An Explainable Fuzzy–AI Framework for Cybersecurity Threat Assessment under Uncertainty

Rahib Imamguluyev*,

Baku Business University, Baku, Azerbaijan

Abstract: Artificial-intelligence-based intrusion detection systems can achieve strong predictive performance, but their probability outputs are often treated as final decisions without an explicit representation of operational context. This study proposes an explainable hybrid Fuzzy–AI framework that separates attack detection from threat assessment. A supervised detector estimates a calibrated probability of malicious activity (AP) from network-flow features, and a Mamdani fuzzy layer combines AP with traffic volume, connection duration, protocol anomaly and source reliability to produce a continuous Cyber Threat Level (CTL, 0–100) and five linguistic priority classes. The revised rule base (49 rules) covers all 243 antecedent combinations, guarantees that a high-confidence detection is never rated below Moderate, and keeps AP influential in every context. Experiments used a reproducible 10% file-wise sample of CICIDS2017 (269,091 flows). Because destination port and initial TCP window sizes act as capture artefacts, the detector was re-evaluated without them, under a stratified random split and two temporal hold-outs, over five seeds. With the leakage-reduced feature set, XGBoost reached F1 = 0.9875 ± 0.0014 on the random split, but F1 = 0.9691 ± 0.0014 on a within-file chronological hold-out and 0.4478 ± 0.0185 when all Friday traffic was held out, showing that near-perfect random-split scores overstate generalisation. Isotonic calibration changed the expected calibration error from 0.0052 ± 0.0005 to 0.0008 ± 0.0002. Run on every test flow with training-only context estimators, the revised fuzzy layer placed 98.60% ± 0.17 of attacks and 1.54% ± 0.14 of benign flows in the High/Critical classes (detector probability alone: 0.11% ± 0.02 of benign flows), whereas the original 35-rule base was undefined for 20.03% ± 0.10 of test flows. When Friday traffic was held out, contextual indicators did not recover the attacks missed by the detector. Controlled analyses showed that two events with the same AP of 0.82 received CTL 81.2 (Critical) and 50.0 (Moderate) under severe and benign context, and variance-based sensitivity analysis attributed the largest share of CTL variance to AP, followed by protocol anomaly and source reliability. SHAP explains the detector and the fired rules explain the threat level, supporting fuzzy contextualisation as a transparent post-detection layer for alert prioritisation rather than as a replacement for the classifier.

Keywords: Cybersecurity; fuzzy logic; explainable artificial intelligence; intrusion detection; SHAP; threat assessment; probability calibration; CICIDS2017.

1. INTRODUCTION

Modern enterprise networks, cloud services, Internet of Things infrastructures, and cyber-physical systems produce high-volume and heterogeneous traffic while facing rapidly changing attack patterns. Intrusion detection systems (IDSs) therefore increasingly employ machine learning (ML) and deep learning to identify nonlinear patterns that are difficult to capture with static signatures. Benchmark datasets such as CICIDS2017 were developed specifically to support more realistic and reproducible evaluation of intrusion detection methods by combining benign traffic with multiple contemporary attack families [1]. At the same time, several audits have shown that benchmark results can be inflated by labelling errors, capture artefacts and evaluation choices [21–24,35], so near-perfect scores must be interpreted with care.

Despite strong classification performance, an important gap remains between detection probability and operational risk. A classifier output such as P(attack)=0.82 expresses model confidence under a learned statistical mapping; it does not, by itself, encode the operational meaning of the event. The same probability may arise for a short connection from a trusted source with limited anomalous behavior or for a high-volume connection from an untrusted source with severe protocol deviation. Treating both events identically can produce poor alert prioritization and can obscure the reasoning needed by security analysts.

A second challenge is interpretability. Complex ensemble and neural models can be difficult to inspect directly. SHAP provides a theoretically grounded additive feature-attribution framework for explaining individual model predictions [2]. In cybersecurity, explainable AI (XAI) has become increasingly important because analysts must understand why an alert was generated, evaluate false positives, and justify consequential responses [3,4,25,26]. NIST likewise identifies explainability and interpretability among the characteristics relevant to trustworthy AI risk management [5].

Fuzzy logic offers a complementary mechanism. Since Zadeh introduced fuzzy sets [6], fuzzy inference has been used to represent gradual membership, linguistic concepts, and approximate reasoning. Mamdani-type inference [7] is particularly attractive for decision support because its rules can be expressed in human-readable IF–THEN form, and linguistic fuzzy systems are a recognised route to explainable AI

*Address correspondence to this author at the Baku Business University, Baku, Azerbaijan;
E-mail: rahib.aydinoglu@gmail.com

[27,28]. Concepts such as high traffic, low source reliability, moderate anomaly, and critical threat can therefore be represented explicitly rather than forced into hard thresholds.

This study combines these two perspectives. The ML component performs data-driven attack detection; the fuzzy component performs context-aware threat assessment. The attack probability is deliberately treated as one input to the risk engine rather than as the final decision. This separation enables a security event to be evaluated according to both learned evidence and contextual conditions.

The principal contribution is therefore not another classifier, but a transparent decision layer placed after the classifier. The framework is designed to answer two distinct questions: (i) why did the AI detector consider the flow suspicious? and (ii) why did the decision-support layer assign the resulting event a particular threat level? SHAP addresses the first question; fuzzy rule activation addresses the second.

The contributions of the paper are:

- A hybrid architecture that explicitly separates AI-based intrusion detection from fuzzy threat assessment, with a stage-by-stage specification of what each component passes to the next (Section 3.1).
- A five-input Mamdani risk engine with fully specified inputs (units, construction and training-only estimators), membership functions, an interpretable CTL scale, a 49-rule base with verified 100% antecedent coverage, and an explicit AP-floor policy.
- An evaluation of the decision layer on every test flow of CICIDS2017, compared with the detector probability alone and with two transparent baselines (weighted score and crisp lookup table) built on the same five inputs.
- A leakage-aware detector evaluation: a leakage-reduced feature set, two temporal hold-outs, isotonic probability calibration with reliability diagrams, and mean ± SD over five seeds.
- Global and local explanations: SHAP for representative true-positive, false-positive, true-negative and false-negative flows, paired with the corresponding fuzzy rule traces.
- Sensitivity analyses of the decision layer (AP sweeps, variance-based Sobol indices and membership-function perturbation) and an accompanying Python implementation.

2. RELATED WORK

2.1. Machine learning for intrusion detection and benchmark pitfalls

CICIDS2017 is widely used for IDS research because it provides labeled flow-level data derived from benign activity and multiple attacks, including brute force, denial-of-service, distributed denial-of-service, botnet, infiltration, Heartbleed, and web attacks [1]. Tree ensembles are particularly attractive for tabular network-flow data. Random Forests aggregate randomized decision trees to improve generalization and robustness [8], while XGBoost provides a scalable gradient-boosting implementation with regularization and sparsity-aware learning [9]. Neural models can capture more complex nonlinear representations, although their gains must be evaluated against computational cost, calibration, and interpretability.

Several studies caution against taking benchmark scores at face value. Engelen *et al.* [21] and Lanvin *et al.* [22] documented flow-construction and labelling errors in CICIDS2017 and showed that correcting them changes detection results. Arp *et al.* [23] list recurring pitfalls in ML-based security research, including spurious correlations introduced by the data-collection process, data snooping and inappropriate random splits; Pendlebury *et al.* [24] show that ignoring temporal order inflates reported performance; and Sommer and Paxson [35] argue that anomaly detection in real networks is far harder than closed-world benchmarks suggest. These findings motivate the leakage-reduced feature set and temporal hold-outs used in this revision.

2.2. Explainable AI in cybersecurity

Lundberg and Lee introduced SHAP as a unified framework for additive feature attribution [2]. Cybersecurity surveys have subsequently emphasized that predictive accuracy alone is insufficient in security-critical environments because analysts require explanations that support trust, debugging, accountability, and response decisions [3,4]. Rjoub *et al.* classify XAI approaches across cybersecurity applications and highlight continuing challenges involving explanation quality, human interpretation, and the trade-off between model complexity and transparency [3]. Charmet *et al.* similarly identify the inability of many AI-based security systems to justify predictions in a human-understandable way [4]. Wang *et al.* [26] applied SHAP to produce local and global explanations of an IDS, and Neupane *et al.* [25] survey explainable IDS designs, distinguishing post-hoc explanation of black-box detectors from intrinsically interpretable (white-box) models such as rule-based systems.

Table 1: Positioning of the proposed framework relative to related approaches.

Approach	Role of fuzzy logic	ML detector probability used	Explanation provided	Evaluation
ML IDS (e.g., RF/XGBoost [8,9])	none	is the output	none or post-hoc	detection metrics
SHAP-explained IDS [2,27]	none	is the output	feature attributions	detection metrics
Fuzzy / GA anomaly IDS [17]	detector	no	fuzzy rules	detection metrics
Neuro-fuzzy IDS [18,20]	detector (learned rules)	replaced by neuro-fuzzy output	learned rules	detection metrics
Fuzzy alert prioritization [15]	post-detection ranking	no	rule-based priority	alert ranking case studies
Fuzzy cyber-risk assessment [16]	risk scoring	no	rule-based risk	illustrative cases
This work	post-detection policy layer	yes, calibrated AP is one of five inputs	SHAP (detector) + rule trace (CTL)	all test flows; temporal hold-outs; baselines; sensitivity

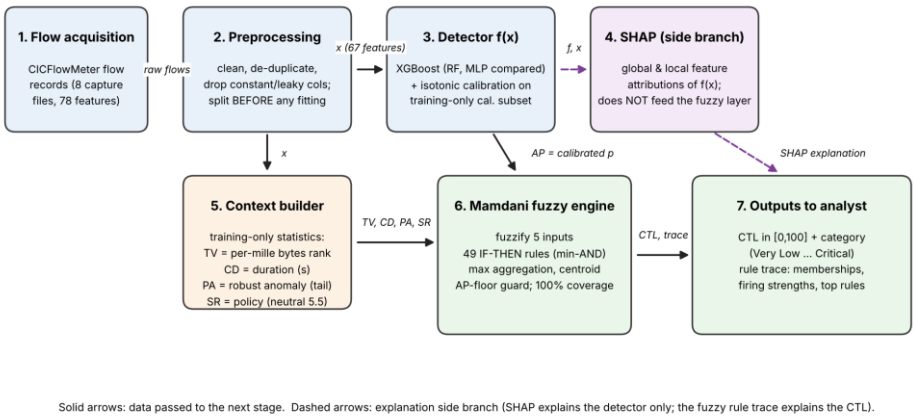


Figure 1: Architecture and data flow of the proposed framework. Solid arrows carry data to the next stage; dashed arrows form the explanation side branch. The detector supplies AP, the context builder supplies TV, CD, PA and SR, and only these five values enter the fuzzy engine.

Table 2: Stage-by-stage data flow.

