configures the gate, and applies the selected policy to incoming flows. Third, the framework is evaluated using six detectors and three response policies on real CICIoT2023 flow records, with additional ablation tests for explanation stability and the twin residual. Fourth, confidence and action-distribution analyses are reported to show how the selected gate behaves at the evaluated operating point.

II. Related Work and Problem Formulation

Tree-based ensemble methods are widely used for intrusion detection with tabular network-flow data. Random Forest, XGBoost, LightGBM, and CatBoost can capture nonlinear relations among traffic features while maintaining practical inference costs [10]–[12]. Neural-network-based IDS models

are also suitable for real-time detection, although their performance and computational requirements depend on the feature representation and deployment environment [3], [13]. Datasets such as CICIoT2023 and Edge-IIoTset provide diverse benign and malicious traffic records for evaluating these models [1], [14].

Explainability methods such as SHAP and LIME identify the features that contribute most strongly to an IDS prediction [5], [6]. Although these explanations help an analyst assess a model decision, they do not specify which response should follow from that decision. Digital Twin-based security systems address a different part of the problem by maintaining a stateful representation of a monitored device or network and comparing new observations with its expected condition [7], [8], [15]. Some recent Digital Twin IDS designs update this state through replayed or streaming traffic [16]. Cost-sensitive intrusion-response methods, in contrast, select actions according to the consequences of incorrect, delayed, or overly aggressive responses [9]. CARE-GATE combines these elements by using detector confidence, explanation stability, and twin-state consistency within a cost-aware response policy.

For each network flow x_i, the trained detector produces a predicted class $\hat{y}_i$ and a probability vector p_i. The confidence assigned to the prediction is

$$C_i = \max_c p_{i,c}. \tag{1}$$

The predicted class is mapped to an attack-severity level A_i through the operator-defined function $g(\hat{y}_i)$. The explanation module provides a stability score $S_i \in [0,1]$, while the device-state twin provides a normalized residual $R_i \in [0,1]$ that represents the mismatch between the current observation and the stored device state. CARE-GATE must select

$$a_i \in \{\text{monitor}, \text{throttle}, \text{isolate}, \text{escalate}\}. \tag{2}$$

Let $M(y_i, a_i)$ be the operational cost of taking action a_i when the true class is y_i. For candidate weights $w = (w_1, w_2, w_3)$ and threshold τ, the policy objective is

$$J(w,\tau) = \frac{1}{N}\sum_{i=1}^{N} M\left(y_i, \pi(A_i, C_i, S_i, R_i; w, \tau)\right), \tag{3}$$

where $w_j \geq 0$ and $\sum_j w_j = 1$. Automatic isolation is subject to the safety constraint in Section III.

The adversary may generate or replay malicious flows and vary traffic rate, timing, protocols, and connection flags. The defender observes flow-level features at an authenticated gateway and maintains a virtual state for each monitored device. Poisoning, gateway compromise, model extraction, twin-service compromise, and control-channel takeover are outside the present scope.

III. CARE-GATE Framework and Response Algorithm

Figure 1 separates offline policy calibration from online response. Offline, candidate values of (w, τ) are evaluated using labeled calibration data and the operational cost matrix.

Algorithm 1 CARE-GATE calibration and online response

Require: Training set D_{tr}, calibration set D_{cal}, candidate sets $\mathcal{W}, \mathcal{T}$, cost matrix M

Train detector f on D_{tr}
for all $(w,\tau) \in \mathcal{W} \times \mathcal{T}$ **do**
  $J(w,\tau) \leftarrow 0$
  for all $(x_i, y_i, d_i) \in D_{\text{cal}}$ **do**
    $(\hat{y}_i, p_i) \leftarrow f(x_i)$; $C_i \leftarrow \max_c p_{i,c}$; $A_i \leftarrow g(\hat{y}_i)$
    Obtain S_i from the selective explanation gate and R_i from twin d_i
    Compute U_i using (4) and a_i using (5)
    $J(w,\tau) \leftarrow J(w,\tau) + M(y_i, a_i)$
  end for
end for
$(w^*, \tau^*) \leftarrow \arg\min_{w,\tau} J(w,\tau)$
for all incoming flows x **do**
  Compute $\hat{y}, C, A, S, R, U$ using f and (w^*, τ^*)
  Apply (5); execute or recommend the action; update the twin
end for

Online, each flow passes through the detector, evidence gate, uncertainty calculation, and response policy. The chosen action is then written back to the device-state twin.

