Published by :
https://www.ijert.org/
An International Peer-Reviewed Journal

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 10 , October - 2026

A Resilience-Aware Intrusion Detection Framework: Quantifying Firewall Endurance Under Sustained Cyber Attacks using Machine Learning

N Ravi, Nikethan S R, Mugileshwar M G, Tharun J, Dhevaa N, Simeon Abraham B

Department of Information Technology, PSG Polytechnic College Coimbatore,

Tamil Nadu ,India

Abstract - Most of the existing intrusion detection systems are evaluated according to their ability to detect malicious activity with proper labeling. It is clear that such evaluation only tells whether the attack was detected or not. It gives no information about the defense resilience to the attack and the reaction time of the firewall. While the previous RF NRA approach tried to go beyond labeling by evaluating the risk level with the help of a random forest on NSL KDD dataset without any interaction with a real defense and staying offline. The current research is based on this idea but focuses more on the resilience measurement. The approach uses isolation forest to detect unusual traffic based on its learning from normal traffic and random forest to make a decision about the attack. Based on this decision, a severity level determines the type of response such as monitor, throttle, block or isolation. The firewall is implemented with the use of nftables to apply the rules which expire after some period of time and also include allowing list for rollback. Everything is logged with time stamp. The value of FRS is calculated every second on the basis of the log and it includes several measures including reaction time and service availability. Such measures as attack endurance time describe the time when the value is above some threshold while other measures evaluate the decline speed and recovery of the system. Evaluation will take place in a laboratory environment with five machines playing roles of hosts, IDS, firewall and attacker and re-played data. At least ten paired experiments are planned against default firewalls and signature-based systems. No results were produced at the moment, but they should be ready soon. It seems that such approach can tell more than just accuracy, although I am not completely sure about the exact correlation of measures.

Index Terms - Intrusion detection system plays a role here.

Firewall resilience and attack endurance are correlated somehow. Isolation Forest and Random Forest play their part in detection process, automated mitigation with note able can be used. Metrics of resilience must be considered.

1. INTRODUCTION

Most intrusion detection systems attempt to detect malicious traffic and classify the type of attack. The former approach uses rules-based IDSs such as Snort while the latter relies on either traffic normalcy or labeled training data. People commonly evaluate their performance using accuracy and other similarity metrics on partitioned data sets although it is not sufficient to reflect the real behavior.

The RF-NRA paper proposed a different approach in which authors transformed conventional attack categories into ordered risk values ranging from safe to compromised. A random forest algorithm predicted risk associated with each record while an escalation rule detected the presence of three increasing windows in a row. Although such approach seemed more focused on estimating risk than classifying the attacks themselves, its performance significantly decreased on the challenging data set and no attacks were classified as the highest risk since they were absent from the training data set.

That is why authors suggested using isolation forests to detect rare attacks. The problem with existing approaches is that none of the scores reflects the impact of detection on defense capabilities. Even two systems with identical F1 metrics can perform differently by being faster or slower, blocking too much legitimate traffic. The length of attack worsens the problem because both logs and tables continue growing during an attack. Attacks at low rates in particular are sensitive to timing.

This paper extends the concept of risk and evaluates overall defense resilience against different attacks in terms of time. To do this, it uses an isolation forest-based gateway and a random forest classifier along with some firewall rules and records the process using timestamps. Based on this information it computes resilience and survival metrics. The experiment takes place within a controlled laboratory environment in which attack length and power can be adjusted manually. The rest of the paper discusses previous research, describes the design and metrics, and presents experiments.

2. RELATED WORK

2.1 Random Forest for Intrusion Detection

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 10 , October - 2026

Random Forest is a technique that employs multiple decision trees, using bootstrapped samples while restricting each split to a random subset of features. This approach decorrelates the trees and reduces variance. Zhang et al. utilized this method to systematically assess misuse, anomaly, and hybrid network intrusion detection on the KDD'99 dataset. The RF NRA framework highlighted four key motivations for using Random Forest. It inherently manages multi-class classification, determines class probabilities based on tree vote fractions, naturally measures feature importance through impurity reduction, and benefits from existing Random Forest intrusion detection system (RF-IDS) research for comparison. It applied balanced class weights in a training set significantly skewed towards SAFE records over COMPROMISED records by more than a thousand-to-one. Additionally, the top features, such as src_bytes and dst_host_srv_count, align with the traffic volume and service diversity characteristic of DoS and probe attacks. Overall, tree ensembles continue to perform well on heterogeneous tabular data, akin to flow and window statistics, which is also the case with the classifier used here.

2.2 Limitations of Binary and Nominal Classification

The RF-NRA paper indicated that within the nominal metrics framework, misidentifying a probe as U2R was considered as significant as misclassifying it as normal traffic, even though the latter error was much more critical. In the ordinal version, however, the difference between two types of errors was measured; 73.32% of test cases were accurately predicted, and 74.60% only deviated by one level. Although the ordinal variant altered how the classifier presented its results, the evaluation process continued to be conducted offline and based on records.

