A Prioritized Risk Scoring Metric for Host-Based Threat Detection in Distributed NGFW Environments

Hafis V.C.[1,2]* and Khandakar F. Rahman[1]

[1] Dept. of Computer Science & Engineering, Banasthali Vidyapith, Rajasthan, India
kfrahman@banasthali.in
[2] Higher Colleges of Technology, Abu Dhabi, United Arab Emirates
hafisali.phd@gmail.com*

Abstract. NGFWs at enterprise scale generate several largely independent telemetry streams per session—application identification, threat signatures, sandbox verdicts, policy decisions—yet security operations teams still triage most of this as separate alert feeds rather than a single per-host risk figure. We present the Prioritized Risk Scoring Metric (PRSM), which fuses six such channels—traffic behavioral deviation, App-ID reconstruction residual, threat trigger frequency, WildFire sandbox verdicts, policy denial tallies, and hourly activity divergence—into one empirically weighted composite score per host, computed from source IP alone so that no identity-management infrastructure is required. Two of these six channels cannot be evaluated against real vendor telemetry in this study: the public CICIDS2017 benchmark carries no native App-ID or WildFire fields, so we substitute proxy flow-level features for those two channels specifically, to keep the benchmark reproducible. That substitution is a real limitation and not a footnote: it means the results below demonstrate that the composite-scoring approach works, not that all six channels have been validated against actual NGFW output. Under a leakage-controlled protocol—coefficients and thresholds tuned on a validation partition, all results reported exactly once on a disjoint test partition—PRSM attains an AUC-ROC of 0.903, exceeding every standalone channel, equal-weight blending, and the single-algorithm baselines tested, with a 16% false-positive-rate reduction relative to equal-weight blending. Ablation shows this benchmark's discriminative power concentrates almost entirely in two channels (App-ID anomaly and temporal divergence); the remaining four are architecturally specified but show no measurable standalone contribution on this corpus, so the present evaluation should be read as validating a two-channel detector, with the other four awaiting confirmation on native NGFW telemetry. A follow-up incident-level check finds no consistent time-to-detection advantage over an equal-weight comparator, so the "prioritized" framing refers to delivering one composite, actionable score ahead of manual per-channel alert triage, not to measured earlier detection.

Keywords: Risk scoring · Ngfw telemetry · Host-based anomaly detection · Autoencoder · Distributed enterprise networks · Wildfire

© The Author(s) 2026
A. Shastri et al. (eds.), *Proceedings of the Fourth Workshop on Applications of Computational Intelligence and Interactive Systems (WACIIS 2026)*, Advances in Intelligent Systems Research 217,
https://doi.org/10.2991/978-94-6239-832-0_2

1 Introduction

Signature-driven perimeter defences have proven inadequate against the intrusion styles that now dominate incident reports: patient targeted campaigns, credential theft, and living-off-the-land (LotL) tradecraft [1]. A modern NGFW such as the Palo Alto Networks PA-445 observes each session at several inspection layers at once—App-ID application fingerprinting, Content-ID payload analysis, WildFire cloud sandboxing, and inline threat signatures [2]—and emits structured log records for each. That gives an operations team enormous visibility in principle, and, in practice, more log volume and format variety than anyone can reasonably parse by hand.

Part of the reason is what the field still benchmarks itself against. KDD-99 and the NSL-KDD revision of it [3] remain standard reference points in anomaly-detection research, despite containing nothing resembling what a production NGFW actually logs — no App-ID tags, no WildFire verdicts, no policy-engine context of any kind. Deployed detectors, meanwhile, mostly stop at a binary verdict or a standalone alert; none of that gets aggregated into a single risk figure an analyst can act on directly. Axelsson's base-rate argument [4] explains why this matters more than it might seem: because attacks are rare relative to total traffic volume, even a detector with strong per-event accuracy generates enough false positives in absolute terms to overwhelm a SOC. Sommer and Paxson [5] identified the other half of the same problem — detectors that perform well on a static lab corpus routinely degrade once deployed against the drift of live production traffic.

A third obstacle gets less attention in published work than it deserves given how often it turns up operationally: most behavioral-profiling schemes assume every session maps to a named user via an identity connector, Palo Alto's User-ID agent being the usual example. Deploying and maintaining that mapping is not free, and several deployment scenarios rule it out entirely for reasons that have nothing to do with the detector itself — budget, operational overhead, or a privacy policy that simply forbids the binding. The result is an organization sitting on complete firewall telemetry with no reliable identity layer on top of it, unable to use any detection scheme built around per-user profiles.