Stage	Input	Operation	Output passed on
1 Flow acquisition	packets / capture files	CICFlowMeter flow export (CICIDS2017 MachineLearningCSV)	one record per flow, 78 numeric features + label
2 Preprocessing	flow records	remove non-finite values and duplicates, drop constant columns, optionally drop leakage-prone columns; split into training/test BEFORE any fitting	feature vector x; training and test partitions
3 Detector	x	XGBoost (RF and MLP compared); isotonic calibration fitted on a calibration subset of the training partition	AP = calibrated $p \in [0,1]$
4 SHAP (side branch)	f, x, training background sample	TreeSHAP attributions (global mean |SHAP|; local per flow)	feature-level explanation shown to the analyst
5 Context builder	x, training-only reference statistics, SR policy	TV, CD, PA computed from flow fields (Section 3.8); SR from reputation/policy	TV, CD, PA, SR
6 Fuzzy engine	AP, PA, TV, CD, SR	fuzzification, 49 Mamdani rules, max aggregation, centroid, AP-floor guard	CTL, category, rule trace
7 Analyst view	CTL, category, rule trace, SHAP	prioritised alert queue with both explanations	triage decision

2.3. Fuzzy and neuro-fuzzy intrusion detection

Fuzzy sets provide a formal representation for gradual concepts whose boundaries are not naturally crisp [6], and Mamdani's rule-based inference makes such concepts operational through interpretable antecedent–consequent rules [7]. In intrusion detection, fuzzy logic has mostly been used inside the detector. Hamamoto *et al.* [17] combined a genetic algorithm, which builds a digital signature of normal traffic, with a fuzzy-logic decision on whether an observed interval is anomalous. Neuro-fuzzy models learn fuzzy rules from data: Toosi and Kahani [18] used an ensemble of neuro-fuzzy classifiers whose outputs are combined by a fuzzy inference system tuned with a genetic algorithm, and Liu *et al.* [19] built an interpretable deep structure from ANFIS [20], decision trees and k-means. In all of these works the fuzzy component competes with, or replaces, the statistical classifier, and its output is still a detection decision.

2.4. Fuzzy risk scoring and alert prioritization

A second line of work uses fuzzy reasoning after detection, to rank or score alerts. FuzMet [15] is a fuzzy-logic alert-prioritization engine that combines alert metrics such as applicability, importance of the victim, attack severity and service/social activity into a priority score. Alali *et al.* [16] used a Mamdani fuzzy inference system to estimate cyber-security risk from vulnerability, threat and impact factors. These systems demonstrate the value of linguistic risk policies, but they do not take a calibrated ML attack probability as an input, they are not evaluated on flow-level benchmark data together with the detector that feeds them, and they do not report coverage, monotonicity or sensitivity properties of the rule base. Interpretability of linguistic fuzzy systems depends on such properties, including completeness of the rule base and distinguishable, well-covered fuzzy partitions [28,29].

2.5. Research gap and positioning

Table **1** positions the present work. Existing IDS studies optimise classification metrics, XAI studies explain the classifier after the prediction has been made, and fuzzy risk engines rank alerts without a calibrated detector probability. The gap addressed here is a transparent, verifiable transformation of calibrated classifier evidence into an operational threat level under contextual uncertainty, evaluated end-to-end on benchmark flows and against simple transparent alternatives. The claim is deliberately narrow: the contribution is the decision layer and its evaluation protocol, not a new detection algorithm.

3. MATERIALS AND METHODS

3.1. Proposed Fuzzy–AI Architecture and Data Flow

Figure **1** shows the architecture. It has a detection path, a context path and an explanation side branch. Let x denote the preprocessed feature vector of one network flow. A trained and calibrated classifier f returns p=f(x), the estimated probability that the flow is malicious; p is passed to the fuzzy layer as AP. In parallel, a context builder derives four contextual indicators (TV, CD, PA, SR) from the same flow record and from policy information. The fuzzy layer returns CTL=F(AP, PA, TV, CD, SR) ∈ [0,100], a linguistic class

Table 3: Fuzzy input and output variables.

Variable	Symbol	Range / unit	Fuzzy sets	Construction on CICIDS2017
AI Attack Probability	AP	0–1 (probability)	Low, Medium, High	XGBoost malicious-class probability after isotonic calibration (Section 3.10.4).
Protocol Anomaly	PA	0–1 (training percentile)	Low, Moderate, High	Mean robust z-score (|x−median|/(1.4826·MAD), capped at 10) over 16 TCP-flag, header, packet-length and inter-arrival fields; PA = clip((F(s) − 0.90)/0.10, 0, 1), where F is the training distribution of the score, so PA > 0 only for the 10% most deviant training flows (Section 3.8.1). Label-free.
Traffic Volume	TV	0–1000 (per-mille rank)	Low, Medium, High	Total bytes of the flow (forward + backward payload), expressed as its per-mille rank among training flows; TV = 850 means the flow carries more bytes than 85% of training flows.
Connection Duration	CD	0–300 s	Short, Medium, Long	CICIDS2017 Flow Duration (μs) divided by 10^6 and capped at 300 s.
Source Reliability	SR	0–10 (ordinal trust score)	Untrusted, Moderate, Trusted	Reputation/asset-policy score supplied by the organisation; neutral value 5.5 on CICIDS2017 (Section 3.8.3).
Cyber Threat Level	CTL	0–100 (ordinal priority score)	Very Low, Low, Moderate, High, Critical	Centroid of the aggregated output set; interpreted in Section 3.5.

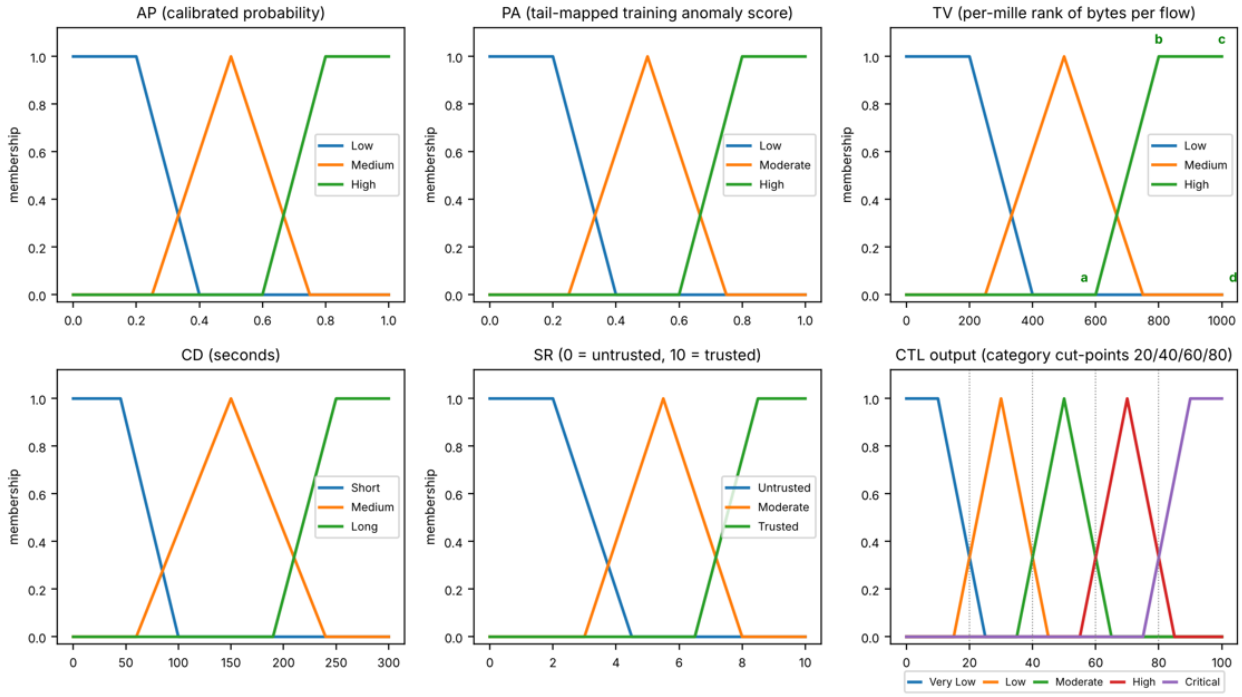


Figure 2: Membership functions of all fuzzy variables: the five inputs (AP, PA, TV, CD, SR) and the CTL output. For TV High the trapezoid parameters a, b, c, d are marked. Dotted lines in the CTL panel show the category cut-points used in Table **5**.

and a rule trace. SHAP is a side branch: it explains f but its output is not an input to the fuzzy layer. Table **2** lists, stage by stage, what each component receives and what it passes on.

Formally, p = f(x), CTL = F(TV, CD, PA, SR, p), CTL ∈ [0,100].

3.2. Input Variables and Operationalization

Table **3** defines each fuzzy variable, its unit and how it is computed. All reference statistics used to construct TV and PA are estimated on the training partition only and none of the context variables uses the class label.

3.3. Membership Functions

Triangular and trapezoidal membership functions are used because they are computationally efficient, transparent and easy to audit [28]. A triangular set A=(a,b,c) has membership $\mu A(x)=\max(\min((x-a)/(b-a),(c-x)/(c-b)),0)$: a and c are the left and right feet, where membership becomes zero, and b is the peak, the single most typical value of the concept. A trapezoidal set A=(a,b,c,d) has $\mu A(x)=\max(\min((x-a)/(b-a),1,(d-x)/(d-c)),0)$: [b,c] is the core, the interval of values that fully belong to the concept, while [a,b] and [c,d] are the transition zones in which membership rises from 0 to 1 and falls back to 0. For shoulder sets at the edge of a universe (e.g., a=b=0 for TV Low) the membership is 1 at the boundary. Figure **2** shows all six variables, with a, b, c and d marked for TV High.

The parameters in Table **4** were set by expert design before any data were examined, following three principles: (i) adjacent sets overlap so that every point of each universe belongs to at least one set with non-zero membership (complete partitions); (ii) adjacent sets cross at membership values of about 0.28–0.33, so that no input is simultaneously 'fully' in two sets; and (iii) the cores are anchored to interpretable reference points, as listed in the last column. The parameters were not optimised on test data. Their influence on the outputs is quantified by the perturbation analysis in Section 4.6.

3.4. Fuzzy Inference

The system uses Mamdani min–max inference. For rule r with antecedent memberships μ1,...,μk, firing strength is αr=min(μ1,...,μk); a negated antecedent (written NOT) uses 1−μ. The consequent membership function is clipped at αr; rule outputs are aggregated using the maximum operator. Centroid defuzzification of the piecewise-linear aggregated set on a 1001-point grid of [0,100] produces the crisp threat level:

$$CTL^* = \int y\ \mu_{agg}(y)\ dy \ / \int \mu_{agg}(y)\ dy$$

3.5. Interpreting the Cyber Threat Level

CTL is an ordinal priority score, not a probability and not an expected loss. Its value states where the centre of gravity of the activated consequents lies on the five-label output scale. Each crisp CTL is assigned the output label in which it has the largest membership; for the output sets in Table **4** this gives the cut-points 20,

Table 4: Membership-function parameters and their origin (parameters unchanged from the original submission).

Fuzzy set	Function	Parameters (a, b, c[, d])	Anchoring / origin
AP Low	trapmf	[0, 0, 0.2, 0.4]	probability thirds; High starts at 0.6 and is full above 0.8, i.e., when the calibrated detector is confident
AP Medium	trimf	[0.25, 0.5, 0.75]	
AP High	trapmf	[0.6, 0.8, 1, 1]	
PA Low	trapmf	[0, 0, 0.2, 0.4]	upper tail of the training anomaly score: PA High starts at the most deviant 4% and is full for the most deviant 2% of training flows
PA Moderate	trimf	[0.25, 0.5, 0.75]	
PA High	trapmf	[0.6, 0.8, 1, 1]	
TV Low	trapmf	[0, 0, 200, 400]	per-mille ranks: Low ≤ 20th–40th percentile, High ≥ 60th–80th percentile of training flows
TV Medium	trimf	[250, 500, 750]	
TV High	trapmf	[600, 800, 1000, 1000]	
CD Short	trapmf	[0, 0, 45, 100]	seconds: Short ≤ 45–100 s (typical request/response), Long ≥ 190–250 s (persistent sessions)
CD Medium	trimf	[60, 150, 240]	
CD Long	trapmf	[190, 250, 300, 300]	
SR Untrusted	trapmf	[0, 0, 2, 4.5]	ordinal trust scale: Untrusted ≤ 2–4.5 (deny-listed/unknown), Trusted ≥ 6.5–8.5 (allow-listed/managed asset)
SR Moderate	trimf	[3, 5.5, 8]	
SR Trusted	trapmf	[6.5, 8.5, 10, 10]	
CTL Very Low	trapmf	[0, 0, 10, 25]	five equally spaced peaks (10, 30, 50, 70, 90) with 15-point transitions
CTL Low	trimf	[15, 30, 45]	
CTL Moderate	trimf	[35, 50, 65]	
CTL High	trimf	[55, 70, 85]	
CTL Critical	trapmf	[75, 90, 100, 100]	

Table 5: CTL categories, cut-points and an example triage policy.