2.3 Temporal Constraints in IDS Evaluation

Certain assumptions in machine learning-based intrusion detection are not practical for real-world operations. Since attacks are infrequent, even a small number of false positives can overwhelm the alarm system. Common errors include temporal snooping and the use of inappropriate metrics. Another concern is the quality of datasets. For instance, while NSL-KDD has removed duplicate entries from the KDD'99 dataset, it doesn't include packet captures or timestamps. On the other hand, CIC-IDS2017 contains packet captures and an attack schedule, but labeling and flow creation errors are documented. Some algorithms show strong performance on specific datasets but fail when applied to different network traffic. Additionally, much of the literature either omits time entirely or eliminates it through random splitting, making it challenging to assess how a defense mechanism responds during an attack.

2.4 Resilience-Based Defense Evaluation

According to NIST SP 800-160 Vol. 2, cyber resiliency refers to “the ability to anticipate, tolerate, adapt to, and rapidly recover from adverse conditions”. Typically, in systems engineering, resilience is estimated using the performance curve throughout the disruptive event, e.g., through the area of functionality loss. Kott and Linkov maintained that cyber resiliency should be measured quantitatively prior to its improvement. Moreover, in, Weisman et al. proposed infrastructure for repeatable quantitative resilience experimentation on a cyber-physical platform. Catillo et al. evaluated web-server denial-of-service defenses based on their effects on the server. In this study, the concept of performance curve from the above works will be used for evaluating the ML enforcement point.

2.5 Automated Mitigation

The intrusion response systems convert detections into actions, which involve a trade-off between speed and the risk of an erroneous reaction. Toth and Kruegel analyzed the effect of automated response on legitimate users, and Ligo etal. claimed that autonomous cyber defense creates a threat that should be addressed. In this research, the important thing is that the mitigation could be the cause of an outage that the mitigation intended to avoid, which means that the mitigation layer should be considered in the process of defense assessment. Table 1 illustrates how the suggested research connects with traditional IDS and with RF-NRA.

Machine learning based intruder detection, random forest classification, automated firewall responses, and cyber-resilience are all well-trodden research areas. But the gap filled by this paper is narrower and concerns the evaluation of these components. There are three gaps found in the literature survey above – all of which describe current practice, and thus do not imply any lack of research on them.

Gap 1: Lack of composite, time-resolved defense outcome measure. Composite metrics are available for vulnerability severity and resiliency engineering practice, and resilience can even be derived from performance curves. However, there is no widespread index openly specified and used that combines detection time, blocking efficiency, cost of false positives, availability, recovery, resource headroom, and throughput that can be computed on a second-by-second basis from firewall data.

Gap 2: Methodology ignores the time dimension. Random splits of training/validation/test datasets based on flow records make degradation that depends on duration impossible to observe. Even the timestamped datasets are usually consumed as feature tables with no regard to time, and RF-NRA was unable to validate its escalation rule due to this fact.

Gap 3: Metrics for detection and mitigation results are usually considered independently. IDS research usually uses the F1-score, whereas mitigation research reports throughput and availability. The relationship between them, like how a rise in detection latency or false positive probability leads to availability loss, is almost never quantified in one system.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 10 , October - 2026

TABLE 1
COMPARISON OF EXISTING IDS, RF-NRA, AND THE PROPOSED APPROACH

Aspect	Conventional IDS	RF-NRA	Proposed framework
Question	Is the traffic malicious, and which at-tack is it?	How risky is the current traffic/state?	How well does the defense keep functioning while the attack persists, and for how long?
Output	Alert or class label	One of six ordinal risk states; escalation trend	Response tier and enforcement action; FRS(t), AET, temporal metrics
Models	Signatures or an ML classifier	Random Forest	Isolation Forest, Random Forest, defined severity con-struct
Data	Public datasets, usually shuffled	NSL-KDD, 41 features	Sliding-window features from CIC-IDS2017 replay and controlled laboratory traffic; NSL-KDD for continuity
Time	Usually removed by random splitting	Escalation rule defined but not validated on ordered data	Attack duration is a controlled experimental variable
Enforcement	Separate component, not evaluated with detection	Not included	nftables mitigation with safeguards; self-inflicted denial measured
Evaluation	Accuracy, precision, recall, F1, AUC	Macro-F1, ordinal error, per-tier ROC-AUC	Classification metrics plus latency, availability, throughput, resources, recovery, FRS, AET

3. THE CONTRIBUTION OF THIS RESEARCH IS AN INTEGRATION FRAMEWORK AND MEASUREMENT APPROACH THAT ENTAILS THE FOLLOWING:

1. Multi-staged pipeline consisting of Isolation Forest, Random Forest, severity construct, and mitigation where each stage issues timestamps enough for detecting, classifying, deciding, enforcing, and recovering.
2. The Firewall Resilience Score is a bounded, non-compensatory, sliding-window evaluation of defensive success that uses published constants and weights through sensitivity analysis.
3. Duration of Endurance Against Attack and related temporal measures (Normalized endurance, degradation rate, resilience half-life, resilience area, recovery time) based on attack onset.
4. A duration controlled experimental approach wherein attack time and intensity are independent variables with ten or more repetitions per condition.
5. Direct measure of self-denial inflicted due to the mitigation layer itself.

4.PROPOSED SYSTEM ARCHITECTURE

The proposed system architecture is shown in Fig. 1. The traffic originating from the attack generator (VM1) and a legitimate client (VM2) is received by the defense node (VM3), which then forwards any traffic allowed by the defense node to the protected server (VM4). VM3 contains all components of the pipeline: traffic inspection, sliding window feature generation, the Isolation Forest gate, the Random Forest classifier, severity estimation, firewall policy decision, and nftables mitigation. Each step reports its events to the telemetry bus with timestamps. The telemetry is stored, FRS, AET, and temporal metrics are computed, and the dashboard is served by the measurement node (VM5).