The detector comparison includes Random Forest, XGBoost, LightGBM, CatBoost, a multilayer perceptron (MLP), and Soft-Voting. All models receive the same 44-dimensional numerical input. The predicted class is converted to severity through $g(\cdot)$. SHAP is invoked for nontrivial, borderline, or high-severity alerts, while easy benign decisions bypass explanation. The evaluation reports a global stability value for the selectively explained alerts, and its contribution is examined through the policy ablation.

For device d, the lightweight twin stores current risk, attack count, last predicted class, confidence, residual, and selected action. Its residual R_i records normalized disagreement between the current alert and the stored device state. This layer is a stateful virtual risk representation rather than a high-fidelity physical model. The combined uncertainty is

$$U_i = w_1(1-C_i) + w_2(1-S_i) + w_3 R_i, \qquad \sum_{j=1}^{3} w_j = 1. \tag{4}$$

The safety interlock permits isolation only when the alert is severe and $U_i \leq \tau$. When isolation is blocked, an operator-defined fallback policy $\rho(A_i, C_i, R_i)$ assigns monitor, throttle, or analyst escalation:

$$\pi_i = \begin{cases} \text{isolate}, & A_i = \text{severe} \ \wedge \ U_i \leq \tau, \\ \rho(A_i, C_i, R_i), & \text{otherwise.} \end{cases} \tag{5}$$

In the evaluated policy, ρ assigns monitor to benign or low-risk cases and throttle to the remaining non-isolated alerts. No test sample activates the analyst-escalation branch.

Algorithm 1 summarizes the CARE-GATE calibration and response procedure. Candidate gate parameters are selected by minimizing operational cost on labeled calibration data, after which the selected configuration is applied to incoming test flows.

IV. Experimental Setup and Evaluation Protocol

The evaluation uses real flow records from CICIoT2023 [1]. A balanced eight-class subset is constructed for Benign, DDoS,