Category	CTL interval	Typical activated consequents	Example operational meaning
Very Low	[0, 20)	Very Low (+ Low)	log only; no analyst action
Low	[20, 40)	Low (+ Very Low/Moderate)	batch review / threat-hunting backlog
Moderate	[40, 60)	Moderate	queue for analyst review within the shift
High	[60, 80)	High (+ Moderate/Critical)	investigate promptly; consider containment
Critical	[80, 100]	Critical (+ High)	immediate response; incident-handling procedure

40, 60 and 80 (the crossover points of adjacent output sets). Because the extreme output sets are trapezoids, the attainable range of the centroid is approximately 9.3–90.7 rather than 0–100. For example, CTL = 75 has membership 0.67 in High and 0 in Critical: the event is High priority and the activated rules are mixing High with some Critical evidence, so it is closer to escalation than a CTL of 65. A CTL of exactly 50 is obtained when only Moderate rules fire. Table **5** summarises the mapping together with an example triage policy; response times are illustrative and must be set by each organisation.

3.6. Fuzzy Rule Base

Table **6** lists the complete revised rule base. The 35 original rules are retained, three of them (R21, R22, R24) are modified, and 14 rules are added. The added rules R36–R47 complete the 27-cell AP × PA × SR core of the rule base with consequents chosen to be monotone with respect to their neighbours (higher AP or PA, or lower SR, never yields a lower consequent). R21b and R22b split the former context-only Critical rules so that AP remains influential in severe contexts. The reasons for these changes are given in Section 3.7.

3.7. Rule-Base Coverage, Fallback Behaviour and the AP Floor

Coverage. With five inputs and three linguistic labels per input there are $3^5 = 243$ antecedent combinations. A combination is covered if at least one rule's antecedent is satisfied by it. The original 35 rules covered 173 of 243 combinations (71.2%). On 100,000 input vectors

drawn uniformly from the input space, no rule fired for 12.5% of them, so the centroid was undefined (0/0). The example raised by the reviewer (AP 0.10, PA 0.50, TV 500, CD 150, SR 5.5) is one such case. Because every point of each input universe has non-zero membership in at least one label (complete partitions, Figure **2**), it is sufficient that every cell of a sub-grid of variables be covered by a rule that involves only those variables. Rules R36–R47 complete the AP × PA × SR grid, which raises coverage to 243/243 (100%) and guarantees that at least one rule fires for every admissible input; no undefined output occurred in the 100,000 samples. The reviewer's example now fires R45 (AP Low ∧ PA Moderate ∧ SR Moderate → Low) and yields CTL = 30.0 (Low)

Fallback. If the rule base or the membership functions are edited in future so that no rule fires ($\Sigma\alpha = 0$), the implementation does not return an undefined value: it assigns CTL = 50 (Moderate), flags the event as 'no-rule' in the trace and routes it to manual review. With the rule base of Table **6** this branch is unreachable, and it was never triggered in the experiments.

AP floor. The reviewer showed that in the original system AP could have no effect (Scenario A: CTL ≈ 81.2 for every AP from 0.05 to 1.0) or could be overridden by a trusted-source context (Scenario E: CTL ≤ 33.7, Low, even for AP = 0.9–1.0). Two mechanisms caused this. First, the context-only rules R21 and R22 produced Critical regardless of AP. Second, the context-only de-escalation rule R24 (PA Low ∧ TV Low ∧ SR Trusted → Very Low) fired together with the AP-High rules and pulled the centroid down. The revision (i) adds 'AP NOT Low' to R21 and R22 and introduces R21b/R22b (same context with AP Low → High); (ii) adds 'AP NOT High' to R24; and (iii) adds an explicit policy guard applied after defuzzification:

$$CTL_{final} = \max(\, CTL^{*},\ 40 \cdot \mu_{AP,High}(AP)\,)$$

so that a detection with full membership in AP High (AP ≥ 0.8) is never rated below Moderate, and the bound fades linearly to zero as AP falls to 0.6. The guard encodes the policy 'a high-confidence detection is never suppressed by context alone; context can only lower it to Moderate, i.e., to analyst review'. We retain context-only escalation (R16, R21b, R22b, R32): a severe protocol anomaly from an untrusted source is still rated High when the detector is not confident, because a detector trained on a benchmark can miss novel attacks.

3.8. Context Variables on Benchmark Data

3.8.1. Protocol Anomaly (PA)

CICIDS2017 does not provide a protocol-anomaly score, so PA is computed from the flow record by a label-free estimator whose statistics come from the training partition only. For each of 16 protocol-behaviour fields (FIN, SYN, RST, PSH, ACK, URG, ECE and CWE flag counts; forward PSH and URG flags; forward and backward header length; down/up ratio; packet-length variance; minimum packet length; minimum flow inter-arrival time) the training median and the scaled median absolute deviation (1.4826·MAD; the standard deviation is used if MAD = 0) are stored. The raw anomaly score of a flow is the mean of the robust z-scores $|x-\text{median}|/\text{scale}$, each capped at 10 [36]. With F the empirical distribution function of this score on the training partition, $PA = \text{clip}((F(s) - 0.90)/0.10, 0, 1)$. PA is therefore 0 for the 90% least deviant training-like flows and rises linearly over the most deviant 10%; PA ≥ 0.6 (membership in High) corresponds to the most deviant 4% and PA ≥ 0.8 (full membership) to the most deviant 2%. The tail mapping encodes the premise that anomalous protocol behaviour is rare. An initial version used the plain percentile F(s) as PA; by construction it gives 40% of all training flows non-zero membership in PA High, and on the test flows it made the fuzzy layer rate 11% of benign flows High or Critical. The 0.90 threshold was fixed from this label-free prevalence criterion (about 5% of training flows with PA ≥ 0.5), not tuned on test performance, and the percentile variant is reported as an ablation (Supplementary Table S**3**). Destination port and initial window sizes are excluded from PA for the same leakage reasons that they are excluded from the reduced detector (Section 3.10.2). The trace reports the field with the largest z-score, which tells the analyst why PA is high.

3.8.2. Traffic Volume (TV) and Connection Duration (CD)

TV is the per-mille rank (0–1000) of the flow's total transferred bytes among training flows; CD is the flow duration in seconds, capped at 300 s. CICFlowMeter terminates flows after an activity timeout of 120 s, so on CICIDS2017 CD never exceeds about 120 s and the CD Long set is inactive in the benchmark evaluation; it is retained because long-lived sessions are relevant in deployment. This limitation is consistent with the low Sobol index of CD (Section 4.6).

3.8.3. Source Reliability (SR) policy

CICIDS2017 MachineLearningCSV contains no source identifiers and no external reputation score. Deriving SR from attacker IP addresses known to the dataset creators would leak the label. The explicit SR policy for all benchmark experiments is therefore SR = 5.5 (full membership in Moderate) for every flow. With this policy only the rules whose SR antecedent is Moderate, or that do not involve SR, can fire, and the effect of SR is studied only in the controlled scenarios and the sensitivity analysis. In deployment SR can be supplied by threat-intelligence reputation, allow/deny lists, asset identity or prior incident history.

Table 6: Revised Mamdani rule base (49 rules).

Rule	Antecedent	Consequent CTL	Status
R1	AP High AND PA High AND SR Untrusted	Critical	original
R2	AP High AND PA High AND SR Moderate	Critical	original
R3	AP High AND PA Moderate AND SR Untrusted	High	original
R4	AP High AND PA Moderate AND SR Moderate	High	original
R5	AP High AND PA Low AND SR Trusted	Moderate	original
R6	AP High AND TV High AND SR Untrusted	Critical	original
R7	AP High AND CD Long AND PA High	Critical	original
R8	AP High AND TV Medium AND PA Moderate	High	original
R9	AP Medium AND PA High AND SR Untrusted	High	original
R10	AP Medium AND PA High AND TV High	High	original
R11	AP Medium AND PA Moderate AND SR Moderate	Moderate	original
R12	AP Medium AND PA Low AND SR Trusted	Low	original
R13	AP Medium AND TV High AND SR Untrusted	High	original
R14	AP Medium AND CD Long AND PA High	High	original
R15	AP Low AND PA High AND SR Untrusted	High	original
R16	AP Low AND PA High AND TV High	High	original
R17	AP Low AND PA Moderate AND SR Untrusted	Moderate	original
R18	AP Low AND PA Low AND SR Trusted	Very Low	original
R19	AP Low AND PA Low AND SR Moderate	Low	original
R20	AP Low AND TV Low AND SR Trusted	Very Low	original
R21	PA High AND TV High AND SR Untrusted AND AP NOT Low	Critical	modified
R22	PA High AND CD Long AND SR Untrusted AND AP NOT Low	Critical	modified
R23	PA Moderate AND TV High AND SR Untrusted	High	original
R24	PA Low AND TV Low AND SR Trusted AND AP NOT High	Very Low	modified
R25	PA Moderate AND SR Trusted AND AP Low	Low	original
R26	TV High AND CD Long AND AP High	Critical	original
R27	TV High AND CD Long AND AP Medium	High	original
R28	TV Medium AND PA Moderate AND AP Medium	Moderate	original
R29	TV Low AND PA Low AND AP Medium AND SR Trusted	Low	original
R30	TV Low AND CD Short AND PA Low AND AP Low	Very Low	original
R31	SR Untrusted AND AP High	High	original
R32	SR Untrusted AND PA High	High	original
R33	SR Trusted AND PA Low AND AP High	Moderate	original
R34	SR Trusted AND PA Moderate AND AP Medium	Moderate	original
R35	SR Moderate AND PA Low AND AP Low	Low	original

(Table 6). Continued.

Rule	Antecedent	Consequent CTL	Status
R21b	PA High AND TV High AND SR Untrusted AND AP Low	High	new
R22b	PA High AND CD Long AND SR Untrusted AND AP Low	High	new
R36	AP High AND PA Low AND SR Moderate	High	new
R37	AP High AND PA Moderate AND SR Trusted	Moderate	new
R38	AP High AND PA High AND SR Trusted	High	new
R39	AP Medium AND PA Low AND SR Untrusted	Moderate	new
R40	AP Medium AND PA Low AND SR Moderate	Low	new
R41	AP Medium AND PA Moderate AND SR Untrusted	High	new
R42	AP Medium AND PA High AND SR Moderate	High	new
R43	AP Medium AND PA High AND SR Trusted	Moderate	new
R44	AP Low AND PA Low AND SR Untrusted	Low	new
R45	AP Low AND PA Moderate AND SR Moderate	Low	new
R46	AP Low AND PA High AND SR Moderate	Moderate	new
R47	AP Low AND PA High AND SR Trusted	Low	new

3.9. Transparent Baselines

To test whether a fuzzy system is needed at all, two transparent scores are computed from exactly the same five inputs. (i) Weighted score: CTL_w = 100·(0.35·AP + 0.25·PA + 0.20·(1−SR/10) + 0.10·TV/1000 + 0.10·CD/300). The weights were fixed a priori to mirror the importance order implied by the rule base (AP > PA > SR > TV = CD) and were not tuned. (ii) Crisp lookup table: each input is assigned the single label with the largest membership, the most severe consequent among the rules of Table **6** that match these crisp labels is selected, and its category centre (10, 30, 50, 70 or 90) is returned. The lookup table therefore uses the same expert policy as the fuzzy engine but without graded membership. AP alone (100·AP) is reported as a third reference.