There are two key aspects of the design in Fig. 1 for our measurements. First, VM2 carries out an availability probe at a low frequency in addition to benign background workload. The availability probe pretends to be a legitimate user, submits requests through the firewall to the protected service, and returns to the measurement node whether the request was successful or not. Its success rate reflects whether legitimate users are being served. Second, the measurement engine is running on VM5, not on VM3. If FRS was computed on the defense node, it would require the same amount of CPU and memory that the resource headroom component is supposed to measure.

The table below outlines the learning/decision process in three stages. The Random Forest model operates exclusively on windows passing through the Isolation Forest gate, thus ensuring that the classifier does not get on the path for normal traffic.

5. PROPOSED METHODOLOGY

5.1 Traffic Monitoring and Feature Extraction

Packets will be collected from the input interface of VM3 using AF_PACKET, and the features will be extracted every second using sliding windows of 1 s, 5 s, and 30 s duration. The set of features will include number of packets and bytes, number of new connections per second, ratio of SYN/ACK, RST, and FIN packets, mean and standard deviation of packet inter-arrival times and packet sizes, Shannon entropy of source IP addresses and destination ports, fraction of traffic from the most active sources, and number of half-open connections. They are similar to the time and host-based traffic features extracted by RF-NRA for NSL-KDD, which summarize the connections from the previous 2 seconds and 100 connections to each host, respectively.

5.2 Anomaly Detection

Stage 1 implements Isolation Forest, an algorithm which isolates outliers by constructing random trees and calculating the average path length of each data point, where anomalies tend to have shorter path lengths due to their isolation in a smaller number of partitions. The model is trained on the normal traffic samples extracted during the calibration procedure (Section 7.2). The parameter configure ration was chosen according to the recommendation of 100 trees and sub-sample size of 256. The threshold value was selected on the

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 10 , October - 2026

validation set of normal data only, where the 99 percentiles of normal scores corresponds to 1% false alarm rate, thus avoiding any information about attack classes during threshold selection.

The motivation behind this choice was driven by the observation from RF-NRA that a closed set Random Forest classifier failed to distinguish unseen attacks of the CRITICAL level from normal traffic, misclassifying them as SAFE. On the contrary, Isolation Forest is able to identify traffic patterns that deviate from normal, regardless of whether they were seen during training or not, thus addressing the zero-day attack scenario. Nevertheless, this approach cannot guarantee complete protection against novel attacks, as it might fail to isolate attacks with traffic patterns similar to normal or benign traffic fluctuations. Hence, the 30 s time window was introduced to reduce benign drift impact, and the moment of gate opening is registered as a detection trigger for Stage 2.

5.3 Classification of Attacks

The second stage is a Random Forest classifier with seven target classes: DoS/DDoS, Port Scan, Brute Force, Web Attack, Infiltration, Botnet, and Benign. The arguments given by RF NRA for Random Forest (Section 2.1) are applicable to this scenario. The proposed design consists of 200 trees, the square root of the number of features to split on, a minimum of two samples per leaf, and balanced-subsample class weights. RF NRA found that on NSL-KDD .The cross-validated macro-F1 plateaued between 100 and 150 trees; a similar sweep will be performed on the new features prior to fixing the tree number.

The classifier outputs the classification result, the probability vector, and classify. If the alarm is fired, but the output of the classifier indicates Benign with high probability, then the window is recorded as a candidate false positive and no escalation takes place. The six RF-NRA risk states are not output classes because they represent the risk of traffic and not the attack type.

The training data consist of windows that have been re-extracted from the CIC-ID S2017 packet traces [14] via the extractor from Section 5.1, labelled using the CIC-IDS2017 labels where available. Windows are very similar, which means that a random split will leak information between training and testing. Since each of the attacks in the CIC-IDS2017 occur on a certain day, each attack session has been split temporally (first 60% for training, then 20% for validation, last 20% for testing) using a 30 s delay between partitions. Macro-F1 is the main performance measure used here, since accuracy can be misleading by class imbalance, which was demonstrated by the RF-NRA where the deepest trees were shown to have the best accuracy but the worst cross-validated macro-F1.

The RF-NRA setting, consisting of 100 trees, class balance, fixed random seed, and stratified 5-fold CV based on the 41 NSL-KDD features, is independently replicated as an implementation baseline. This verifies that the replication reproduces the previous classifier. It is not used for resiliency assessment because NSL-KDD does not include any packet captures or timestamps and has different classes from those in the above list..

5.4 Severity Analysis

The public IDS data sets do not have ground truth about severity labels, hence these are considered in the sense of a published scheme, not a learned one.

The basic scheme is

$$SEV = 10 \cdot (I \cdot C \cdot R \cdot P)1/4$$

with $I \in [0, 1]$ being the effect of the episode, the result of a configured weight times the criticality of the asset under attack; C is the maximum probability of the class as performed by the classifier; R is the ratio of the actual attack rate and service capacity during calibration, normalized to; and P is the duration of the episode normalized to a configured persistence threshold, normalized to [0, 1]. The geometric mean is intended: if any of the components are close to zero, then the severity remains low, so an unconfident detection on a low value asset will not trigger isolation.