PRSM targets exactly that gap. It reduces six NGFW-derived telemetry channels to one actionable number per host, keyed to source IP rather than to any authenticated identity, computed independently per endpoint at each branch. Nothing in that construction needs to resolve who was behind the keyboard, which has three practical consequences: the metric is privacy-preserving by design, it carries over to a different firewall vendor without modification, and it can be deployed anywhere ordinary log forwarding is already in place. This paper makes three contributions:

- a formally specified composite score built on the six host-level dimensions drawn entirely from operational NGFW logs, with no dependency on user-identity binding anywhere in its construction;

- an evaluation protocol that is reproducible end to end against the publicly available CICIDS2017 corpus, giving other researchers a verifiable reference point rather than a number they have to take on faith;
- empirical evidence, worked through in Sections 5–7, that weighted multi-channel fusion at the host layer outperforms any single channel on its own, naïve equal-weight blending, and the monolithic single-model baselines we tested.

2 Related Work

2.1 Neural Approaches to Traffic Anomaly Identification

Deep learning reshaped this space largely on the back of two ideas: layered feature learning [6], which let models discover traffic representations that nobody had to hand-engineer, and the LSTM's gated recurrence [7], which turned out to be a natural fit for anything ordered in time, traffic included. Sakurada and Yairi [8] took the autoencoder version of this idea and applied it cleanly: train only on the benign data, then treat an unusually large reconstruction residual as the anomaly signal itself. Kitsune [9] is probably the furthest this line of work has been pushed toward real deployment — an ensemble of small autoencoders running at packet level, in real time. Its limitation is telling for the argument in this paper: Kitsune consumes raw flow statistics and nothing else, so everything an NGFW captures beyond that — application-layer semantics, cloud sandbox verdicts — simply isn't visible to it.

DeepLog [10] applied LSTMs to system-log event sequences and caught departures from the learned execution paths; Zavrak and İskefiyeli [11] also showed that variational autoencoders over flow features can surface attack traffic never seen in the training. Garcia-Teodoro et al.'s survey [12] grouped the field into statistical, knowledge-driven, and learning-based families, and put a name to the difficulty that persists across all three: detection capability rarely converts cleanly into everyday operational value. The toolbox has since broadened further with payload-level CNN/RNN hybrids [22] and Buczak and Guven's data-mining catalogue [19], and more recent surveys [23, 24] converge on the same finding across public benchmarks — hybrid and ensemble approaches consistently outperform single models.

2.2 Endpoint-Centric Behavioral Profiling

Endpoint-anchored behavioral profiling is not a new direction, but it has become more attractive precisely as identity-centric approaches keep running into the wall described above. An observation from Chandola et al.'s survey [13] underlies a good deal of how PRSM is designed: a host's own fingerprint — its connection patterns, its data volume, the regularity of when it is active — frequently signals a compromise before any classical indicator catches it. Beehive [14] confirmed this at enterprise scale: compromised machines measurably alter their communication graphs well ahead of any indicator-of-compromise alert firing.

Zero trust is worth a mention here too, mostly because it is often assumed to depend on the same identity infrastructure PRSM is designed to avoid needing. Kindervag's original formulation [15] and its later NIST SP 800-207 codification [16] both hold, in principle, that network presence alone should never confer trust, with identity verification sitting at the center of that doctrine. In practice, though, most real deployments lean far more on device posture and host attestation than on verified user identity — which is closer to the assumption PRSM actually makes. The clearest precedent for our approach is Aleroud and Karabatis's survey of context fusion for intrusion detection [17]: stacking several context categories onto one detector measurably improves attack prediction, and that is the same bet PRSM's six-channel design is making.

2.3 Composite Risk Quantification in SOC Workflows

Detection itself is a well-studied problem from nearly every angle. What gets far less attention is the narrower, more operational question of compressing many detections into a single number an analyst can actually rank hosts by. Commercial SIEMs all ship some kind of scoring engine, but the internals are proprietary and, as far as we can tell, none of them have ever been benchmarked in the open literature. The stakes for closing this gap are not abstract: the Verizon DBIR [1] still puts the median breach-to-discovery window at days to weeks across most sectors, which is a long time for a compromised host to sit unscored and unranked in an analyst's queue.

Existing work addresses pieces of this problem but stops short of PRSM's specific combination of scope and validation. Kitsune [9] demonstrates that ensemble autoencoder fusion works well for real-time anomaly scoring, but operates purely on packet-level flow statistics and produces no risk value above the level of an individual detector; by construction it cannot incorporate application-identification or sandbox-verdict signal even where such telemetry exists. DeepLog [10] and the Beehive system [14] both show that sequence- and graph-based host profiling surfaces compromise earlier than point alerts, but each is scoped to a single telemetry source (system-call logs and communication-graph features, respectively) rather than a composite drawn from heterogeneous NGFW channels. Aleroud and Karabatis's survey [17] identifies contextual fusion across categories as a promising direction and catalogues where it has been attempted, but does not identify a peer-reviewed instantiation that fuses NGFW-native channels—application identification, sandbox verdicts, policy enforcement, temporal behavior—into one validated, host-level, identity-free score. PRSM's contribution is therefore not the general idea of weighted multi-signal fusion, which is well established, but its specific, reproducible instantiation for NGFW telemetry: a six-channel formulation validated end-to-end under a leakage-controlled protocol, with an honest ablation reporting which channels actually carry the discriminative signal on the evaluated benchmark rather than assuming all channels contribute equally.