3.10. Experimental Design

3.10.1. Dataset

The experimental protocol used CICIDS2017 from the Canadian Institute for Cybersecurity at the University of New Brunswick. The eight MachineLearningCSV files were processed using a deterministic file-wise random sampling procedure in which 10% of each 100,000-row chunk was retained with random seed 42, preserving representation from every capture file. After removal of non-finite values and duplicates and exclusion of constant columns, the experimental matrix contained 269,091 flows (220,534 benign, 48,557 attack) and 70 features. The binary target encoded BENIGN as 0 and all attack labels as 1. The capture file and the original row order of each flow were retained to construct the temporal hold-outs. The re-implemented pipeline removes rows with non-finite values before de-duplication, which yields slightly fewer flows than the 271,394 reported in the original submission.

3.10.2. Leakage-aware preprocessing and feature sets

- All scalers, calibrators, PA/TV reference statistics and SHAP background samples are fitted on the training partition after the split; nothing is fitted on test data.
- Full feature set: the 70 features used in the original submission.
- Reduced feature set (67 features): Destination Port, Init_Win_bytes_forward and Init_Win_bytes_backward are removed. Destination port identifies the targeted service rather than malicious behaviour, and the initial TCP window sizes are largely determined by the operating systems of the attacker and victim machines in the testbed; both can be learned as shortcuts that do not transfer to other networks [21–23]. The reduced set is the primary configuration in this revision.
- Class imbalance is handled with class weights computed on the training data only.
- 3.10.3. Data splits

Table 7: Performance metrics used in model evaluation.

Metric	Definition
Accuracy	(TP+TN)/(TP+TN+FP+FN)
Precision	TP/(TP+FP)
Recall / TPR	TP/(TP+FN)
F1-score	2·Precision·Recall/(Precision+Recall)
False-positive rate	FP/(FP+TN)
Balanced accuracy	(TPR+TNR)/2
ROC-AUC, PR-AUC	areas under the ROC and precision–recall curves
Brier score	mean squared error of the predicted probability against the binary outcome
Expected calibration error (ECE)	Σb (nb/n)·|observed attack rate in bin b − mean predicted probability in bin b|, 15 equal-width bins [32]
Triage metrics (fuzzy layer)	share of attack and of benign flows rated High/Critical and ≥ Moderate; precision among the top 1% and 5% of flows ranked by score

Table 8: Dataset profile after preprocessing and sizes of the evaluation partitions.

Statistic	Value
Total cleaned flows	269,091
Benign flows	220,534
Attack flows	48,557
Features (full / reduced set)	70 / 67
Test flows (attacks): random 80/20	53,819 (9,712)
Test flows (attacks): T1 within-file chronological	53,822 (3,975)
Test flows (attacks): T2 leave-Friday-out	67,491 (27,414)

- Random: stratified 80/20 split (as in the original submission), repeated for five seeds (42, 7, 123, 2024, 31337).
- Temporal T1 (within-file chronological): in each capture file the first 80% of flows in recorded order form the training set and the last 20% the test set. The MachineLearningCSV files carry no timestamp, so record order is used as the time proxy.
- Temporal T2 (leave-Friday-out): training on Monday–Thursday, testing on all Friday traffic (botnet, port scan and DDoS). This is a deliberately harsh test because most Friday attack types are absent from training.
- For T1 and T2 the partition is fixed and the five seeds vary model initialisation and the calibration subset.

3.10.4. Machine-learning models and probability calibration

Three tabular classifiers were evaluated: Random Forest [8], XGBoost [9], and a multilayer perceptron (MLP) optimized with Adam [10]. Random Forest used 150 trees, square-root feature subsampling, balanced subsample class weights. XGBoost used 250 trees, maximum depth 6, learning rate 0.10, subsample 0.85, column subsample 0.85, and class-imbalance weighting derived from the training split. The MLP used two hidden layers of 96 and 48 units, standardized inputs, early stopping, and a 20-iteration cap; because the optimizer had not fully converged at the cap, its result is interpreted as a neural baseline rather than an optimized upper bound. The implementation used scikit-learn [11] and XGBoost [9].

The original submission described AP as 'calibrated' without naming a method; this has been corrected. In the revision, 20% of each training partition (stratified) is held out as a calibration subset; models are trained on the remaining 80%, and an isotonic regression [31] mapping raw scores to probabilities is fitted on the calibration subset and applied to the test scores. Isotonic regression was chosen over Platt scaling because it makes no parametric assumption about the shape of the miscalibration [30]. The calibrated XGBoost probability is the AP passed to the fuzzy layer.

3.10.5. Evaluation metrics

All detector metrics use a 0.5 threshold and are reported as mean ± SD over the five seeds.

3.10.6. Evaluation of the fuzzy layer on test flows

For every test flow the calibrated XGBoost probability (reduced features) is used as AP and TV, CD and PA are computed with the training-only estimators of Section 3.8, with SR = 5.5. The original and revised rule bases, the two baselines and AP alone are applied

Table 9: Predictive performance under the stratified random split (mean ± SD over five seeds; raw probabilities, threshold 0.5).

Model (features)	Accuracy	Precision	Recall	F1	ROC-AUC	PR-AUC	FPR
RandomForest (full)	0.9982 ± 0.0003	0.9963 ± 0.0001	0.9938 ± 0.0016	0.9950 ± 0.0008	0.9997 ± 0.0001	0.9982 ± 0.0005	0.0008 ± 0.0000
XGBoost (full)	0.9990 ± 0.0001	0.9954 ± 0.0004	0.9989 ± 0.0003	0.9972 ± 0.0002	1.0000 ± 0.0000	0.9998 ± 0.0000	0.0010 ± 0.0001
MLP (full)	0.9833 ± 0.0006	0.9577 ± 0.0060	0.9493 ± 0.0064	0.9535 ± 0.0017	0.9983 ± 0.0002	0.9929 ± 0.0004	0.0092 ± 0.0014
RandomForest (reduced)	0.9957 ± 0.0006	0.9890 ± 0.0022	0.9873 ± 0.0016	0.9881 ± 0.0015	0.9975 ± 0.0002	0.9949 ± 0.0007	0.0024 ± 0.0005
XGBoost (reduced)	0.9955 ± 0.0005	0.9844 ± 0.0023	0.9907 ± 0.0016	0.9875 ± 0.0014	0.9996 ± 0.0000	0.9987 ± 0.0001	0.0035 ± 0.0005
MLP (reduced)	0.9831 ± 0.0009	0.9588 ± 0.0133	0.9471 ± 0.0116	0.9528 ± 0.0024	0.9972 ± 0.0004	0.9902 ± 0.0012	0.0090 ± 0.0031

to the same inputs. Because the fuzzy layer is a prioritisation policy rather than a classifier, the main outputs are the distribution of CTL categories by true class, the triage metrics of Table **7**, the frequency with which each rule fires, the proportion of flows on which the original rule base fires no rule, and the proportion on which the AP floor is binding. ROC-AUC and PR-AUC of the scores are reported for completeness.

3.10.7. Sensitivity analyses of the decision layer

- AP sweeps: AP is varied from 0.05 to 1.0 with the other inputs fixed at each scenario's values, and over 3,000 random contexts to quantify how often AP has little effect (range < 5 CTL points) and how often a confident detection (AP ≥ 0.8) falls below Moderate.
- Global variance-based sensitivity: first-order (S1) and total-order (ST) Sobol indices [33] computed with SALib [34] using 12,288 model evaluations over the full input space.
- Membership-function robustness: every interior breakpoint is perturbed by a uniform random amount of up to ±5% and ±10% of the variable's range (300 draws each) and the scenario CTLs are recomputed.

3.10.8. Explainability evaluation

Global SHAP importance is summarised by mean absolute SHAP value on 3,000 test flows, using interventional TreeSHAP with a background of 100 training flows (sub-sampled by the SHAP library from 1,000 drawn at random). Local explanations are reported for one representative true-positive, false-positive, true-negative and false-negative flow (the flow of each group whose predicted probability is closest to the group median). For the same flows, the fuzzy explanation consists of the input memberships, the most strongly fired rules and the defuzzified CTL, which yields a traceable path from flow features to attack probability and from attack probability plus context to operational threat. Complementary interpretation methods include accumulated local effects [12] and

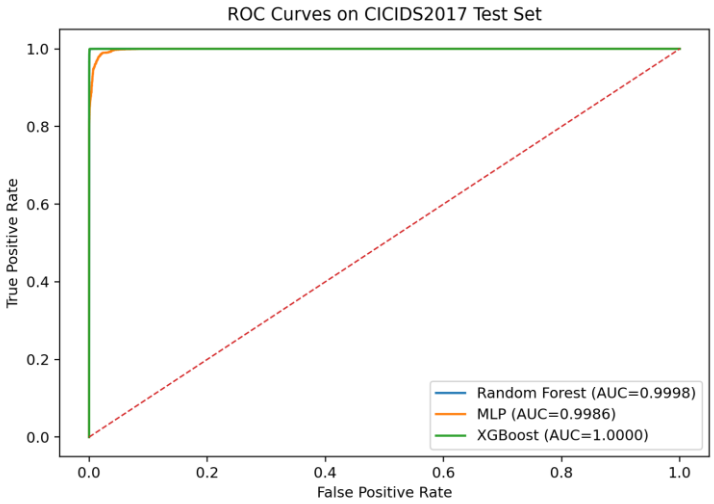


Figure 3: Receiver operating characteristic curves of the evaluated classifiers (original 70-feature configuration, random split, seed 42).

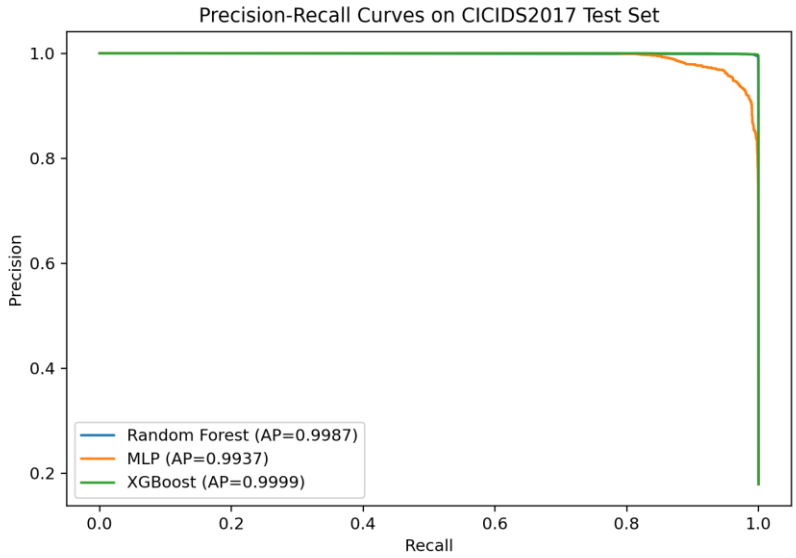


Figure 4: Precision–recall curves of the evaluated classifiers (original 70-feature configuration, random split, seed 42).

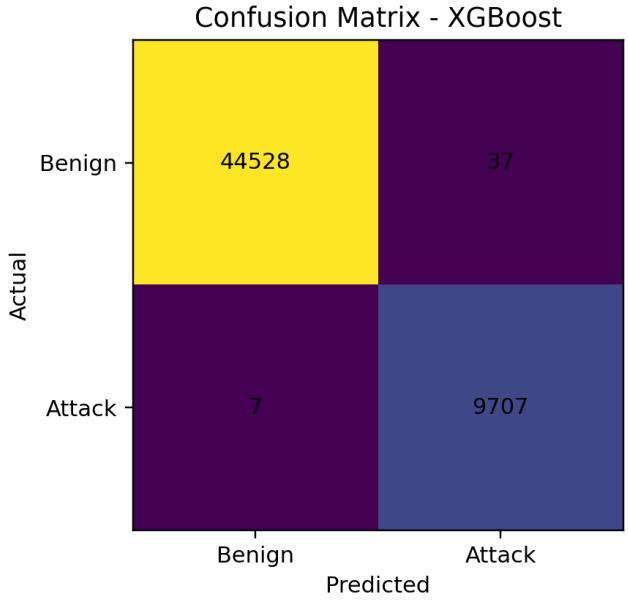


Figure 5: Confusion matrix of XGBoost (original 70-feature configuration, random split, seed 42).