The other two input variables, service availability and resource headroom, characterize the defense situation not the attack. They are utilized as the escalation rule, which raises the tier of defense by one until the ISOLATE tier if during the active episode the availability sub-index s is below the threshold of service level objective and the resource headroom sub-index s is below the configurable floor. All parameters listed above are fixed during calibration. The Gradient Boosting Regressor can be trained on this rubric for speeding up the calculation process.

The four levels of responses listed in Table 3 are independent of the six states of RF-NRA risk. The states indicate the proximity to traffic being compromised, while the tiers define the firewall response.

Table 2

SEV	Tier	Enforcement Action	Rule TTL
00.0–2.5	**MONITOR**	Log only; raise sampling	—
2.6–5.0	**THROTTLE**	Per-source rate and connection limit	**120 s**
5.1–7.5	**BLOCK**	Add source to notables drop set	**300 s**
7.6–10.0	**ISOLATE**	Block source range; cap service connections; alert	**Configured**

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 10 , October - 2026

5.5 Firewall Mitigation

The enforcement system works by utilizing notables, which are in VM3. The blocked sources are elements of named sets with timeouts defined individually for each entry. Rate limiting is enforced by meters, and connection counting is achieved by connection-count expressions. Suricata is run in the IPS mode as part of the baseline configuration and can be used in conjunction with the proposed pipeline. The mitigation controller updates the firewall via the notables library API to avoid the overhead of forking a new process for each rule, which would unnecessarily add to the firewall's latency. The t-enforce field is the time at which the rule takes effect.

Automated mitigation presents a risk to the protected service if the classifier fails. The controller. Therefore, implements four protections for such cases. First, all dynamic rules have a time-to-live (TTL) value, after which the rule is deleted. Second, no allowlist entries can be blocked, including the management interface, gateway, and availability probes. Third, there is a maximum number of rules that can be inserted per second to prevent large-scale effects of a classifier error. Finally, the last mitigation-specific update to the firewall always rolls back if the probe's availability degrades within a short time after the enforcement started while no increase in attack traffic was observed. The self-denial event is then logged as a defensive failure.

5.6 Telemetry Collection

Every attack scenario has seven timestamps:(the first time malicious packet comes to the VM3 input interface, either from attack controller or the replay schedule) (selected tier), (last malicious packet) and Recovery is considered to occur if the attack scenario is detected at the beginning of the first triplet of windows succeeding where throughput and availability metrics fit into the calibrated tolerances of their baselines.

Latency is computed based on the following timestamps:
The decision latency, differentiates the severity and policy overhead from the kernel commit time. If an attack scenario is not detected or fails to recover within the observation period, its associated period will be considered as right-censored.

where τ is the latency tolerance. The default value of τ is 5 seconds, at which $s \approx 0.37$ when L = tolerance. Before the gate triggers, the exact value of latency is not yet known but is at least the duration of elapsed time since the attack onset, hence during [t, t], the elapsed time is used in.

It causes FRS to decrease during the period in which the attack is not yet detected rather than hide the attack. Blocking effectiveness and retaining precision. s is the fraction of malicious packets in the sliding window that is denied access by the firewall. It is the recall of the enforcement layer and not the detector layer, and the gap between the two represents the amount of attack detected that was also blocked. s = 1 – FPR, where FPR is the false positive ratio of the benign packets in the sliding window. When there are no attack, s and s are set to 1.

where τ is the latency tolerance. The default value of τ is 5 seconds, at which $s \approx 0.37$ when L = tolerance. Before triggering the gate, the precise value of latency has not been established yet, but it is at least the duration of the elapsed time from the start of the attack, thus during [t, t], the elapsed time is substituted into (7). It leads to reduction in FRS within the time interval that the attack has not yet been detected, not concealment of the attack. Blocking efficiency and precision preservation. s is the proportion of malicious packets in the sliding window that is rejected by the firewall. It is the recall of the enforcement layer and not the detector layer, and the difference between the two is the portion of the attack detected and also blocked. s = 1 – FPR, where FPR is the false positive rate of the benign packets in the sliding window. If there are no attack, s and s are both assigned 1.

where τ is the latency tolerance. The value of τ is 5 seconds by default, and hence s is approximately 0.37 for L = tolerance. Before opening the gate, the exact value of the latency has not yet been found out, but it is at least the amount of time that has elapsed since the initiation of the attack, and therefore during [t, t], the elapsed time is used in. It results in lowering the value of FRS in the time interval in which the attack has not been detected, not the concealment of the attack. Efficiency of blocking and maintaining precision. s is the ratio of malicious packets among those rejected by the firewall in the sliding window. It is the recall value of the enforcement layer and not that of the detector layer, and the difference between the two values is the fraction of attack detected and also blocked.

s = 1 – FPR, where FPR is the false positive rate of the benign packets in the sliding window. When there is no attack, both s and s get the value of 1.

6. EVALUATION OF FIREWALL RESILIENCE

The FRS measures the "state" of the defense process at any particular moment of time. It is calculated once per second on the VM5, using a trailing window of W = 10s. For each of the parameters it is normalized to be in [0, 1], with 1 being the best case. And all the constants are calibrated once and locked in version control before an attack experiment so that they cannot be tuned after an experiment has been performed.