3 System Architecture

3.1 Deployment Topology and Telemetry Acquisition

The reference deployment described here operates across five branch offices in the United Arab Emirates, each fronted by its own Palo Alto Networks PA-445 running PAN-OS 11.2.4-h7. All five appliances also forward their logs over syslog (UDP 514) to a single central Kiwi Syslog collector, and four log families make it into the pipeline from there: traffic session records, threat-engine events (antivirus, anti-spyware, and vulnerability signatures together), URL-category verdicts, and the WildFire cloud-analysis outcomes. Every internet-facing security rule at every branch carries a WildFire Analysis profile attached to it, which is what makes the file-submission coverage complete rather than partial.

Getting from raw syslog to a PRSM value happens across four stages, run in sequence:

Stage 1, Ingestion. A dedicated log-forwarding profile sits on every active security rule and is what actually carries each PA-445's records to the central collector — without it, a rule's traffic simply never reaches the pipeline.

Stage 2—Normalization: Raw entries are parsed, field-normalized, tagged with the originating host IP, and time-aligned across branches. Binding each record to its source IP obviates the need for any User-ID linkage.

Stage 3—Anomaly Modeling: An autoencoder pipeline computes the D_2 (App-ID Anomaly) channel from distributional deviations at host granularity. A separately trained LSTM sequence model targets sequential deviations but is used only to construct the LSTM-only comparison baseline (Table 3); it is not one of the six PRSM channels and does not feed the composite score.

Stage 4—Score Synthesis: The six normalized channel outputs are fused into per-host and per-branch PRSM values; configurable threshold bands trigger graduated response workflows.

3.2 Reference-Behavior Calibration

Normal-behavior profiles are calibrated by collecting and analyzing traffic telemetry from every branch over a continuous 30-day window. Two granularity tiers emerge: individual host fingerprints keyed on IP address, and aggregate branch-level traffic and threat profiles.

Two model families operate on this calibration data, serving different roles. A mirror-symmetric autoencoder compresses a 36-attribute feature vector down to a 10-attribute bottleneck; whenever the reconstruction residual exceeds the 99th-percentile calibration threshold, an anomaly flag is raised, and this residual is the D_2 channel used in the PRSM composite score. A separate stacked LSTM with layers of 128 and 64 units, followed by a dense head, consumes length-24 input sequences (each step representing one hour) and forecasts the subsequent step; the magnitude of the forecast residual serves as the anomaly signal for the LSTM-only baseline reported in Table 3. This LSTM output is not a PRSM channel and does not contribute to the composite score.

4 Prioritized Risk Scoring Metric

4.1 Mathematical Formulation

Let h denote a host identified by its source IP and t denote the evaluation window. The composite risk value is:

$$\mathrm{PRSM}(h,t) = \sum_{i=1}^{6} w_i \cdot D_i(h,t) \tag{1}$$

with the constraints $w_i \geq 0.05\ \forall i$ (the grid-search floor; no channel weight may vanish entirely) and $\sum_{i=1}^{6} w_i = 1$. Every channel $D_i(h,t)$ is scaled to $[0,1]$ for uniform comparability. Table 1 catalogues the six channels.

Table 1. PRSM Channels with Telemetry Origins

i Channel	Origin	Computation
1 Host Traffic Dev.	Traffic logs	Mahalanobis dist.
2 App-ID Anomaly	App-ID logs	Autoencoder residual
3 Threat Det. Freq.	Threat logs	Normalized rate
4 WildFire Verdicts	WildFire logs	Malicious ratio
5 Policy Violation	URL/traffic logs	Deny/block count
6 Temporal Anomaly	Timestamps	J-S divergence

4.2 Channel Computation

D_1—Traffic Behavioral Deviation. The Mahalanobis distance separates the host's present-window feature vector from its calibration-period centroid:

$$D_1(h,t) = \min\left(1, \frac{M(\mathbf{x}_{h,t}, \boldsymbol{\mu}_h, \boldsymbol{\Sigma}_h)}{M_{\max}}\right) \tag{2}$$

The feature vector $\mathbf{x}_{h,t}$ encodes session count, distinct destination IPs, distinct destination ports, inbound-to-outbound byte ratio, mean session duration, and count of previously unseen destinations. $\boldsymbol{\mu}_h$ and $\boldsymbol{\Sigma}_h$ are the calibration-period mean and covariance; $M_{\max}$ caps the normalization.

D_2—App-ID Reconstruction Residual. An autoencoder trained on the host's baseline application-usage profile outputs a reconstruction residual that flags novel or shifted application patterns. Input attributes cover per-category bandwidth shares, new-application count, and encrypted-session ratio; the residual is normalized against the 99th-percentile calibration residual.

D_3—Threat Trigger Rate. Antivirus, anti-spyware, and vulnerability-signature firings per host are counted within a four-hour sliding window and normalized by the branch-level historical mean.