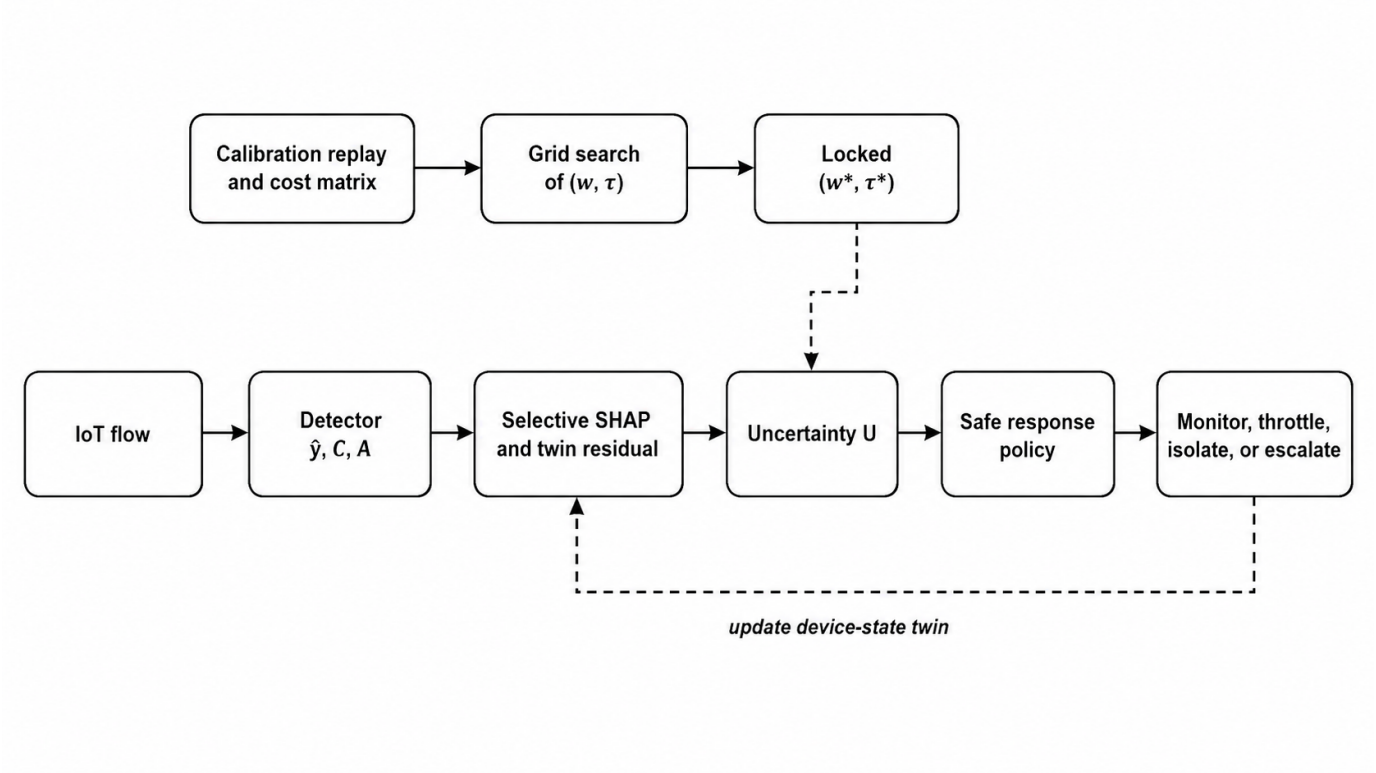

Fig. 1. CARE-GATE process with an offline calibration path and an online response path.

DoS, Recon, Spoofing, BruteForce, Web, and Mirai traffic. Each flow is represented by 44 numerical features. A stratified 75/25 split with seed 42 produces 5400 training flows and 1800 held-out test flows. Twenty virtual devices are used to associate the test sequence with device-state updates.

The detector metrics are accuracy, weighted F1-score, multiclass ROC-AUC, and notebook-reported inference time per sample. The latency values cover model inference only and exclude feature extraction, SHAP, twin updates, communication, and action execution. They are therefore reported as detector-level latency rather than end-to-end response time.

The response evaluation reports action rates, wrong aggressive actions, missed severe attacks, mean uncertainty, and expected cost per alert. Table I lists the operational cost matrix used by CARE-GATE. Wrong containment on low-severity traffic and missed severe attacks receive the highest penalties, whereas monitoring and routine throttling have low costs. Candidate thresholds are $\{0.30, 0.35, 0.40, 0.45, 0.50, 0.55\}$. The grid search selects $w^* = (0.20, 0.30, 0.50)$ and $\tau^* = 0.30$.

All results are obtained under the same reproducible seed-42 train/test protocol. CatBoost records 1371 correct predictions, which gives a 95% Wilson interval of 74.14%–78.08% around its 76.17% accuracy. Pairwise significance between detectors is not claimed because the present study reports one stratified split. Repeated runs can be used in future work to quantify variation across seeds.

TABLE I
OPERATIONAL COST MATRIX USED IN THE CARE-GATE EVALUATION.

Type	Event	Cost
Aggressive error	Wrong isolate/escalate on low-severity truth	25.0
Omission error	Severe attack monitored only	40.0
Low-risk friction	Throttle on benign/low-severity truth	2.0
Justified isolate	Correct but disruptive isolation	0.5
Human-in-loop	Analyst escalation load	3.0
Routine throttle	Throttle otherwise	0.2
Routine monitor	Monitor otherwise	0.0

V. RESULTS AND DISCUSSION

A. Detector Performance

Table II compares the six detectors. CatBoost gives the highest weighted F1-score at 76.09%, followed by Soft-Voting at 74.21%. CatBoost also records 76.17% accuracy and 96.95% ROC-AUC. Its notebook-reported inference time is 0.0061 ms/sample. MLP is also fast at 0.0088 ms/sample but has lower weighted F1 (71.26%). CatBoost is therefore used for the response-policy analysis.