Components and their weights proposed for headline score are included in Table 4. There are some conventions that need to be followed in order to define the components every second from the recovery objectives during calibration [value to be determines during calibration]. Resource headroom includes utilization of CPU, memory, and connection tracking on the defense node: with suggested weights $\alpha = 0.5$, $\beta = 0.3$, $\gamma = 0.2$. This is the part which is expected to show the long duration failure: the defense node at very high CPU usage is near to failure even as it is still defending. The throughput retention is $s = \min(1, \Theta / \Theta)$, where the goodput baseline is determined during calibration

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 10 , October - 2026

TABLE 3
FRS COMPONENTS (PROPOSED WEIGHTS AND DEFINITIONS)

Sym.	Component	Definition over window W	Weight	Reason for inclusion
S_A	Service availability	Successful / attempted availability-probe requests	0.20	The purpose of the defense; measured independently of the firewall
S_B	Blocking efficacy	Malicious packets denied / malicious packets received	0.20	The defense's direct effect on the attack; enforcement recall, not detector recall
S_L	Detection promptness	$\exp(-L_d / \tau_L)$; elapsed time used before detection	0.15	Sets the length of the unprotected exposure window
S_F	Precision preservation	1 − FPR on benign traffic	0.15	False positives lead to self-inflicted denial
S_R	Recovery capability	$\exp(-T_{rec} / \tau_R)$; carried forward during an attack	0.10	Residual harm after the attack ends
S_Θ	Throughput retention	$\min(1, \Theta_{attack} / \Theta_{baseline})$	0.10	Quality of service, not only reachability
S_U	Resource headroom	$1 - (0.5U_{cpu} + 0.3U_{mem} + 0.2U_{ct})$	0.10	Leading indicator of failure in long attacks

6.2 FRS Formulation

The main index score is the weighted geometric mean of the components:

FRS(t)=100·A(t)^{0.20} ·B(t)^{0.20} ·L(t)^{0.15} ·F(t)^{0.15} \c dots

Each element being floored at ε = 0.01 to prevent a single zero value from causing the score to be 0 and thus remove all diagnostic information from the score. The geometric score restricts the amount of compensation between elements. For example, if availability s = 0.1 while all the other values are 1, then the score for the arithmetic form is 82 but the geometric score is 100 · 0.1 0.20 ≈ 63. Good blocking cannot hide the fact that a service is highly unavailable. The arithmetic form is included as a diagnostic measure alongside it.

FRS (t) = 100 · Σ w s(t)

A big difference between only one component is sick while the others work well. While the sole use of the geometric mean allows for a good score even with low availability, as demonstrated by the example below, the proposal introduces an essential availability gate. If s < 0.5 in the window, FRS(t) is limited to 49. Such a defense, under which the majority of valid requests is rejected, cannot be considered successful no matter what its other characteristics say. Results are presented both with and without the gate. To report the results, FRS is converted into bands based on the proposal:

90 – 100 Excellent, 75 – 89 Strong, 50 – 74 Moderate, 25 – 49 Weak, and 0 – 24 Critical resilience.

Weights for are a matter of design choice, and are based on Table 4 component-by-component. The base formula is not changed, and its sensitivity is tested using three different configurations of weights: equal weights (1/7 each); availability-dominated weights (0.40, 0.15, 0.10, 0.10, 0.10, 0.10, 0.05); and a detection-dominated set of weights for this study (s 0.30,s 0.25, s 0.15, s 0.10, s 0.10, s 0.05, s 0.05). Tolerance τ is also varied at ±50%. Each configuration's report notes when ranking of the defense

Configurations are changed. To be tested experimentally. Each element is minimized to ε = 0.01 to prevent a perfect zero from collapsing the value to exactly 0 and removing all diagnostic capability. The geometric formulation restricts the extent of compensation among dimensions. For example, if availability falls to s = 0.1 and all other elements are 1, then the weighted arithmetic mean will still be 82, but (9) results in 100 · 0.1 0.20 ≈ 63. Even excellent blocking cannot disguise a degraded service. The arithmetic formulation

6.3 Attack Endurance Time

The FRS shows the present state of the system; the AET shows the duration that the acceptable state was maintained. This is the longest period from the time of the attack to when the FRS stays above a certain value of θ:

AET(θ) = max{ Δ : FRS(t) ≥ θ for all t ∈ [t , t + Δ] }

where t = t. The value of AET is known at θ = 90, 75, and 50. If FRS is never less than θ until the end of the attack, AET(θ) is equal to D and is denoted as right-censored because the defense could have lasted longer if the attack had continued. The endurance of an attack is normalized to allow comparison of attacks of different durations:

NAET(θ) = AET(θ) / D ∈ [0, 1]

Since AET begins right from the moment of the attack, it captures the pre-mitigation phase. In light of the pre-detection approach defined in Section 6.1, the decrease in FRS begins right from the moment the attack commences; hence AET (90) and AET (75) could be very small in many simulations. This was done purposefully since it captures the exposure prior to any defense being implemented against the attack. To ensure the separation of such an exposure from the behavior of the mitigation when put in place, the mitigation relative AET (θ), which is measured from t m enforce instead of t,0 is provided.