LIME [13]; broader interpretability practice is summarised by Molnar [14].

4. RESULTS

4.1 Dataset Profile

4.2. Predictive Performance under the Random Split

Table **9** compares the full and reduced feature sets under the stratified random split (mean ± SD over five seeds). With the full feature set XGBoost reached F1 = 0.9972 ± 0.0002; removing destination port and initial window sizes gave F1 = 0.9875 ± 0.0014, ROC-AUC = 0.99965 ± 0.00002 and PR-AUC = 0.99871 ± 0.00011. Figures **3**–**5** are retained from the original submission and refer to the full-feature, seed-42 configuration.

4.3. Temporal Hold-outs

Table **10** reports the leakage-reduced detectors under the two temporal hold-outs. XGBoost obtained F1 = 0.9691 ± 0.0014 under the within-file chronological split (T1) and F1 = 0.4478 ± 0.0185 with recall 0.2912 ± 0.0152 and PR-AUC = 0.6842 ± 0.0241 under leave-Friday-out (T2). The gap between these values and the random-split result is the main reason why the near-perfect random-split figures of the original submission are no longer presented as evidence of generalisation.

4.4. Probability Calibration

Because AP is consumed as semantic evidence by the fuzzy layer, its calibration matters. For XGBoost on the random split, isotonic calibration changed the Brier

Table 10: Performance of the leakage-reduced detectors under temporal hold-outs (mean ± SD over five seeds).

Split: model (features)	Accuracy	Precision	Recall	F1	ROC-AUC	PR-AUC	FPR
T1: RandomForest (reduced)	0.9941 ± 0.0007	0.9747 ± 0.0018	0.9439 ± 0.0083	0.9591 ± 0.0046	0.9953 ± 0.0004	0.9803 ± 0.0011	0.0019 ± 0.0001
T1: XGBoost (reduced)	0.9954 ± 0.0002	0.9667 ± 0.0018	0.9714 ± 0.0032	0.9691 ± 0.0014	0.9985 ± 0.0001	0.9902 ± 0.0008	0.0027 ± 0.0002
T1: MLP (reduced)	0.9890 ± 0.0023	0.9526 ± 0.0151	0.8959 ± 0.0362	0.9230 ± 0.0170	0.9916 ± 0.0013	0.9592 ± 0.0027	0.0036 ± 0.0012
T2: RandomForest (reduced)	0.6908 ± 0.0391	0.9745 ± 0.0206	0.2435 ± 0.0963	0.3816 ± 0.1390	0.6799 ± 0.0109	0.6239 ± 0.0116	0.0033 ± 0.0003
T2: XGBoost (reduced)	0.7085 ± 0.0062	0.9704 ± 0.0022	0.2912 ± 0.0152	0.4478 ± 0.0185	0.7059 ± 0.0423	0.6842 ± 0.0241	0.0061 ± 0.0003
T2: MLP (reduced)	0.7105 ± 0.0020	0.9793 ± 0.0195	0.2935 ± 0.0021	0.4516 ± 0.0019	0.7987 ± 0.0166	0.7290 ± 0.0140	0.0043 ± 0.0041

Table 11: Calibration of XGBoost probabilities before and after isotonic calibration (mean ± SD over five seeds).

Split: model (reduced)	Brier raw	Brier isotonic	ECE raw	ECE isotonic	F1 isotonic
Random: XGBoost	0.00375 ± 0.00030	0.00294 ± 0.00028	0.0052 ± 0.0005	0.0008 ± 0.0002	0.9907 ± 0.0011
Random: RandomForest	0.00372 ± 0.00033	0.00334 ± 0.00035	0.0035 ± 0.0004	0.0008 ± 0.0002	0.9893 ± 0.0012
Random: MLP	0.01402 ± 0.00096	0.01243 ± 0.00104	0.0100 ± 0.0017	0.0016 ± 0.0003	0.9545 ± 0.0026
T1: XGBoost	0.00448 ± 0.00022	0.00375 ± 0.00033	0.0092 ± 0.0005	0.0033 ± 0.0008	0.9675 ± 0.0058
T1: RandomForest	0.00563 ± 0.00026	0.00515 ± 0.00034	0.0062 ± 0.0003	0.0018 ± 0.0006	0.9563 ± 0.0023
T1: MLP	0.00973 ± 0.00144	0.00900 ± 0.00106	0.0088 ± 0.0014	0.0054 ± 0.0019	0.9252 ± 0.0184
T2: XGBoost	0.29121 ± 0.00448	0.31380 ± 0.01774	0.2992 ± 0.0074	0.3304 ± 0.0183	0.3074 ± 0.1123
T2: RandomForest	0.31215 ± 0.00552	0.33350 ± 0.00406	0.3400 ± 0.0053	0.3600 ± 0.0034	0.0004 ± 0.0000
T2: MLP	0.28864 ± 0.00202	0.28852 ± 0.00201	0.2902 ± 0.0020	0.2891 ± 0.0018	0.4529 ± 0.0024

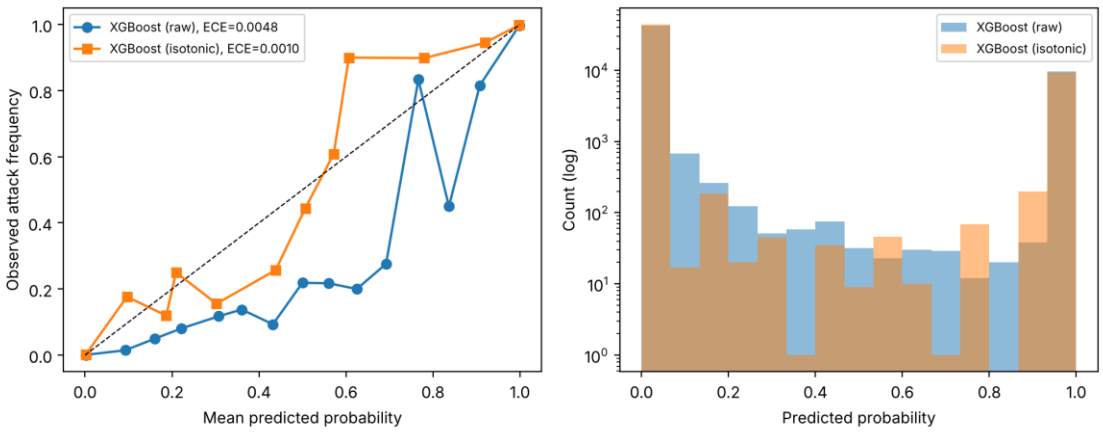


Figure 6: Reliability diagram (left) and probability histogram (right) of XGBoost on the random-split test set, raw and after isotonic calibration (reduced features, seed 42).

score from 0.00375 ± 0.00030 to 0.00294 ± 0.00028 and the ECE from 0.0052 ± 0.0005 to 0.0008 ± 0.0002 (Table **11**). The reliability diagram (Figure **6**) shows observed attack frequency against mean predicted probability; most probabilities lie near 0 or 1, as the histogram panel shows. Calibration did not transfer under distribution shift: under leave-Friday-out, isotonic regression fitted on Monday–Thursday data left the ECE at about 0.3 or made it worse, so a calibrated AP is reliable only while the traffic resembles the training traffic.

4.5. SHAP Analysis

In the original 70-feature model (Figure **7**), Init_Win_bytes_forward had the largest mean absolute SHAP magnitude (1.910), followed by Destination Port

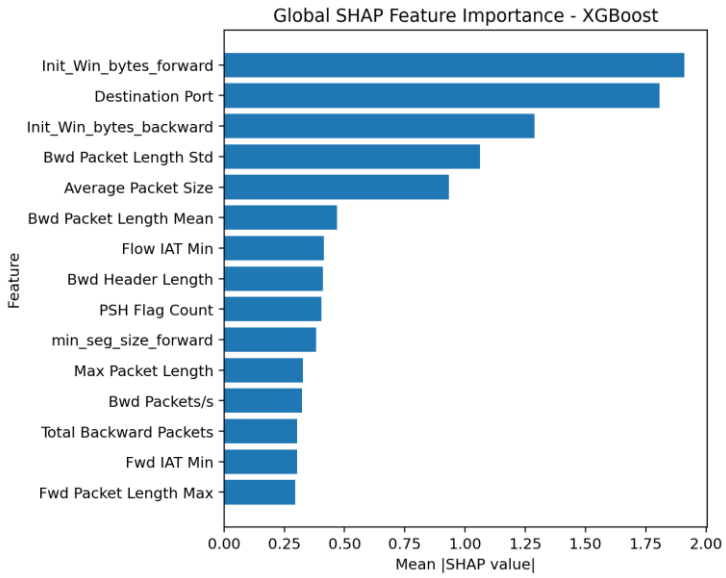


Figure 7: Global SHAP feature importance of the original 70-feature XGBoost detector (seed 42).

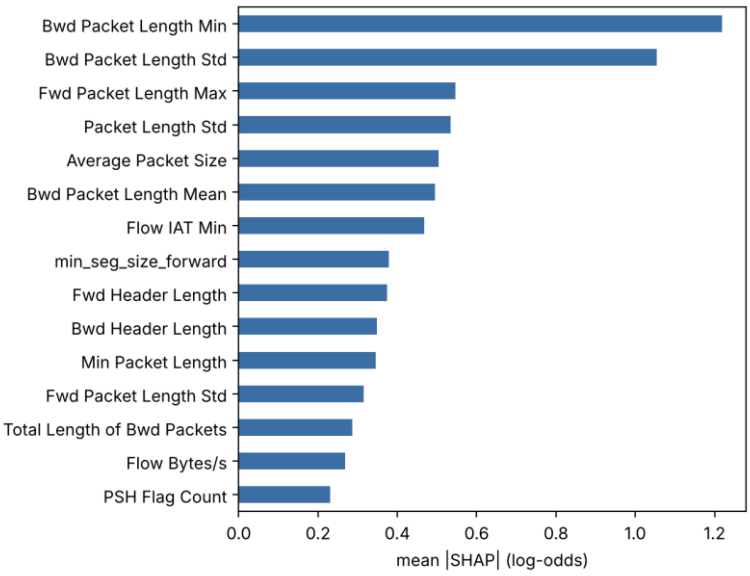


Figure 8: Global SHAP feature importance of the leakage-reduced XGBoost detector (seed 42).

(1.807), Init_Win_bytes_backward (1.289), Bwd Packet Length Std (1.063) and Average Packet Size (0.933). The dominance of these fields is what motivated the reduced feature set. After their removal (Figure **8**), the five strongest contributors were Bwd Packet Length Min, Bwd Packet Length Std, Fwd Packet Length Max, Packet Length Std, Average Packet Size. These attributions explain the detector's learned decision surface rather than causal mechanisms [2].

Figure **9** and Table **12** show local explanations for representative true-positive, false-positive, true-negative and false-negative flows together with the fuzzy rule trace of the same flows. The pairing shows

Table 12: Fuzzy inputs, most strongly fired rules and CTL for the flows in Figure 9.