D_4—*Sandbox Verdict Distribution.* The fraction of file submissions receiving malicious or grayware verdicts from WildFire is computed as:

$$D_4(h,t) = \frac{N_{\text{mal}} + 0.5 \cdot N_{\text{gray}}}{\max(N_{\text{total}}, 1)} \tag{3}$$

Grayware entries carry half the weight of confirmed-malicious entries.

D_5—*Policy Denial Rate.* Firewall block actions, URL-filter denials, and application-control rejections attributed to the host are aggregated and scaled relative to that host's calibration-period average.

D_6—*Temporal Activity Divergence.* The Jensen–Shannon divergence between the host's current hourly activity histogram and its calibration histogram gauges schedule-based anomalies:

$$D_6(h,t) = \min\left(1, \frac{JSD(P_{h,t} \parallel P_{h,\text{cal}})}{JSD_{\max}}\right) \tag{4}$$

Both $P_{h,t}$ and $P_{h,\text{cal}}$ are 24-bin distributions over the hours of the day.

4.3 Coefficient Optimization

Coefficients are selected via exhaustive grid search at 0.05 resolution under the unit-sum constraint: w_1–w_4 range over $\{0.05, \ldots, 0.40\}$, w_5 over $\{0.05, \ldots\}$ of the remainder, and w_6 absorbs the balance. AUC-ROC is maximized on the *validation* partition only (Sect. 5.1); the test partition on which all results in Sect. 6 are reported is never used for coefficient or threshold selection. The optimum places w_2 at its 0.40 search-space boundary, so a marginally higher App-ID weight is not excluded by this grid. Table 2 lists the resulting values.

Table 2. Optimized PRSM Coefficients and Rationale

Channel	w_i	Rationale
D_1 Traffic Dev.	0.05	Baseline behavioral gauge
D_2 App-ID Anomaly	0.40	Co-dominant application signal
D_3 Threat Freq.	0.05	Direct threat evidence
D_4 WildFire Verdicts	0.05	Cloud-confirmed malware
D_5 Policy Denial	0.05	Indirect policy signal
D_6 Temporal Div.	0.40	Co-dominant schedule signal

The rationale column records what each channel was designed to capture. Tuning on validation places the weight almost entirely on two channels—D_2 (App-ID anomaly) and D_6 (temporal divergence), 0.40 apiece—and leaves the other four at the 0.05 floor of the grid. The ablation study (Sect. 6.3, Table 5) supports this concentration: removing D_2 or D_6 costs 0.054 and 0.139 AUC points respectively on the test partition, whereas removing D_1, D_3, D_4, or D_5 individually leaves AUC essentially unchanged or marginally higher, so the present evaluation validates a two-channel detector rather than a six-channel fusion.

4.4 Severity Bands

The resulting composite values are partitioned into four operational tiers: a quiescent zone spanning $[0.00, 0.25)$, a moderate zone spanning $[0.25, 0.50)$, an elevated zone spanning $[0.50, 0.75)$, and a critical zone from 0.75 upward. Tier boundaries are tunable per deployment and dictate the escalation level of automated containment actions. They are operational conventions rather than evaluated detection thresholds; the binary operating threshold used in Sect. 6 (0.41, selected on the validation partition by F1) lies within the moderate band.

5 Evaluation Protocol

5.1 Dataset and Rationale

Reproducibility is secured by running PRSM against the publicly available CICIDS2017 corpus [18, 20], which bundles labelled flows covering benign traffic alongside seven attack scenarios (14 labelled sub-categories in Table 4)—brute-force credential guessing, Heartbleed exploitation, botnet command-and-control, denial-of-service flooding, volumetric DDoS, web-application exploits, and stealthy infiltration. CICIDS2017 provides only flow-level NetFlow-style attributes and carries no Palo Alto-specific App-ID classifications or WildFire sandbox verdicts, so the six PRSM channels are computed from proxy feature groups selected from the corpus's 78 flow-level attributes: D_1 from duration, packet-count, byte-count, and inter-arrival-time features via Mahalanobis distance; D_2 from port, protocol, TCP-flag, and packet/segment/header-length features via autoencoder reconstruction error, standing in for App-ID classification residuals; D_3 from TCP-flag firing counts (SYN, FIN, RST, PSH, URG, ECE, CWE); D_4 from forward-flow, idle-time, active-time, and subflow features, standing in for true WildFire sandbox verdicts; D_5 from flag and bulk-transfer features, standing in for policy-denial signals; and D_6 from duration and inter-arrival-time features via Jensen–Shannon divergence. This substitution is necessary for benchmark reproducibility but means D_2's and D_4's CICIDS2017 performance reflects generic flow-pattern anomaly detection rather than genuine application-identification or malware-sandbox behavior; production deployment would recompute these two channels from native App-ID and WildFire logs, which this manuscript does not evaluate. Monday's purely benign segment (529,918 flows) seeds the calibration phase. The Tuesday-through-Friday segments (2,300,825 flows) are split 50/50, stratified per attack label with fixed random seed 42, into a *validation* partition (1,150,408 flows) used exclusively for coefficient grid search and threshold selection, and a *test* partition (1,150,417 flows) on which the frozen configuration is evaluated exactly once; all figures reported in Sect. 6 are test-partition results. Stratification guarantees that every attack class, including Heartbleed ($n = 11$ overall), appears on both sides of the split.