Figure 2 visualizes the weighted F1 ranking. The margin between CatBoost and Soft-Voting is modest, but CatBoost still leads the compared models and does so with a much smaller inference cost than Soft-Voting and Random Forest. This supports its use as the front-end detector for the response stage.

TABLE II
DETECTOR COMPARISON ON THE CICIOT2023 TEST SPLIT.

Model	Acc. (%)	W-F1 (%)	AUC (%)	ms/sample
CatBoost	**76.17**	**76.09**	**96.95**	**0.0061**
Soft-Voting	74.28	74.21	96.65	0.4662
Random Forest	72.06	71.74	95.54	0.4033
MLP	71.17	71.26	96.01	0.0088
XGBoost	70.83	70.80	95.80	0.0568
LightGBM	70.72	70.64	95.79	0.3941

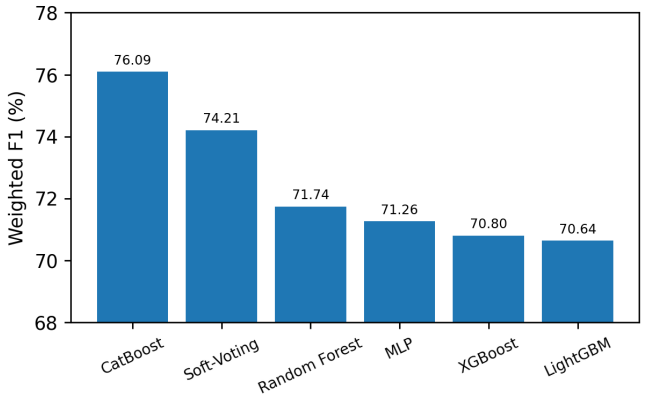

Fig. 2. Weighted F1-score comparison across the six detectors.

The CatBoost confusion matrix in Fig. 3 contains 1371 correct predictions from 1800 flows. Confusion remains between several attack classes, including DDoS and DoS, and among Recon, Spoofing, and Web. This overlap supports the use of an interlock before disruptive action, but it does not by itself establish that the proposed gate is better than a simpler calibrated confidence threshold.

B. Confidence Evidence

Correct predictions have mean confidence 0.720, compared with 0.575 for errors. Figure 4 summarizes this gap. The difference suggests that confidence carries useful reliability information and therefore belongs in the response gate. At the same time, the gap is modest, so confidence alone is not enough to justify an irreversible containment action. A reliability diagram and calibration metrics would still be needed to determine whether the predicted probabilities are well calibrated.

C. Policy Behavior and Additional Result

Selective SHAP is invoked for 82.78% of alerts. The global top-feature stability is $S = 0.875$. Because this value is not alert-specific, it does not reorder uncertainty among samples. The ablation in Table III is consistent with this limitation: removing S leaves the isolate rate and cost unchanged.

CARE-GATE assigns 220 monitor, 1271 throttle, and 309 isolate actions, corresponding to 12.22%, 70.61%, and 17.17% of the test set. Figure 5 shows that throttling dominates the action mix, while isolation is used more selectively. This makes the policy behavior easier to interpret than reporting only cost values.

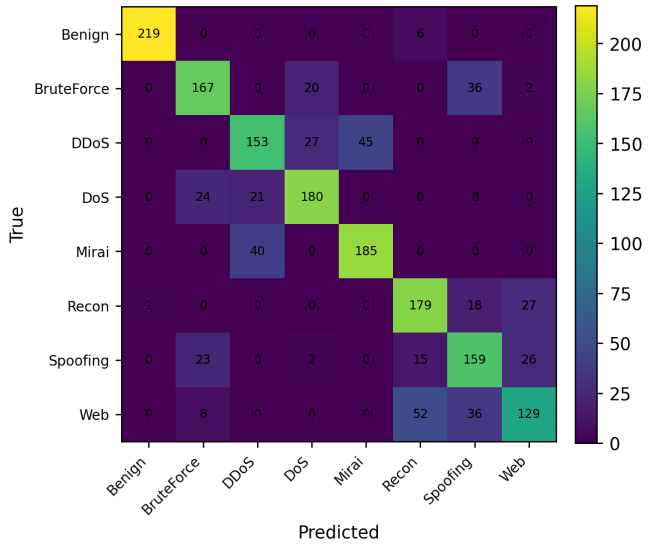

Fig. 3. CatBoost confusion matrix for the eight-class CICIoT2023 test split.