6.4 Temporal Resilience Metrics

There are four additional measures that define the pattern

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 10 , October - 2026

of the FRS trajectory. The degradation rate is the average loss of FRS per second during the attack.

$$\rho = (FRS(t) - FRS(t + D)) / D$$

The half-life of resilience is defined by how long it takes for the FRS to fall to half the value at onset, and if this does not occur before the attack concludes, then it is considered right censored.

$$t = \min\{ t - t : FRS(t) \leq FRS(t) / 2 \}$$

The half-life of resilience is the period needed for the FRS to reach a half the level reached at the time of onset, and if that does not happen before the end of the attack, then it is right-censored.

$$RAUC=\frac{1}{D} \int_{t_0 }^({t_0+D}FRS(t)\,dt)$$

Moreover, the recovery time T as defined in represents any lasting damage caused by the attack, such as saturation of the connection tracking table or rules that persist even after the attack. FRS(t) indicates the level of success of the defense, AET the duration, ρ and t the speed of erosion, RAUC the whole attack, and T its consequences.

7. EXPERIMENTAL METHODOLOGY

7.1 Lab Environment

All experiments are performed in a virtual lab with 5 machines connected to each other through virtual networks exclusively, without NAT, bridged adapter and without any connection to any external networks: VM1 (attacker and traffic generator), VM2 (benign client and availability probe), VM3 (defender with two interfaces and IP forwarding enabled), VM4 (protected web server with nginx and web application & small database) and VM5 (measurements and dashboard). The resources of VM3 are purpose fully set to be modest (2 vCPUs, 4 GB RAM), in order to easily saturate the system at the tested intensities, while virtual CPUs are pinned, to have comparable results for CPU usage measurements. Before each experiment isolation is verified, by checking from all VMs that no external addresses and DNS are reachable, and it is recorded. Each test starts from the same VM snapshots.

TABLE 4
EXPERIMENTAL SCENARIOS (EACH RUN UNDER B1, B2, B3, P1, P2 WITH ≥10 PAIRED REPETITIONS)

ID	Scenario	Traffic	Intensity	Duration	Main measurements
E1	Benign control	Benign load + probe only	—	30 min	FPR, false alarms per hour, baseline FRS (should stay in the Excellent band if calibration is correct)
E2	Duration sweep	Single-vector flood (Mode B)	Medium	30, 120, 300, 900 s	L_d, response latency, FRS(t), AET, NAET, ρ, $t½$, RAUC, T_{rec}; exposure when L_d approaches D
E3	Intensity ramp	Volumetric flood (Mode B)	Stepwise until FRS < 25 or cap-true drops	Fixed per step (set in calibration)	Response latency, availability, CPU, contract saturation, breaking point
E4	Concurrent attacks	Port scan + HTTP flood + brute force, overlapping	Medium	Set in calibration	Per-class confusion matrix, tier oscillation, rule conflicts, FRS(t)
E5	Repeated attacks	5 pulses of 60 s, 120 s gaps	Medium	5 × 60 s	Per-pulse L_d, minimum FRS, T_{rec}; rule-set and memory growth
E6	Low-and-slow (optional)	Slow-rate HTTP (Mode B)	Low	900 s	Detection with 30 s windows, availability, AET
E7	Dataset replay	CIC-IDS2017 and CIC-DDoS2019 captures (Mode A)	As recorded	As recorded	Classification metrics under replay, FRS(t), latencies

7.2 Dataset and Traffic Sources

Three tracks are considered independent:
RF-NRA continuity on NSL-KDD ; detection, which trains the classifier on CIC-IDS2017 windows (Section 5.3) and evaluates it against datasets from UNSW-NB15, due to the problems with transferability of classifiers; and resilience, which has two traffic modes from the live laboratory.

Mode A (dataset replay) replays packet captures from CIC-IDS2017 and CIC-DDoS2019 [30] for high-volume traffic, using pureplay with original timing. This mode provides realistic traffic, but the duration of attacks is fixed with the capture.

Mode B (controlled generation) generates traffic in the laboratory with exactly defined start and end of every attack episode: volumetric floods (hping3), HTTP floods (Locust or work), slow-rate HTTP (slow http test), port scans (n map), and credential brute force against the laboratory account only (hydra). The controller logs the type of attack, attack intensity, start and end time for every episode into telemetry store – this is the ground truth for Mode B. The watchdog stops every generator when its time has come to an end. Mode B defines attack duration as an experimental variable.

7.3 Scenarios' Configurations

The table 5 presents the matrix of scenarios. Scenario E2 is the main scenario in the sense that both the type of attacks and their intensities are fixed and only the duration varies.

7.4 Baseline Configurations

Each of the scenarios has been analyzed using 5 baseline configurations. There are B1 - notables rule set, then there is B2 - Suricata + Emerging Threats Open rules in IDS mode (alerts only). Also there is B3 – Suricata in IPS mode + notables. Plus there are two proposed pipelines: P1 - no changes applied and P2 - availability/headroom escalation rule applied with no roll back check, which makes it possible to extract the feedback signal from the defense state. It should be noted that the telemetry and resilience engine are the same for all configurations, hence the

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 10 , October - 2026

possibility to compare AET and FRS.

7.5 Replicated Experiments

For each scenario-configuration pair, the replicates are carried out at least 10 times using a configuration pair for each scenario. For each configuration pair, the same replay file or generator seed and the same starting snapshot are used. Note that the number of replicates is not random; with 5 configuration pairs, the highest achievable p-value (two-tailed) through the exact Wilcoxon signed-rank test would be 0.0625, which is not statistically significant at the 0.05 level. The replicates involving capture drops or clock offsets are repeated.