Because CICIDS2017 provides flow records rather than hourly host aggregates, the LSTM-only baseline in this benchmark is a windowed next-record

forecaster (10-step windows, a 64-unit LSTM layer with a dense head) trained on Monday's flow-ordered records; it is the benchmark stand-in for, not an instantiation of, the hourly 128/64-unit production model of Sect. 3.2. Reported figures encompass detection accuracy, precision, recall, F1, AUC-ROC, and false-alarm rate. The full pipeline, saved split indices, exact reported results, and per-run metadata are publicly available at `https://github.com/hafisali/prsm-ngfw`.

6 Results and Analysis

6.1 Benchmark Outcomes

Table 3 contrasts PRSM against four reference methods on the CICIDS2017 evaluation partition.

Table 3. Detection Performance Comparison—CICIDS2017

Method	Acc.	Prec.	Rec.	F1	AUC
PRSM	**87.1**	74.7	70.7	72.7	0.9033
LSTM-only	73.1	37.0	15.4	21.8	0.4906
AE-only	86.2	75.1	64.5	69.4	0.8510
Equal-weight	82.9	67.0	58.1	62.2	0.8285
Best single D_i (D_6)	84.2	63.7	80.7	71.2	0.8532

PRSM achieves an AUC-ROC of 0.9033 on the untouched test partition, surpassing every reference configuration in ranking quality: a 7.5-point gain over equal-weight blending (0.8285), 5.0 points over the strongest single channel (D_6, 0.8532), and 5.2 points over the autoencoder alone (0.8510). At the operating threshold selected on validation (0.41), PRSM attains 87.1% accuracy, 74.7% precision, 70.7% recall, and F1 of 72.7%, with a false-positive rate of 7.7%—16% below the equal-weight variant's 9.2% at its own F1-optimal threshold. Two honest caveats accompany these figures. First, AE-only achieves a slightly lower false-positive rate (6.8%) than PRSM, but at substantially lower recall (64.5% versus 70.7%) and AUC. Second, the windowed LSTM baseline performs at approximately chance level (AUC 0.4906) on this flow-ordered corpus: next-row sequence prediction over interleaved multi-host flow records carries little discriminative signal, and we report this result as obtained rather than tuning the baseline further.

None of the AUC gaps above come with a formal significance test, which a careful reader should notice on a single-split evaluation. As a partial check, we bootstrapped (300 resamples with replacement over the test partition) the AUC gap between PRSM and each comparator using the Sect. 6.4 reconstruction pipeline (proxy D_2, not the original Keras model, so absolute AUC values differ from the ones above): PRSM's advantage over equal-weight blending was 0.066 (95% CI [0.0655, 0.0666]) and over the autoencoder alone was 0.061 (95%

CI [0.0600, 0.0621]), with neither interval coming remotely close to zero at this sample size ($n \approx 1.15$M test rows). This supports the ranking claim's robustness to resampling noise, though it is evidence about the reconstruction, not a direct interval estimate of Table 3's own figures. We did not attempt the same bootstrap for the false-positive-rate comparison, because the reconstruction's proxy autoencoder selects a materially different validation-tuned threshold (0.19) than Table 3's pipeline (0.41), so a resampled FPR gap from this reconstruction would not be informative about the 16% figure above; that figure rests on the single test-partition evaluation only and has not been independently interval-estimated.

6.2 Per-Category Hit Rates

Table 4 reports PRSM's per-attack-category hit rates at the single global operating threshold (0.41), together with the number of test-partition instances per category. The picture is sharply bimodal. Categories with strong application-pattern or temporal footprints are detected well: DoS-Slowhttptest reaches 99.2% and Heartbleed 100% (the latter on only 6 test instances, a small-sample result), while the volumetric floods land in the 65–79% range (DoS-Hulk 78.6% on $n = 115{,}537$; DDoS 71.0%; PortScan 66.5%); Infiltration's 77.8% ($n = 18$) belongs with Heartbleed in this small-sample caveat, rather than alongside the large-n flood categories it happens to be listed near in Table 4. In stark contrast, low-rate credential and web attacks are almost entirely missed at this operating point: FTP-Patator 1.1%, SSH-Patator 0.8%, Web Attack–Brute Force 0.1%, XSS 2.5%, and SQL Injection 0.0% (the last of these on only 11 test instances, also a small-sample result). Their mean PRSM scores (0.12–0.20) sit far below the global threshold, so although the composite score still ranks them above much benign traffic in AUC terms, a single institution-wide threshold does not convert that separation into alerts. Per-category or per-band thresholds are the natural remedy and are left to future work; we report the single-threshold result because it reflects the deployment model described in Sect. 4.4.