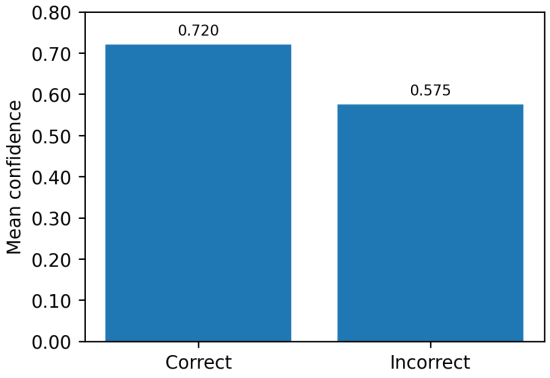

Fig. 4. Mean confidence for correct and incorrect CatBoost predictions.

Table III compares confidence-only, a fixed-weight gate, the full CARE-GATE policy, and two ablations. The calibrated gate reduces cost from 0.2336 to 0.2331 relative to the fixed-weight gate, a difference of 0.0005, while keeping a similar isolation rate. Removing the twin residual returns the policy to the fixed-weight result, so R has a small measurable effect in this evaluation. The confidence-only policy has the lowest cost (0.1661) and zero recorded wrong or missed escalations, but it isolates only 6.17% of alerts. The results therefore expose a policy trade-off between disruption cost and the frequency of decisive containment.

Figure 6 provides a second operational summary by placing mean cost per alert and isolation rate on the same chart. The fixed-weight gate and CARE-GATE are almost identical on isolation frequency, while CARE-GATE remains slightly cheaper. Confidence-only stays farther left on cost, but it also isolates far fewer alerts. This figure makes the cost-isolation trade-off explicit.

The action and cost summaries also clarify why the present study should be read as policy analysis. CARE-GATE does not win by a large margin on the chosen objective. Instead,

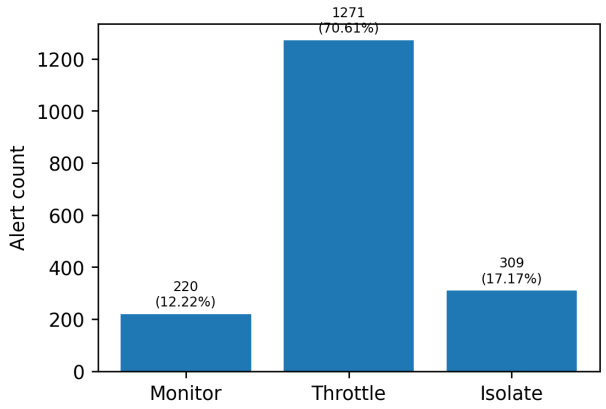

Fig. 5. CARE-GATE action distribution over the 1800-alert test set.

TABLE III
POLICY COMPARISON AND ABLATION ON THE CICIOT2023 EVALUATION.

Variant	Cost	Wrong	Missed	Iso. (%)	Mean U
Confidence-only	0.1661	0.000	0.000	6.17	0.314
Fixed-weight gate	0.2336	0.000	0.000	17.33	0.179
Full CARE-GATE	0.2331	0.000	0.000	17.17	0.113

it exposes how a designer can move between less disruptive and more containment-oriented policies. That distinction is useful even when the best operating point remains application-dependent.

D. Twin-State Evaluation Outcome

The device-state evaluation also yields a twin-level result. All 20 virtual devices end the streamed test sequence in the critical-risk category. Table IV summarizes this outcome. The result shows that the twin accumulates repeated attack evidence rather than treating each alert independently. Because no device returns to a lower state, the present sequence does not evaluate decay, recovery, or post-mitigation stabilization.