8. MEASURES FOR EVALUATION AND ANALYSIS

Three classes of measures are collected. Machine learning measures are calculated for the hold-out partition and replayed traffic: precision, recall, macro F1, class F1,confusion matrix, one vs rest ROC AUC if relevant for that class, and detection and classification latency. Defense measures are obtained through live execution: blocking effectiveness, false positive rate, service availability, throughput retention, 95th percentile service latency, CPU, memory and connection tracking utilization, firewall response latency, recovery time, and self-denial count. Resilience measures are FRS(t) (where FRS is used as the diagnostic test), AET (90), AET (75), AET (50), NAET, ρ, t, RAUC, and T.

The analysis will find correlations between E2 and latency of detection, FRS(t), AET and recovery, between E3 and response latency, availability, and resource consumption, and will find out how close the critical configuration is to the Critical Band; compare P1 to B1-B3 and P2; and compare rank ordering of configurations using the FRS-based diagnosis and rank ordering using macro-F1 on all tests; if the rankings match in all cases, the resilience measure does not give much additional information beyond what a regular evaluation does.

Planned graphs will include an FRS(t) graph for 900 s case, AET(75) and RAUC as functions of duration, availability and response latency as functions of intensity, and resource consumption during the maximum attack; they will be computed by a single program from the results database with error bars based on repeated computations.

9. EXPECTED RESULTS AND DISCUSSION METHODOLOGY

At the moment, there are no experimental results available. The following hypotheses will be shown regardless of whether they are true or not. H1. Latency of detection is not independent from attack time; in a situation where the attack lasts for some time, increased resource consumption on the defense node will result in increased latency. H2. During prolonged attacks, FRS(t) will continuously decrease or have the knee point, whereas macro-F1 will be constant. H3. P1 will have higher AET(75) than B1-B3 at equal intensity. H4. At least in one scenario, the configurations will have different rankings based on FRS and macro-F1. H5. In a situation where the attacks are repeated (E5), the minimum FRS in the last pulse will be lower than the minimum FRS in the first pulse, indicating the degradation. H6. Resource headroom s will decrease before availability s.

The shape of the trajectory of FRS at the 900 s mark is crucial to understanding the result. If the trajectory is flat following the mitigation, it shows the sustainability of the defense; if there is a declining curve, it means exhaustion, and the reasons will be explained by the resource curves; finally, if there is a knee, it shows a capacity threshold, whose source (either CPU usage, connection tracking or rule set size) will be visible in the telemetry. All three results are valuable.H4 is mainly about the metric itself. FRS will not be able to rank configurations differently from macro-F1 – in that case, the claim that the metrics of detection are flawed is unfounded, based on this data. Some of the results will be positive for the baseline: a signature-based system will defeat the windowed ML detector in some attacks within its rules. Comparison of P1 and P2 will let us see whether escalation and rollback are effective or cause oscillations. For the continuity track, the reproduction of the NSL-KDD classifier should produce the CRITICAL tier failure found with the RF-NRA.

10.LIMITATIONS AND THREATS TO VALIDITY

External validity. The laboratory consists of one virtualized defense node and hence the findings cannot prove any behavior in the production environment or real hardware environment. The traffic is generated through a playback of attack traffic or synthetic traffic. Encrypted traffic is not considered, thereby hindering independent detection of any attacks regardless of the payload. CIC IDS2017 data set has known flaws that can affect the model trained on this dataset, while NSL-KDD is the older traffic dataset that only serves the continuity tracks.

Construct validity. The FRS weighting and severity rubric are based on the researcher's choice. The sensitivity analysis has determined the impact of the chosen design, but the design choice still holds. The severity is not validated through any incidents' impact, and the carry forward practice of s as well as the pre detection practice of s affects FRS. Furthermore, the 10 second window also delays FRS adjustment by up to the window duration. Internal validity. There is an existence of scheduling noise due to the virtualization, which can be minimized by pinning, snapshot reset, and randomized ordering. The robustness of the detector against any adversary is not tested.

11. ETHICS AND SAFETY

All attacks were launched from the isolated lab against the lab’s systems. There are no targets anywhere except for the lab and no systems are attacked; the paper does not contain any information about attacks on real-world systems. Isolation is confirmed and documented prior to each experiment, generators are operating under the control of the watchdog, and prior

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 10 , October - 2026

permission from the institution is obtained before the experiment is carried out. The data sets are used in accordance with their terms of use, and there is no personal data in lab telemetry. Regarding the defense, the safety mechanisms discussed in Section 5.5 will protect against the damage caused by automation.

12. CONCLUSION

In this paper, we have presented the Intrusion Detection system that knows about its resilience, being the extension of the RF-NRA technique in terms of moving from the risk estimation produced by the network traffic to the assessment of the reaction on it. The technique we propose includes the anomaly gate based on Isolation Forest, attack classifier with Random Forest, severity measure, notables' mitigation, protection from denial of service produced by itself, and recording of the timings and results of each stage. The telemetry data allows calculating the Firewall Resilience Score that shows the resilience level of the defense for every second and Attack Endurance Time that is the period from the attack beginning till the defense goes below the certain threshold. Degradation rate, resilience half-life, resilience area, and recovery time describe the rest of the trajectory of the resilience behavior. Our contribution in the area is the measurement technique, but not the detection one. Section 7 introduces our experiments that will allow testing whether the duration-controlled measurements provide any additional information comparing to simple detection accuracy.