6.3 Ablation Study

Systematic removal of individual channels, with the validation-tuned weights held fixed and renormalized, shows the composite's signal is concentrated in two channels. D_6 (Temporal Divergence) matters most: removing it takes test-partition AUC from 0.9033 to 0.7646, a 0.1387-point fall. D_2 (App-ID Anomaly) is the second load-bearing channel, costing 0.0542 points (down to 0.8491) when it is dropped. Removing any one of D_1, D_3, D_4, or D_5 leaves the AUC flat or fractionally higher (Table 5)—on this benchmark those four add no demonstrable standalone detection value and may inject mild noise into the composite. This does not rule out their value in principle—D_1, D_3, D_4, and D_5 were designed for attack patterns (e.g., low-and-slow reconnaissance against production infrastructure) that CICIDS2017's proxy features cannot express—but the honest characterization of the present evaluation is a two-channel detector ($D_2 + D_6$)

Table 4. PRSM Per-Category Hit Rates at the Global Operating Threshold—CICIDS2017 Test Partition

Attack Category	Hit Rate (%)	N
Heartbleed	100.0	6
DoS-Slowhttptest	99.2	2,750
DoS-Hulk	78.6	115,537
Infiltration	77.8	18
DDoS	71.0	64,014
PortScan	66.5	79,465
DoS-Slowloris	64.9	2,898
Bot	60.9	983
DoS-GoldenEye	54.8	5,147
Web Attack–XSS	2.5	326
FTP-Patator	1.1	3,969
SSH-Patator	0.8	2,949
Web Attack–Brute Force	0.1	754
Web Attack–SQL Injection	0.0	11

with four channels awaiting validation on native NGFW telemetry. Table 5 reports the exact AUC-ROC obtained with each channel excluded.

Table 5. Ablation Results: AUC-ROC With Each Channel Removed

Configuration	AUC-ROC	Δ (Full − Without)
Full (all six channels)	0.9033	–
$-D_1$ (Traffic Deviation)	0.9067	−0.0034
$-D_2$ (App-ID Anomaly)	0.8491	+0.0542
$-D_3$ (Threat Freq.)	0.9042	−0.0009
$-D_4$ (WildFire Verdicts)	0.9075	−0.0042
$-D_5$ (Policy Denial)	0.9051	−0.0018
$-D_6$ (Temporal Div.)	0.7646	+0.1387

6.4 Baseline Robustness and Lead-Time Check

Two follow-up checks address questions that Sections 6.1–6.3 could not: whether the LSTM-only baseline's near-chance result (Table 3) reflects a genuine limitation of sequence models on this task, and whether PRSM's "prioritized" framing has measured temporal-lead evidence behind it. Both were run on a lightweight reproduction pipeline (scikit-learn MLP proxies in place of the original Keras autoencoder/LSTM, trained on a capped subsample of Monday's baseline traffic) rather than the original TensorFlow environment; the absolute figures below are consequently not directly comparable to Table 3 and are reported only to establish direction, not to replace the headline results.

Baseline construction. The published LSTM-only baseline predicts each row from the immediately preceding row in raw file-concatenation order, which interleaves flows from different hosts and days with no temporal relationship to one another. Rebuilding the same next-flow-prediction task ordered instead by (source IP, timestamp) — so each prediction genuinely uses the preceding flow from the same host — raises test-partition AUC from 0.49 to 0.67 under an identical evaluation protocol (validation-selected threshold, test-partition-only scoring, accuracy 78.7% vs. 73.1%, F1 43.3 vs. 21.8). This confirms the original baseline's weak result reflected its construction rather than a fundamental inability of sequence models to detect these attacks; a fairly-built sequential baseline is a real, if still PRSM-inferior, comparator, and we recommend the interleaved construction not be reused as a comparison point in follow-on work.

Lead-time evidence. Using the reconstructed channel scores and per-flow timestamps (available in CICIDS2017's GeneratedLabelledFlows distribution, not the MachineLearningCSV feature file Table 3 is computed from, since that file carries neither IP nor timestamp columns).[3] Attack flows in the test partition were grouped into 17 per-category incidents (bursts separated by >30-minute quiet gaps), and the first flow at which each detector's score crossed its own validation-selected threshold was compared. The result does not support a temporal-lead claim: against equal-weight blending, PRSM ties exactly on 14 of 17 incidents, trails by one minute on one incident (FTP-Patator), and misses 2 incidents outright (an 11-flow SQL Injection burst and a 3-flow PortScan burst, consistent with the near-zero detection already reported for these categories in Table 4). Against the autoencoder-only baseline, PRSM ties on 13 incidents and is strictly ahead only in the sense that AE-only fails to alert at all during two incident windows that PRSM does catch. At the granularity available in this corpus, we find no consistent evidence of earlier detection; "prioritized" in this paper's framing is best read as producing one composite, actionable score ahead of manual triage of scattered per-channel alerts, not as a measured time advantage over the equal-weight or autoencoder-only comparators.