Case	AP	Context inputs (main PA field)	Strongest rules (α)	CTL	Category
TP	1.000	PA 0.00 (Packet Length Variance); TV 899; CD 99.4 s; SR 5.5	R36→High (1.00)	70.0	High
FP	0.787	PA 0.00 (ACK Flag Count); TV 108; CD 0.0 s; SR 5.5	R36→High (0.94)	70.0	High
TN	0.000	PA 0.00 (Min Packet Length); TV 399; CD 0.0 s; SR 5.5	R35→Low (1.00); R19→Low (1.00); R30→Very Low (0.01)	29.8	Low
FN	0.041	PA 0.00 (Flow IAT Min); TV 108; CD 0.0 s; SR 5.5	R35→Low (1.00); R30→Very Low (1.00); R19→Low (1.00)	18.8	Very Low

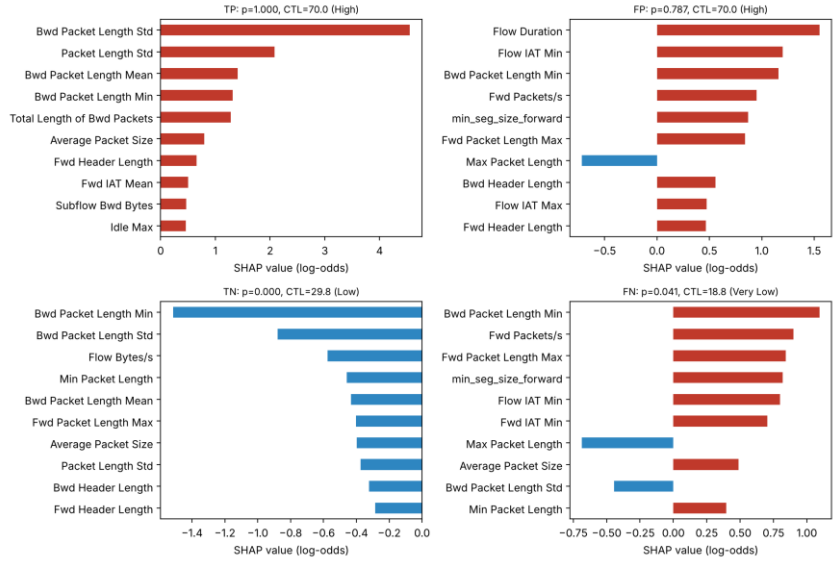


Figure 9: Local SHAP explanations (top 10 features, log-odds) for representative TP, FP, TN and FN test flows; titles give the calibrated probability and the resulting CTL.

Table 13: Controlled scenarios: CTL (category) under the original and revised rule bases and the two baselines.

Scenario	AP / PA / TV / CD / SR	Original	Revised	Weighted	Lookup	Strongest revised rules (α)
A: severe context	0.82 / 0.91 / 850 / 210 / 2	81.22 (Critical)	81.22 (Critical)	83.0 (Critical)	90 (Critical)	R6→Critical (1.00); R32→High (1.00); R31→High (1.00); R21→Critical (1.00)
B: benign context	0.82 / 0.18 / 250 / 60 / 9	30.30 (Low)	50.00 (Moderate)	39.7 (Low)	50 (Moderate)	R5→Moderate (1.00); R33→Moderate (1.00)
C: AI uncertain, context severe	0.5 / 0.9 / 900 / 250 / 1.5	81.22 (Critical)	81.22 (Critical)	74.3 (High)	90 (Critical)	R9→High (1.00); R32→High (1.00); R27→High (1.00); R22→Critical (1.00)
D: AI low, context benign	0.15 / 0.1 / 150 / 25 / 9.5	9.29 (Very Low)	9.29 (Very Low)	11.1 (Very Low)	10 (Very Low)	R30→Very Low (1.00); R24→Very Low (1.00); R20→Very Low (1.00); R18→Very Low (1.00)
E: AI high, trusted source	0.9 / 0.15 / 300 / 55 / 9	33.72 (Low)	50.00 (Moderate)	42.1 (Moderate)	50 (Moderate)	R5→Moderate (1.00); R33→Moderate (1.00)
F: AI high, trusted, quiet	0.95 / 0.1 / 100 / 5 / 9.5	28.08 (Low)	50.00 (Moderate)	37.9 (Low)	50 (Moderate)	R5→Moderate (1.00); R33→Moderate (1.00)
G: AI low, severe anomaly, untrusted	0.05 / 0.95 / 950 / 280 / 0.5	81.22 (Critical)	70.00 (High)	63.3 (High)	70 (High)	R32→High (1.00); R22b→High (1.00); R21b→High (1.00); R16→High (1.00)
H: reviewer example (no rule in original)	0.1 / 0.5 / 500 / 150 / 5.5	undefined	30.00 (Low)	35.0 (Low)	30 (Low)	R45→Low (1.00)

the two levels of explanation side by side: SHAP explains why the detector assigned its probability, and the fired rules explain why that probability, combined with PA, TV, CD and SR, led to the reported CTL.

4.6. Decision-Layer Analysis under Controlled Inputs

Table **13** reports the five original scenarios and three additional boundary cases under the original and revised rule bases and the two baselines. The original values (81.22, 30.30, 81.22, 9.29, 33.72) are reproduced exactly by the revised implementation when the original rules are used.

With the revised rule base the two AP = 0.82 events still differ strongly: severe context gives 81.22 (Critical) and benign context 50.00 (Moderate). The separation is smaller than in the original submission (31.2 instead of 50.9 points) because the AP floor now prevents a

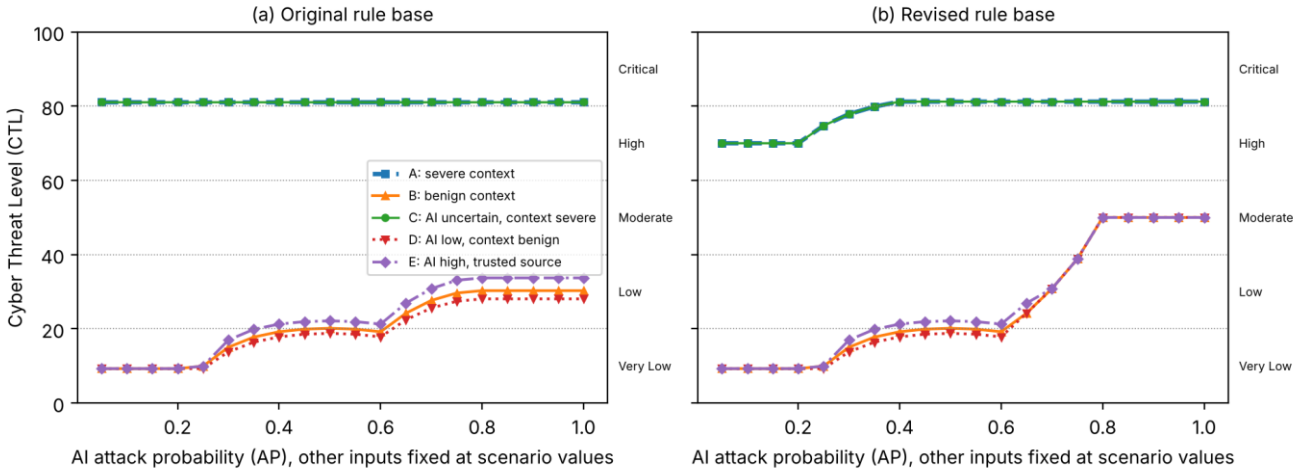


Figure 10: CTL as a function of AP with the other inputs fixed at the values of Scenarios A–E: (a) original and (b) revised rule base. Scenarios A and C coincide. Dotted lines are the category cut-points.

Table 14: Coverage and influence of AP: original vs. revised decision layer and baselines (3,000 random contexts; AP grid 0–1 in steps of 0.05).

Property	Original	Revised	Weighted	Lookup
Antecedent combinations covered (of 243)	173 (71.2%)	243 (100%)	n/a	243 (100%)
Uniform input samples with no rule fired	12.5%	0%	n/a	0%
Contexts with an undefined CTL somewhere on the AP grid	32.7%	0%	0%	0%
Contexts where AP changes CTL by < 5 points	9.2%	0.0%	0.0%	2.3%
Median CTL range spanned by AP	20.4	38.3	35.0	40.0
Contexts where AP ≥ 0.8 gives CTL < 40	12.9%	0.0%	2.1%	0.0%
Largest CTL change for ΔAP = 0.01	–	4.28	0.35	40

Table 15: Sobol sensitivity indices of the revised decision layer (S1 first-order, ST total-order) and of the weighted baseline.

Input	Revised S1	Revised ST	Weighted S1 (= ST)
AP	0.437	0.524	0.500
PA	0.241	0.320	0.255
SR	0.166	0.247	0.163
TV	0.018	0.051	0.041
CD	0.013	0.019	0.041

confident detection from being rated Low. Scenario E rises from 33.72 (Low) to 50.00 (Moderate) for the same reason. Scenario G shows the converse behaviour: severe anomaly from an untrusted source with AP = 0.05 is rated High (70.0) rather than Critical, so context-only escalation remains possible but full escalation requires some detector support. Scenario H, for which the original system was undefined, is now rated Low (30.0).

Figure **10** shows the AP sweeps. In the original system Scenario A stayed at 81.22 for every AP and Scenario E never exceeded 33.72. In the revised system Scenario A moves from 70.0 (High) at AP = 0.05 to 81.2 (Critical) for AP ≥ 0.4, and Scenario E moves from 9.3 (Very Low) to 50.0 (Moderate) for AP ≥ 0.8.

Table **14** quantifies the same effects over random contexts. The revised layer is defined everywhere, AP changes the CTL by at least 5 points in every context, and no confident detection falls below Moderate. The CTL is not strictly monotone in AP: in 17.1% of contexts it decreases by more than 0.5 points somewhere on the AP grid, with a largest decrease of 2.8 points. These small dips occur where AP Low and AP Medium overlap and are a known property of max–centroid aggregation. The weighted baseline is monotone and smooth by construction but allows a confident detection to fall

Table 16: Robustness of scenario CTLs to random perturbation of all input membership-function breakpoints (300 draws per level).

Scenario	Nominal CTL	±5%: mean ± SD	±5%: same category	±10%: mean ± SD	±10%: same category
A: severe context	81.22	81.02 ± 0.25	100.0%	80.84 ± 0.43	95.7%
B: benign context	50.00	48.84 ± 2.02	100.0%	46.80 ± 4.44	88.7%
C: AI uncertain, context severe	81.22	81.22 ± 0.00	100.0%	81.12 ± 0.19	100.0%
D: AI low, context benign	9.29	9.29 ± 0.00	100.0%	9.30 ± 0.05	100.0%
E: AI high, trusted source	50.00	50.00 ± 0.00	100.0%	50.00 ± 0.00	100.0%

Table 17: Decision layer on all test flows of the random split and the two temporal hold-outs (mean ± SD over five seeds; SR = 5.5 for all flows; undefined CTL treated as 0).