The proposed framework therefore provides a unified methodology for linking intrusion detection, mitigation, telemetry, and resilience measurement. By recording detection and response timing together with the resulting resilience trajectory, the framework is intended to support evaluation beyond conventional classification accuracy. The planned experimental scenarios will evaluate whether these duration-aware resilience measures provide additional information about defense behavior under varying attack duration, intensity, concurrency, and repetition.

Future work will focus on completing the planned experiments, analyzing the resulting resilience trajectories, and validating the proposed measurements against the defined evaluation metrics.

REFERENCES

[1] M. Roesch, "Snort—Lightweight intrusion detection for networks," in *Proc. 13th USENIX Conf. System Administration (LISA)*, 1999, pp. 229–238.

[2] D. E. Denning, "An intrusion-detection model," *IEEE Trans. Softw. Eng.*, vol. SE-13, no. 2, pp. 222–232, 1987, doi: 10.1109/TSE.1987.232894.

[3] R. Sommer and V. Paxson, "Outside the closed world: On using machine learning for network intrusion detection," in *Proc. IEEE Symp. Security and Privacy*, 2010, pp. 305–316, do: 10.1109/SP.2010.25.

[4] D. Arp, E. Quiring, F. Pendlebury, A. Warnecke, F. Pierazzi, C. Wessinger, L. Cavallaro, and K. Rieck, "Dos and don'ts of machine learning in computer security," in *Proc. 31st USENIX Security Symp.*, 2022, pp. 3971–3988.

[5] G. Apruzzese, P. Laskov, and J. Schneider, "Sok: Pragmatic assessment of machine learning for network intrusion detection," in *Proc. IEEE 8th European Symp. Security and Privacy (Europe)*, 2023, pp. 592–614, doi: 10.1109/EuroSP57164.2023.00042.

[6] N. Ravi, Inban V. N., Dhaarun Kumar K., Dinesh A. C., Harishankar P., and Raja Boopathy R., "Predictive network security assessment using risk analysis and Random Forest (RF-NRA)," Dept. of Information Technology, PSG Polytechnic College, Coimbatore, India, manuscript. [Publication venue, year, and DOI to be completed before submission.]

[7] A. Kuzmanovic and E. W. Knightly, "Low-rate TCP-targeted denial of service attacks: The shrew vs. the mice and elephants," in *Proc. ACM SIGCOMM*, 2003, pp. 75–86, doi: 10.1145/863955.863966.

[8] N. Tripathi and N. Hubballi, "Application layer denial-of-service attacks and defense mechanisms: A survey," *ACM Comput. Surv.*, vol. 54, no. 4, Art. no. 86, 2022, doi: 10.1145/3448291.

[9] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001, doi: 10.1023/A:1010933404324.

[10] J. Zhang, M. Zulkernine, and A. Haque, "Random-forests-based network intrusion detection systems," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 38, no. 5, pp. 649–659, Sep. 2008, doi: 10.1109/TSMCC.2008.923876.

[11] L. Grinsztajn, E. Oyallon, and G. Varoquaux, "Why do tree-based models still outperform deep learning on typical tabular data?" in *Proc. NeurIPS Datasets and Benchmarks Track*, 2022.

[12] S. Axelsson, "The base-rate fallacy and the difficulty of intrusion detection," *ACM Trans. Inf. Syst. Secur.*, vol. 3, no. 3, pp. 186–205, 2000, doi: 10.1145/357830.357849.

[13] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," in *Proc. IEEE Symp. Computational Intelligence for Security and Defense Applications (CISDA)*, 2009, doi: 10.1109/CISDA.2009.5356528.

[14] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," in *Proc. 4th Int. Conf. Information Systems Security and Privacy (ICISSP)*, 2018, pp. 108–116, doi: 10.5220/0006639801080116.

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

[16] L. Liu, G. Engelen, T. Lynar, D. Essam, and W. Joosen, "Error prevalence in NIDS datasets: A case study on CIC-IDS-2017 and CSE-CIC-IDS-2018," in *Proc. IEEE Conf. Communications and Network Security (CNS)*, 2022, pp. 254–262, doi: 10.1109/CNS56114.2022.9947235.

[17] M. Catillo, A. Del Vecchio, A. Pecchia, and U. Villano, "Transferability of machine learning models learned from public intrusion detection data sets: The CICIDS2017 case study," *Softw. Qual. J.*, vol. 30, no. 4, pp. 955–981, 2022.

[18] R. Ross, V. Pillitteri, R. Graubart, D. Bodeau, and R. McQuaid, "Developing cyber-resilient systems: A systems security engineering approach," NIST, Gaithersburg, MD, USA, SP 800-160 Vol. 2 Rev. 1, 2021, doi: 10.6028/NIST.SP.800-160v2r1.

[19] M. Bruneau et al., "A framework to quantitatively assess and enhance the seismic resilience of communities," *Earthq. Spectra*, vol. 19, no. 4, pp. 733–752, 2003, doi: 10.1193/1.1623497.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)