This twin-state summary matters because it shows how the state layer influences interpretation. The response logic is not acting on isolated predictions alone. It is acting on predictions that are filtered through a running device context. Future work should preserve that context while adding recovery rules and benign intervals so that the twin reflects both accumulation and stabilization.

E. Scope and Limitations

The evaluation shows that a response layer can combine confidence, explanation evidence, virtual device state, and an explicit action-cost model on real CICIoT2023 traffic. The study uses a balanced subset and one random seed, so repeated runs and cross-dataset validation would strengthen the generality of the results. Explanation stability is reported globally rather than per alert. The virtual device states also saturate: all 20 devices reach the critical category, so the current test sequence does not examine risk decay or recovery. Finally, the latency values exclude most of the response process.

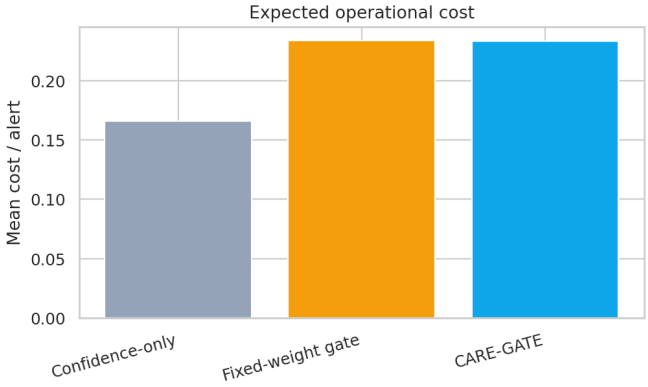

Fig. 6. Cost and isolation-rate comparison for the three response policies.

TABLE IV
SUMMARY OF THE VIRTUAL DEVICE-STATE EVALUATION.

Twin-state metric	Value
Virtual devices in evaluation	20
Devices at critical risk (final)	20
Devices at low/medium/high risk (final)	0
Selective SHAP coverage	82.78%
Global explanation stability	0.875
Analyst escalations triggered	0

Future evaluation should include repeated seeds, cross-dataset testing, calibrated probabilities, per-alert explanation stability, and cost-matrix sensitivity analysis. The response objective can also include a minimum severe-attack containment rate or report a Pareto curve between cost, false isolation, containment, and analyst load. These additions would clarify when the twin-backed gate offers a consistent advantage over a simpler confidence policy.

VI. CONCLUSION

CARE-GATE formulates IoT intrusion handling as cost-aware action selection under uncertainty. The framework combines detector confidence, explanation stability, and a virtual device-state residual, and it prevents automatic isolation above a calibrated uncertainty threshold. On the CICIoT2023 evaluation, CatBoost gives the strongest detector result at 76.17% accuracy and 76.09% weighted F1. CARE-GATE slightly reduces cost relative to a fixed-weight gate, while confidence-only remains cheaper and isolates fewer alerts. The results provide a transparent view of the trade-off between operational cost and containment behavior. Repeated evaluation, per-alert explanation stability, and end-to-end deployment measurements remain important next steps.

ACKNOWLEDGMENT

The authors would like to thank the Center for Emerging Networks & Technologies (CENT) Lab at the University of Central Punjab (UCP), Lahore, Pakistan, for technical support and constructive feedback.

CODE AVAILABILITY

To support public reproducibility, files are available at https://github.com/Nauman-Irshad/CARE_GATE_IoT.

REFERENCES

[1] E. C. P. Neto, S. Dadkhah, R. Ferreira, A. Zohourian, R. Lu, and A. A. Ghorbani, "CICIoT2023: A real-time dataset and benchmark for large-scale attacks in IoT environment," *Sensors*, vol. 23, no. 13, Art. no. 5941, 2023, doi: 10.3390/s23135941.

[2] N. Chaabouni, M. Mosbah, A. Zemmari, C. Sauvignac, and P. Faruki, "Network intrusion detection for IoT security based on learning techniques," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 3, pp. 2671–2701, 2019, doi: 10.1109/COMST.2019.2896380.