Split: score	Attacks High/Crit.	Benign High/Crit.	Attacks ≥ Mod.	Benign ≥ Mod.	Prec. top 1%	ROC-AUC	PR-AUC
Random: AP alone (100·AP)	98.63% ± 0.18	0.11% ± 0.02	98.77% ± 0.20	0.15% ± 0.03	99.96% ± 0.08	0.9996 ± 0.0001	0.9982 ± 0.0004
Random: Original fuzzy CTL	1.85% ± 0.07	4.49% ± 0.07	1.86% ± 0.07	4.49% ± 0.07	13.53% ± 2.85	0.0355 ± 0.0010	0.1815 ± 0.0008
Random: Revised fuzzy CTL	98.60% ± 0.17	1.54% ± 0.14	98.64% ± 0.18	4.93% ± 0.07	99.63% ± 0.23	0.9889 ± 0.0014	0.9849 ± 0.0016
Random: Weighted score	2.08% ± 0.12	0.00% ± 0.00	98.26% ± 0.26	2.47% ± 0.04	100.00% ± 0.00	0.9919 ± 0.0006	0.9849 ± 0.0008
Random: Lookup table	98.62% ± 0.18	3.82% ± 0.15	98.63% ± 0.18	4.20% ± 0.15	85.17% ± 1.32	0.9741 ± 0.0005	0.8418 ± 0.0038
T1: AP alone (100·AP)	94.64% ± 1.19	0.10% ± 0.01	95.41% ± 1.05	0.12% ± 0.01	99.59% ± 0.28	0.9981 ± 0.0001	0.9885 ± 0.0009
T1: Revised fuzzy CTL	93.90% ± 0.49	0.74% ± 0.13	94.52% ± 0.88	2.08% ± 0.00	94.94% ± 0.52	0.9650 ± 0.0078	0.9431 ± 0.0073
T1: Weighted score	1.27% ± 0.33	0.00% ± 0.00	93.83% ± 0.32	1.18% ± 0.01	99.81% ± 0.00	0.9852 ± 0.0008	0.9556 ± 0.0025
T2: AP alone (100·AP)	16.49% ± 6.01	0.06% ± 0.01	21.04% ± 8.74	0.09% ± 0.02	98.37% ± 0.76	0.6479 ± 0.0325	0.6067 ± 0.0304
T2: Revised fuzzy CTL	15.51% ± 6.91	1.50% ± 0.01	16.43% ± 6.00	4.63% ± 0.01	97.60% ± 0.24	0.3002 ± 0.0257	0.4556 ± 0.0492
T2: Weighted score	0.00% ± 0.00	0.00% ± 0.00	16.41% ± 6.00	2.47% ± 0.01	99.94% ± 0.08	0.3388 ± 0.0064	0.4711 ± 0.0298

below Moderate in 2.1% of contexts (e.g., Scenario F: 37.9, Low), and it cannot represent interactions such as 'trusted source AND no anomaly'. The lookup table implements the same policy as the fuzzy engine but returns only five values and jumps by up to 40 CTL points when AP crosses a label boundary by 0.01, compared with at most 4.3 points for the fuzzy engine. Over 5,000 random inputs, the Spearman correlation with the revised fuzzy CTL was 0.917 for the weighted score and 0.937 for the lookup table, with category agreement of 60.0% and 80.4%, respectively.

Table **15** shows that AP accounts for the largest share of CTL variance (S1 = 0.44, ST = 0.52), followed by PA (0.32) and SR (0.25); TV and CD contribute little (ST ≤ 0.05). The difference between ST and S1 (about 0.29 in total) is the share of variance due to interactions, which an additive score cannot represent.

Table **16** shows that ±5% perturbations never changed a scenario's category and ±10% perturbations changed it only for Scenario B (11.3% of draws) and Scenario A (4.3%), whose inputs lie close to set boundaries.

4.7. Decision Layer on the Full Test Set

Table **17** summarises the decision layer applied to every test flow of the random split with the leakage-reduced calibrated detector (mean ± SD over five seeds), and Table **18** gives the CTL categories by true class for seed 42. On the test flows the original rule base fired no rule for 20.03% ± 0.10 of flows, which the revised rule base eliminates; the AP floor was binding for 0.00% ± 0.00 of flows. The revised layer rated 98.60% ± 0.17 of attack flows and 1.54% ± 0.14 of benign flows High or Critical (AP alone with the same

Table 18: CTL category by true class on the random-split test set (revised rule base, seed 42).

CTL category	Benign flows	Attack flows	Attack share in category
Very Low	5,643	64	1.12%
Low	36,336	82	0.23%
Moderate	1,470	5	0.34%
High	658	9,524	93.54%
Critical	0	37	100.00%

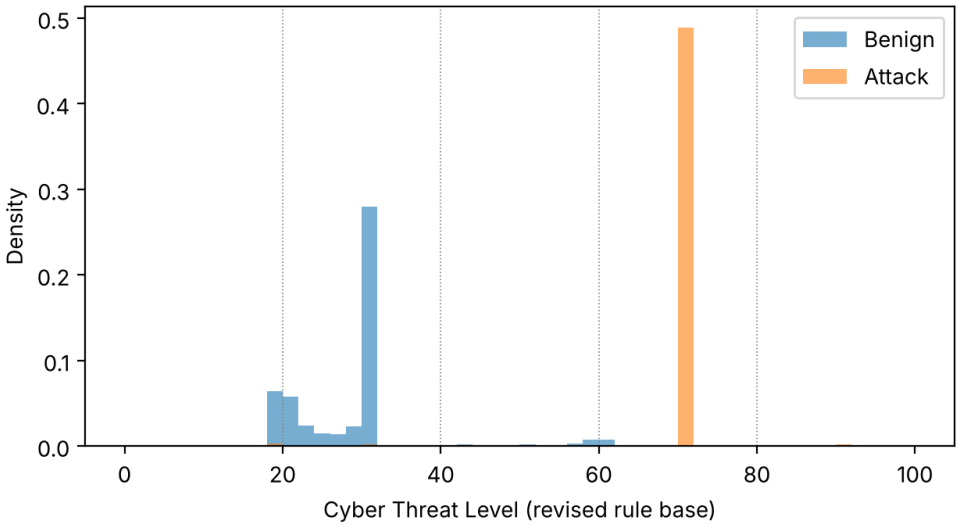


Figure 11: Distribution of the revised CTL for benign and attack test flows (random split, seed 42). Dotted lines are the category cut-points.

cut-points: 98.63% ± 0.18 and 0.11% ± 0.02). Of the 125.4 (mean per seed) attack flows missed by the detector (AP < 0.5), 1.58% ± 0.73 were still rated Moderate or higher because of their protocol and volume context. As a ranking score, the CTL had ROC-AUC 0.9889 ± 0.0014, compared with 0.9996 ± 0.0001 for AP alone, 0.9919 ± 0.0006 for the weighted score and 0.9741 ± 0.0005 for the lookup table. Table **17** also reports the temporal hold-outs.

Three further observations follow from Table **17** and the diagnostics. First, the original 35-rule base was undefined for about one fifth of random-split test flows; most of them are confidently detected attacks with low protocol anomaly and the neutral SR, i.e. the AP High ∧ PA Low ∧ SR Moderate cell that rule R36 now covers. Treating undefined values as 0 gives the original layer a ROC-AUC far below 0.5, which shows that the coverage gap translates into real failures on benchmark data. Second, the extra benign High/Critical ratings come almost entirely from one context-only rule: in a sample of 400 such benign flows (seed 42), R16 (AP Low ∧ PA High ∧ TV High → High) was the strongest High/Critical rule for 383 of them. Whether a highly anomalous, high-volume flow that the detector considers benign should be escalated is a policy choice; an organisation that prefers a lower alert load can demote R16 to Moderate without retraining anything. Third, the AP floor was never binding on benchmark flows, because calibrated probabilities are mostly near 0 or 1 and, with SR = 5.5, rule R36 already rates AP-High flows as High; the floor matters for trusted sources, as in Scenarios B, E and F.

Under leave-Friday-out the picture is different. The detector missed most Friday attacks, and the fuzzy layer did not rescue them: none of the missed attacks was rated Moderate or above, and the CTL ranked flows worse than AP alone (ROC-AUC about 0.30 vs. 0.65). Port-scan and botnet flows are short, low-volume exchanges with ordinary TCP behaviour (median PA 0 and median TV 145 for port-scan flows), so the context indicators point towards benign rather than malicious. Contextual escalation cannot compensate for a detector that has not seen an attack type when the attack is also unremarkable at the protocol level.

5. DISCUSSION

The revision changes the empirical picture in two ways. First, detector performance depends strongly on the evaluation protocol. Under a random split the detector remains highly accurate even without destination port and initial window sizes, but the

temporal hold-outs show what happens when the test traffic is later in time or contains attack types absent from training. The abstract and conclusions now report these results together; the random-split numbers are an upper bound for this dataset, not an estimate of operational performance, and the known labelling problems of CICIDS2017 [21,22] limit how precisely any score on it can be interpreted.

Second, the decision layer is now specified and tested as a policy. Its inputs have units and training-only estimators, its rule base is complete, a confident detection cannot be suppressed below Moderate by context, and AP influences the output in every context. On the benchmark test flows the layer behaves as a prioritisation policy: it orders flows mainly by detector confidence, adds protocol context, and attaches a rule trace to each decision. It is not expected to rank flows better than AP alone on a dataset where the detector is already near-perfect and SR carries no information. The experiments confirm this: the CTL ranks test flows slightly worse than AP alone, it adds a small benign alert load through one context-only rule, and it does not recover attacks that the detector misses under temporal shift. The value of the layer is that the ranking policy is explicit, auditable and adjustable without retraining the detector, and that each decision carries a rule trace. The ablation with percentile-based PA also shows that the definition of the context indicators matters at least as much as the rules: the same rule base produced several times more benign High/Critical ratings when PA was not anchored to the tail of the training distribution.

The comparison with transparent baselines clarifies when a fuzzy system is worth its complexity. A weighted score already separates severe from benign contexts in the controlled scenarios, so that separation alone is not evidence for fuzzy logic. What the fuzzy layer adds is (i) conjunctive policies that an additive score cannot express, such as the AP floor and 'trusted source AND no anomaly', which together account for the interaction share of the variance in Table **15**; (ii) graded transitions, avoiding the jumps of up to 40 CTL points that a crisp lookup table produces for a 0.01 change in AP; and (iii) a rule-level explanation of each decision. If an organisation does not need interactions, the weighted score is a reasonable and simpler alternative.

The two-level explanation remains the central interpretability argument. SHAP identified which flow attributes drove XGBoost predictions, and its ranking of destination port and initial window sizes in the original model exposed a probable shortcut that the revised evaluation removes. The fuzzy rule trace, in contrast, explains why the operational CTL changed after considering anomaly, trust, traffic and duration. SHAP therefore explains model behaviour, whereas the rule base exposes the explicit decision policy [2–4].

Operational deployment requires governance of the fuzzy policy. Membership parameters and rules should be reviewed by domain experts, version controlled, re-verified for coverage after every change, and validated against incident-response priorities. NIST's emphasis on mapping, measuring, managing and governing AI risk provides a useful broader context for this type of transparent decision support [5].

5.1. Limitations and Threats to Validity

- The experiments use a deterministic 10% file-wise sample of CICIDS2017 rather than every flow, and CICIDS2017 contains documented labelling and flow-construction errors [21,22]. Results should be confirmed on corrected versions of the dataset and on additional, more recent benchmarks.
- The temporal hold-outs use record order as a time proxy because MachineLearningCSV has no timestamps; T2 also mixes temporal shift with unseen attack types.
- SR is constant (5.5) in all benchmark experiments because no leakage-free reputation source exists for CICIDS2017; the SR-dependent rules are therefore evaluated only in controlled analyses.
- CD never exceeds about 120 s on CICIDS2017 because of the CICFlowMeter activity timeout, so the CD Long set and the rules that use it are inactive in the benchmark evaluation.
- PA and TV are percentile-based and therefore relative to the training traffic; a shift in the traffic mix changes their meaning and their reference statistics must be refreshed.
- Membership functions, rules and the floor value encode expert assumptions. Sensitivity analyses show that the scenario categories are stable to moderate perturbations, but the parameters are not optimal in any formal sense, and no analyst study has yet tested whether the rule traces improve triage decisions.
- The fuzzy layer was evaluated on flows labelled by the dataset, not on analyst triage outcomes; precision among the top-ranked flows is a proxy for, not a measurement of, analyst workload.
- SHAP explanations describe model behaviour, not causal mechanisms.