7 Discussion

Leaving user identity out of the design is deliberate, not incidental. Per-user profiling buys fine-grained attribution at the price of identity-management infrastructure that many organizations cannot fund, cannot operate, or cannot legally run under their privacy regimes. Keying the analysis to the host IP keeps PRSM deployable in all of those settings while giving up little detection fidelity: when credentials are stolen, the malicious behaviour still shows up in the end-

[3] The two CICIDS2017 distributions were independently regenerated from the same captures, so timestamps were attached by positional row alignment per day-file and verified exact (zero feature-value mismatches) across seven of eight files spanning >2.6M rows; the eighth (Thursday morning) required truncating the timestamp file's extra unlabelled trailing rows to match the feature file's row count before alignment.

point's network footprint no matter which account authenticated, so a host-level score remains a sound first-response lens.

Once leakage is controlled for, the composite's discriminative power turns out to sit in exactly two places: temporal divergence (D_6) and the App-ID reconstruction residual (D_2), both weighted to 0.40 by the grid search. D_6 replaces cross-branch user tracking with schedule-pattern analysis and remains the single strongest signal in the composite — removing it costs 0.1387 AUC points. D_2 contributes the complementary application-pattern evidence, worth 0.0542 points on removal. Table 4's per-category breakdown complicates the aggregate story, though: the gap between threshold-based detection and ranking quality described in Sect. 6.2 — near-zero single-threshold hits on low-rate credential and web attacks despite informative ranking underneath — is the limitation that actually binds in deployment, not the raw AUC number.

Low-rate attack coverage deserves to be stated plainly as a limitation, not buried as a footnote. At the single global threshold used throughout Sect. 6, PRSM catches low-rate, low-volume attacks badly: 1.1% of FTP-Patator, 0.8% of SSH-Patator, 0.1% of Web Attack–Brute Force, and none of the 11 SQL Injection instances in the test set, against 99.2% for DoS-Slowhttptest and 71.0% for DDoS at that same cut-point (Table 4). We want to be precise about what this does and does not say about the underlying signal. It is not evidence that the composite score has nothing to say about these categories — their mean PRSM scores still rank well above benign traffic, and the aggregate AUC-ROC gains from Sect. 6.1 hold. What it reflects instead is a single institution-wide cut-point being asked to serve attack styles whose score distributions barely resemble each other. Left as-is, a deployment running only this threshold would miss most stealthy credential-guessing and low-and-slow web exploitation in a live network, not just in the benchmark table, and that is exactly the attack profile hardest to catch through other means. Fixing this does not require a new channel, only a deployment-time change: per-category or per-volume-band thresholds. Until that is in place, we consider single-threshold operation validated only for the volumetric and application-signature categories where Table 4 already shows strong hit rates.

Beyond this, acknowledged constraints include the following: (i) a 30-day calibration window before scoring even begins; (ii) susceptibility to DHCP-driven IP churn; (iii) current alignment with Palo Alto log schemas, although the methodology generalizes to other vendors; (iv) the necessity of periodic recalibration to track legitimate behavioral evolution; (v) because CICIDS2017 carries no native App-ID or WildFire telemetry, D_2 and D_4 are computed from generic flow-level proxies (Sect. 5.1) rather than genuine application-classification or malware-sandbox signals, so their CICIDS2017 performance should not be read as validating their behavior against the real Palo Alto App-ID/WildFire output; (vi) per the ablation (Table 5), D_1, D_3, D_4, and D_5 add no measurable standalone detection on this benchmark, so what stands validated here is a two-channel (D_2+D_6) detector rather than a six-channel fusion; and (vii) the lead-time check in Sect. 6.4 found no consistent temporal advantage over equal-weight blending

at the flow-arrival granularity available in this corpus, so "prioritized" should be read as one composite score ahead of manual alert triage, not as measured earlier detection.

8 Conclusion

We have presented PRSM, a host-anchored composite metric for quantifying security risk across distributed NGFW deployments. By fusing six telemetry channels into one weighted value, PRSM delivers prioritized, actionable risk visibility without imposing user-identity requirements. Under a leakage-controlled protocol—coefficients and thresholds tuned on a validation partition, results reported once on a disjoint test partition—PRSM attains an AUC-ROC of 0.9033 on CICIDS2017, surpassing every standalone channel, equal-weight blending, and the single-model references, though its signal concentrates in the App-ID and temporal channels and single-threshold detection of low-rate credential and web attacks remains an open weakness. Its host-centric, identity-free design is architecturally deployable in any NGFW-instrumented environment without identity infrastructure, and this deployability claim rests on the design alone, not on demonstrated effectiveness: validated detection performance is presently confined to the two channels (App-ID anomaly and temporal divergence) confirmed on CICIDS2017 proxy features, and broader claims of operational effectiveness across the full six-channel design await confirmation on native, multi-branch NGFW telemetry.