[3] A. Kim, M. Park, and D. H. Lee, "AI-IDS: Application of deep learning to real-time web intrusion detection," *IEEE Access*, vol. 8, pp. 70245–70261, 2020, doi: 10.1109/ACCESS.2020.2986882.

[4] M. A. Al-Garadi, A. Mohamed, A. K. Al-Ali, X. Du, I. Ali, and M. Guizani, "A survey of machine and deep learning methods for Internet of Things (IoT) security," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 3, pp. 1646–1685, 2020, doi: 10.1109/COMST.2020.2988293.

[5] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Proc. 31st Conf. Neural Inf. Process. Syst. (NIPS)*, Long Beach, CA, USA, 2017, pp. 4765–4774.

[6] M. A. Yagiz and P. Goktas, "LENS-XAI: Redefining lightweight and explainable network security through knowledge distillation and variational autoencoders for scalable intrusion detection in cybersecurity," *arXiv preprint arXiv:2501.00790*, 2025.

[7] M. Eckhart and A. Ekelhart, "Digital twins for cyber-physical systems security: State of the art and outlook," in *Security and Quality in Cyber-Physical Systems Engineering*, S. Biffl, M. Eckhart, A. Lüder, and E. Weippl, Eds. Cham, Switzerland: Springer, 2019, pp. 383–412, doi: 10.1007/978-3-030-25312-7_14.

[8] Y. Yigit, C. Chrysoulas, G. Yurdakul, L. Maglaras, and B. Canberk, "Digital twin-empowered smart attack detection system for 6G edge of things networks," in *Proc. 2023 IEEE Globecom Workshops (GC Wkshps)*, Kuala Lumpur, Malaysia, 2023, pp. 178–183, doi: 10.1109/GCWkshps58843.2023.10465218.

[9] T. Hussain, A. Beard, L. Chen, C. Nugent, J. Liu, and A. Moore, "From machine learning based intrusion detection to cost sensitive intrusion response," in *Proc. 2022 6th Int. Conf. Cryptography, Security and Privacy (CSP)*, Tianjin, China, 2022, pp. 124–130, doi: 10.1109/CSP55486.2022.00031.

[10] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, San Francisco, CA, USA, 2016, pp. 785–794, doi: 10.1145/2939672.2939785.

[11] G. Ke *et al.*, "LightGBM: A highly efficient gradient boosting decision tree," in *Proc. 31st Conf. Neural Inf. Process. Syst. (NIPS)*, Long Beach, CA, USA, 2017, pp. 3146–3154.

[12] L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush, and A. Gulin, "CatBoost: Unbiased boosting with categorical features," in *Proc. 32nd Conf. Neural Inf. Process. Syst. (NeurIPS)*, Montréal, QC, Canada, 2018, pp. 6638–6648.

[13] N. Shone, T. N. Ngoc, V. D. Phai, and Q. Shi, "A deep learning approach to network intrusion detection," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 2, no. 1, pp. 41–50, Feb. 2018, doi: 10.1109/TETCI.2017.2772792.

[14] M. A. Ferrag, O. Friha, D. Hamouda, L. Maglaras, and H. Janicke, "Edge-IIoTset: A new comprehensive realistic cyber security dataset of IoT and IIoT applications for centralized and federated learning," *IEEE Access*, vol. 10, pp. 40281–40306, 2022, doi: 10.1109/ACCESS.2022.3165809.

[15] A. Rasheed, O. San, and T. Kvamsdal, "Digital twin: Values, challenges and enablers from a modeling perspective," *IEEE Access*, vol. 8, pp. 21980–22012, 2020, doi: 10.1109/ACCESS.2020.2970143.

[16] H. El Alami and D. B. Rawat, "IDS-Twin: Digital twin-based intrusion detection systems for wireless networks," *IEEE Networking Lett.*, vol. 7, no. 4, pp. 255–259, Dec. 2025, doi: 10.1109/LNET.2025.3606789.