6. CONCLUSION

This paper presented an explainable Fuzzy–AI framework that separates machine-learning intrusion detection from contextual threat assessment. In response to the review, the decision layer was fully

specified (units, training-only context estimators, membership-function rationale and an interpretable CTL scale), its rule base was completed to 100% coverage with an explicit AP floor, and it was evaluated on every test flow, against transparent baselines and with global and local sensitivity analyses. The detector was re-evaluated without capture-artefact features, under temporal hold-outs, with isotonic calibration and over five seeds. Near-perfect random-split scores (F1 = 0.9875 ± 0.0014 with reduced features) did not carry over unchanged to temporal hold-outs (F1 = 0.9691 ± 0.0014 and 0.4478 ± 0.0185), which is why the framework treats the detector probability as evidence to be contextualised rather than as a final decision. On all test flows the revised layer kept nearly all detected attacks in the High/Critical classes, replaced the undefined outputs of the original rule base, and made its policy explicit, but it did not rank flows better than the detector probability and could not recover attacks missed under temporal shift. Two events with the same AP of 0.82 received CTL 81.2 (Critical) and 50.0 (Moderate) under severe and benign context, and AP, PA and SR were the dominant sources of CTL variance. Future work should validate the system on corrected and more recent datasets with real reputation data, learn or tune membership functions under interpretability constraints, investigate interval type-2 fuzzy sets and Z-number reliability, and conduct analyst-centred evaluation in realistic triage environments.

DATA AVAILABILITY

The CICIDS2017 dataset is publicly available from the Canadian Institute for Cybersecurity, University of New Brunswick. The experiment used the eight MachineLearningCSV files distributed with the dataset. No private or participant-level data were used.

CODE AVAILABILITY

The revised reproducible Python workflow is provided as Supplementary File S1 (cicids2017_fuzzy_ai_experiment_v2.py, fuzzy_engine.py, fuzzy_analysis.py and make_figures.py). It covers preprocessing, the full and reduced feature sets, random and temporal splits, model training over five seeds, isotonic calibration, SHAP computation, the context builder, the original and revised Mamdani engines, the baselines and all sensitivity analyses. Seeds and hyperparameters are reported in Sections 3.10.3–3.10.4.

AUTHOR CONTRIBUTIONS

Conceptualization, methodology, investigation, supervision, project administration, and writing - original draft: Rahib Imamguluyev. Formal analysis, software, validation, visualization, and writing - review & editing: Rahib Imamguluyev. The author has read and approved the final version of the manuscript.

FUNDING

This research received no external funding.

ACKNOWLEDGEMENTS

The author acknowledges the Canadian Institute for Cybersecurity at the University of New Brunswick for making the CICIDS2017 benchmark dataset available to the research community, and the reviewers for comments that substantially improved the paper.

DECLARATION OF AI-ASSISTED TECHNOLOGY USE

Generative artificial intelligence was used to assist with language refinement, document organization, formatting, and development of reproducible code scaffolding. The author retained full responsibility for the study design, scientific judgments, verification of references, execution and validation of experiments, interpretation of results, and final manuscript content. No AI-generated or fabricated experimental data were used.

CONFLICT OF INTEREST

The authors declare no conflict of interest regarding the research, authorship, and publication of this article.

ETHICS STATEMENT

The study uses a publicly available network-security benchmark dataset and does not involve human participants or personally identifiable participant data.

SUPPLEMENTAL MATERIALS

The supplemental materials can be downloaded from the journal website along with the article.

REFERENCES

[1] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization," in Proceedings of the 4th International Conference on Information Systems Security and Privacy (ICISSP), 2018, pp. 108–116. doi: 10.5220/0006639801080116.

[2] S. M. Lundberg and S.-I. Lee, "A Unified Approach to Interpreting Model Predictions," in Advances in Neural Information Processing Systems 30, 2017.

[3] G. Rjoub, J. Bentahar, O. Abdel Wahab, R. Mizouni, A. Song, R. S. Cohen, H. Otrok, and A. Mourad, "A Survey on Explainable Artificial Intelligence for Cybersecurity," IEEE Transactions on Network and Service Management, vol. 20, no. 4, pp. 5115–5140, 2023. doi: 10.1109/TNSM.2023.3282740.

[4] F. Charmet, H. C. Tanuwidjaja, S. Ayoubi, P. Gimenez, Y. Han, H. Jmila, G. Blanc, T. Takahashi, and Z. Zhang, "Explainable artificial intelligence for cybersecurity: a literature survey," Annals of Telecommunications, vol. 77, pp. 789–812, 2022.

doi: 10.1007/s12243-022-00926-7.

[5] E. Tabassi, Artificial Intelligence Risk Management Framework (AI RMF 1.0), NIST AI 100-1. National Institute of Standards and Technology, 2023. doi: 10.6028/NIST.AI.100-1.

[6] L. A. Zadeh, "Fuzzy Sets," Information and Control, vol. 8, no. 3, pp. 338–353, 1965. doi: 10.1016/S0019-9958(65)90241-X.

[7] E. H. Mamdani, "Application of fuzzy algorithms for control of simple dynamic plant," Proceedings of the Institution of Electrical Engineers, vol. 121, no. 12, pp. 1585–1588, 1974. doi: 10.1049/piee.1974.0328.

[8] L. Breiman, "Random Forests," Machine Learning, vol. 45, pp. 5–32, 2001. doi: 10.1023/A:1010933404324.

[9] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," in Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016, pp. 785–794. doi: 10.1145/2939672.2939785.

[10] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," in International Conference on Learning Representations (ICLR), 2015.

[11] F. Pedregosa *et al.*, "Scikit-learn: Machine Learning in Python," Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011.

[12] D. W. Apley and J. Zhu, "Visualizing the Effects of Predictor Variables in Black Box Supervised Learning Models," Journal of the Royal Statistical Society: Series B, vol. 82, no. 4, pp. 1059–1086, 2020. doi: 10.1111/rssb.12377.

[13] M. T. Ribeiro, S. Singh, and C. Guestrin, "Why Should I Trust You?: Explaining the Predictions of Any Classifier," in Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016, pp. 1135–1144. doi: 10.1145/2939672.2939778.

[14] C. Molnar, Interpretable Machine Learning: A Guide for Making Black Box Models Explainable, 2nd ed., 2022. Available: https://christophm.github.io/interpretable-ml-book/

[15] K. Alsubhi, I. Aib, and R. Boutaba, "FuzMet: a fuzzy-logic based alert prioritization engine for intrusion detection systems," International Journal of Network Management, vol. 22, no. 4, pp. 263–284, 2012. doi: 10.1002/nem.804.

[16] M. Alali, A. Almogren, M. M. Hassan, I. A. L. Rassan, and M. Z. A. Bhuiyan, "Improving risk assessment model of cyber security using fuzzy logic inference system," Computers & Security, vol. 74, pp. 323–339, 2018. doi: 10.1016/j.cose.2017.09.011.

[17] A. H. Hamamoto, L. F. Carvalho, L. D. H. Sampaio, T. Abrão, and M. L. Proença Jr., "Network anomaly detection system using genetic algorithm and fuzzy logic," Expert Systems with Applications, vol. 92, pp. 390–402, 2018. doi: 10.1016/j.eswa.2017.09.013.

[18] A. N. Toosi and M. Kahani, "A new approach to intrusion detection based on an evolutionary soft computing model using neuro-fuzzy classifiers," Computer Communications, vol. 30, no. 10, pp. 2201–2212, 2007. doi: 10.1016/j.comcom.2007.05.002.

[19] J. Liu, W. Yinchai, T. C. Siong, X. Li, L. Zhao, and F. Wei, "A hybrid interpretable deep structure based on adaptive neuro-fuzzy inference system, decision tree, and K-means for intrusion detection," Scientific Reports, vol. 12, 2022. doi: 10.1038/s41598-022-23765-x.

[20] J.-S. R. Jang, "ANFIS: adaptive-network-based fuzzy inference system," IEEE Transactions on Systems, Man, and Cybernetics, vol. 23, no. 3, pp. 665–685, 1993. doi: 10.1109/21.256541.

[21] G. Engelen, V. Rimmer, and W. Joosen, "Troubleshooting an intrusion detection dataset: the CICIDS2017 case study," in 2021 IEEE Security and Privacy Workshops (SPW), 2021, pp. 7–12. doi: 10.1109/SPW53761.2021.00009.

[22] M. Lanvin, P.-F. Gimenez, Y. Han, F. Majorczyk, L. Mé, and É. Totel, "Errors in the CICIDS2017 dataset and the significant differences in detection performances it makes," in Risks and Security of Internet and Systems (CRiSIS 2022), Lecture Notes in Computer Science, vol. 13857, Springer, 2023, pp. 18–33. doi: 10.1007/978-3-031-31108-6_2.

[23] D. Arp, E. Quiring, F. Pendlebury, A. Warnecke, F. Pierazzi, C. Wressnegger, L. Cavallaro, and K. Rieck, "Dos and don'ts of machine learning in computer security," in Proceedings of the 31st USENIX Security Symposium, 2022, pp. 3971–3988.

[24] F. Pendlebury, F. Pierazzi, R. Jordaney, J. Kinder, and L. Cavallaro, "TESSERACT: eliminating experimental bias in malware classification across space and time," in Proceedings of the 28th USENIX Security Symposium, 2019, pp. 729–746.

[25] S. Neupane, J. Ables, W. Anderson, S. Mittal, S. Rahimi, I. Banicescu, and M. Seale, "Explainable intrusion detection systems (X-IDS): a survey of current methods, challenges, and opportunities," IEEE Access, vol. 10, pp. 112392–112415, 2022. doi: 10.1109/ACCESS.2022.3216617.

[26] M. Wang, K. Zheng, Y. Yang, and X. Wang, "An explainable machine learning framework for intrusion detection systems," IEEE Access, vol. 8, pp. 73127–73141, 2020. doi: 10.1109/ACCESS.2020.2988359.

[27] H. Hagras, "Toward human-understandable, explainable AI," Computer, vol. 51, no. 9, pp. 28–36, 2018. doi: 10.1109/MC.2018.3620965.

[28] M. J. Gacto, R. Alcalá, and F. Herrera, "Interpretability of linguistic fuzzy rule-based systems: an overview of interpretability measures," Information Sciences, vol. 181, no. 20, pp. 4340–4360, 2011. doi: 10.1016/j.ins.2011.02.021.

[29] C. C. Lee, "Fuzzy logic in control systems: fuzzy logic controller, Part I," IEEE Transactions on Systems, Man, and Cybernetics, vol. 20, no. 2, pp. 404–418, 1990. doi: 10.1109/21.52551.

[30] A. Niculescu-Mizil and R. Caruana, "Predicting good probabilities with supervised learning," in Proceedings of the 22nd International Conference on Machine Learning (ICML), 2005, pp. 625–632. doi: 10.1145/1102351.1102430.

[31] B. Zadrozny and C. Elkan, "Transforming classifier scores into accurate multiclass probability estimates," in Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2002, pp. 694–699. doi: 10.1145/775047.775151.

[32] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," in Proceedings of the 34th International Conference on Machine Learning (ICML), PMLR vol. 70, 2017, pp. 1321–1330.

[33] A. Saltelli, P. Annoni, I. Azzini, F. Campolongo, M. Ratto, and S. Tarantola, "Variance based sensitivity analysis of model output. Design and estimator for the total sensitivity index," Computer Physics Communications, vol. 181, no. 2, pp. 259–270, 2010. doi: 10.1016/j.cpc.2009.09.018.

[34] J. Herman and W. Usher, "SALib: an open-source Python library for sensitivity analysis," Journal of Open Source Software, vol. 2, no. 9, p. 97, 2017. doi: 10.21105/joss.00097.

[35] R. Sommer and V. Paxson, "Outside the closed world: on using machine learning for network intrusion detection," in 2010 IEEE Symposium on Security and Privacy, 2010, pp. 305–316. doi: 10.1109/SP.2010.25.

[36] B. Iglewicz and D. C. Hoaglin, How to Detect and Handle Outliers. Milwaukee, WI: ASQC Quality Press, 1993.

Received on 30-09-2026 Accepted on 12-10-2026 Published on 29-10-2026

https://doi.org/10.65879/3070-5789.2026.02.08

© 2026 Rahib Imamguluyev.
This is an open access article licensed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/4.0/) which permits unrestricted use, distribution and reproduction in any medium, provided the work is properly cited.