As recent surveys confirm [21], machine-learning pipelines must be validated under realistic, multi-vector traffic to demonstrate operational readiness. Planned extensions accordingly encompass live validation on production multi-branch telemetry with injected attack scenarios, online coefficient adaptation, cross-vendor log normalization, and end-to-end integration with SOAR orchestration platforms for automated containment workflows.

References

1. Verizon, "Data Breach Investigations Report – 2023 Edition," Verizon Enterprise, Tech. Rep., 2023.
2. Palo Alto Networks, "PA-Series NGFW Platform Overview," Palo Alto Networks, Tech. Rep., 2023.
3. M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," in Proc. IEEE CISDA, 2009, pp. 53–58.
4. S. Axelsson, "The base-rate fallacy and the difficulty of intrusion detection," ACM Trans. Inf. Syst. Security, vol. 3, no. 3, pp. 186–205, Aug. 2000.
5. R. Sommer and V. Paxson, "Outside the closed world: On using machine learning for network intrusion detection," in Proc. IEEE S&P, 2010, pp. 305–316.
6. G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," Neural Comput., vol. 18, no. 7, pp. 1527–1554, 2006.
7. S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural Comput., vol. 9, no. 8, pp. 1735–1780, 1997.

8. M. Sakurada and T. Yairi, "Anomaly detection using autoencoders with nonlinear dimensionality reduction," in Proc. MLSDA Int. Workshop, 2014, pp. 4–11.
9. Y. Mirsky, T. Doitshman, Y. Elovici, and A. Shabtai, "Kitsune: An ensemble of autoencoders for online network intrusion detection," in Proc. NDSS Symp., San Diego, CA, 2018.
10. M. Du, F. Li, G. Zheng, and V. Srikumar, "DeepLog: Anomaly detection and diagnosis from system logs through deep learning," in Proc. ACM SIGSAC Conf. Computer and Communications Security (CCS), 2017, pp. 1285–1298.
11. S. Zavrak and M. İskefiyeli, "Anomaly-based intrusion detection from network flow features using variational autoencoder," IEEE Access, vol. 8, pp. 108346–108358, 2020.
12. P. Garcia-Teodoro, J. Diaz-Verdejo, G. Maciá-Fernández, and E. Vázquez, "Anomaly-based network intrusion detection: Techniques, systems and challenges," Comput. Security, vol. 28, no. 1–2, pp. 18–28, Feb. 2009.
13. V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," ACM Comput. Surv., vol. 41, no. 3, pp. 1–58, 2009.
14. T.-F. Yen et al., "Beehive: Large-scale log analysis for detecting suspicious activity in enterprise networks," in Proc. Ann. Comput. Security Appl. Conf. (ACSAC), pp. 199–208.
15. J. Kindervag, "Build security into your network's DNA: The zero trust network architecture," Forrester Research, Tech. Rep., Nov. 2010.
16. S. Rose, O. Borchert, S. Mitchell, and S. Connelly, "Zero trust architecture," NIST SP 800-207, 2020.
17. A. Aleroud and G. Karabatis, "Contextual information fusion for intrusion detection: A survey and taxonomy," Knowl. Inf. Syst., vol. 52, no. 3, pp. 563–619, 2017.
18. I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," in Proc. Int. Conf. Inf. Syst. Security Privacy (ICISSP), 2018, pp. 108–116.
19. A. L. Buczak and E. Guven, "A survey of data mining and machine learning methods for cyber security intrusion detection," IEEE Commun. Surv. Tutor., vol. 18, no. 2, pp. 1153–1176, 2016.
20. M. Ring, S. Wunderlich, D. Scheuring, D. Landes, and A. Hotho, "A survey of network-based intrusion detection data sets," Comput. Security, vol. 86, pp. 147–167, Sep. 2019.
21. G. Apruzzese, P. Laskov, E. Montes de Oca, W. Mallouli, et al., "The role of machine learning in cybersecurity," Digital Threats: Res. Pract., vol. 4, no. 1, pp. 1–38, Mar. 2023.
22. H. Liu, B. Lang, M. Liu, and H. Yan, "CNN and RNN based payload classification methods for attack detection," Knowl.-Based Syst., vol. 163, pp. 332–341, Jan. 2019.
23. M. A. Ferrag, L. Maglaras, S. Moschoyiannis, and H. Janicke, "Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study," J. Inf. Security Appl., vol. 50, Feb. 2020.
24. A. Aldweesh, A. Derhab, and A. Z. Emam, "Deep learning approaches for anomaly-based intrusion detection systems: A survey, taxonomy, and open issues," Knowl.-Based Syst., vol. 189, Feb. 2020.

Open Access This chapter is licensed under the terms of the Creative Commons Attribution-NonCommercial 4.0 International License (http://creativecommons.org/licenses/by-nc/4.0/), which permits any noncommercial use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

