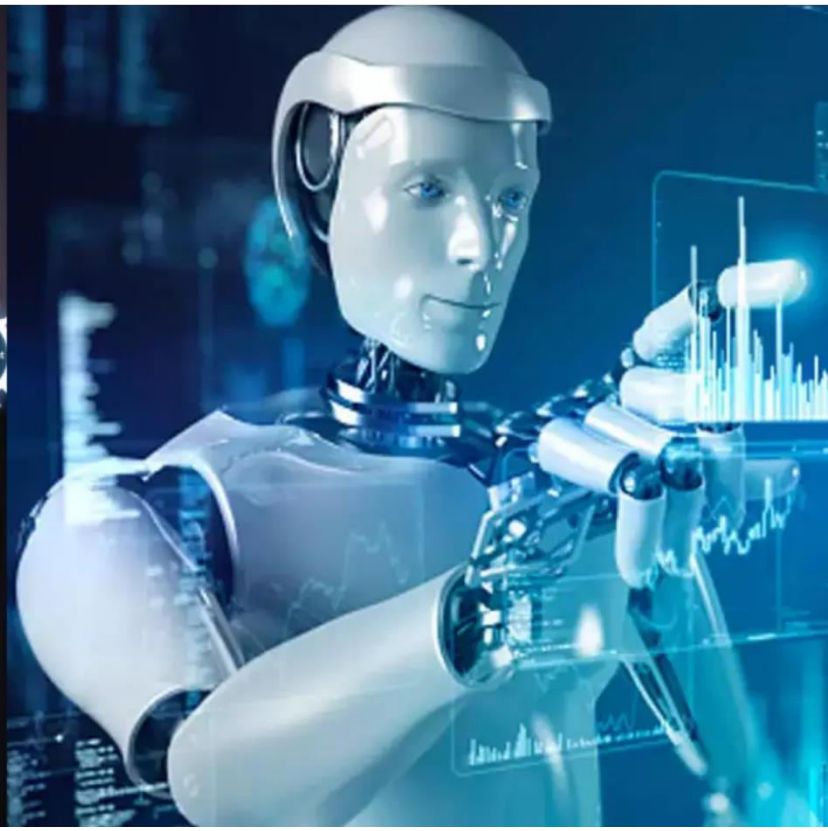
IJIRSET
International Journal of Innovative Research in Science | Engineering | Technology

ISSN(O): 2319-8753
ISSN(P): 2347-6710

International Journal of Innovative Research in Science Engineering and Technology (IJIRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)



Impact Factor: 9.935

Volume 15, Issue 10, October 2026

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 9.935

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 10, October 2026

|DOI: 10.15680/IJIRSET.2026.1510068|

A Risk-Aware Cross-Layer Machine Learning Framework for Web Application Threat Detection and Automated Response

Anjali Sahu[1], Dr. Ravindra Kumar Tiwari[2]

Research Scholar, Department of Computer Application, LNCT University Bhopal (M.P.), India[1]

Professor, Department of Computer Application, LNCT University Bhopal (M.P.) India[2]

ABSTRACT: Web applications are composed of a network, transport, application, identity, and infrastructure stack, while security monitoring is often distributed across intrusion-detection systems, web application firewalls, API gateways, authentication services, and application logs. Analyzing these sources in isolation can lead to missing linked events, inadequate alert context, and delayed or disproportional response actions. In this study, a risk-aware cross-layer security framework is proposed based on the Adaptive Cross-Layer Security Framework (ACSF) and Machine Learning Integrated Security Framework (MLISF) presented in the original thesis. The suggested architecture incorporates TCP/IP-aware telemetry, pre-processing, and feature extraction, complementing machine-learning detectors, contextual event correlation, and policy-governed mitigation. For labeled feature-based classification, we use Random Forest. When appropriate data is available, we use Long Short-Term Memory (LSTM) for temporal sequences. For anomaly scoring, we use Isolation Forest. The architecture also includes protections for automated action such as confidence thresholds, impact assessment, audit trail, analyst escalation and rollback. Overall detection accuracy: 97.1% False-positive rate: 2.1%. Mitigation success >96.9%. These are statistics supplied by the thesis, not results that we computed ourselves. There is a latency inconsistency in the thesis: the abstract states detection latency < 110 ms, whereas the conclusion states ~ 110-150 ms. As such, latency and mitigation claims must be validated against raw logs before release. This paper provides a more explicit layered architecture, a reproducible evaluation process, a plan for baseline comparison, and a measurement framework that isolates classification performance from operational effectiveness. It does not promise that proposed ablation tests or validation steps are already performed. The practical value of the framework needs to be proved with documented datasets, attack-trial denominators, model settings, repeated experiments, and evaluation in different scenarios.

KEYWORDS: Machine Learning, Intrusion Detection, Anomaly Detection, Automated Response, TCP/IP, Random Forest, LSTM, Web Application Security, Cross-Layer Security

I. INTRODUCTION

Web apps power banking, commerce, education, public services, healthcare, collaboration, and cloud-hosted information systems. Requests from their customer's flow through a chain of components: clients, reverse proxies, load balancers, network controls, application servers, identity services, APIs, databases, and container or cloud infrastructure. The Transmission Control Protocol/Internet Protocol (TCP/IP) suite provides for interoperable communication; however, the size of this environment presents many locations at which malicious conduct might occur. An attack can start with an unusual connection pattern, then progress through repeated authentication failures, and finally end with a malicious API call or an attempt to exploit an application vulnerability. Security monitoring with only one component may fail to reconstruct the complete sequence.

Threats to this environment include DoS/DDoS (denial-of-service and distributed denial-of-service), injection attacks, cross-site scripting, credential abuse, session hijacking, API abuse, scanning, and protocol anomalies. Common application security concerns, such as broken access control, injection, security misconfiguration, authentication failures, and logging or monitoring failures, are listed in the OWASP Top 10 [1]. Network-based intrusion detection and prevention systems provide much-needed visibility into traffic and known signatures, and web application firewalls monitor application layer requests. Security information and event management platforms consolidate logs. Each

component provides different information; the difficulty is to coordinate their outputs without presuming that any single source has total visibility [2, 3].

ML can support this coordination by finding patterns in structured data, by modeling temporal sequences, or by emphasizing observations that deviate from a learned reference distribution. But a high categorization score does not guarantee an operating system will be safe. Accuracy can be deceiving when there is a class imbalance. A model can miss infrequent attacks; an alert may not have enough information; and an automated block can inconvenience genuine users. Accordingly, detection, correlation, decision-making, mitigation, and verification should be considered discrete but linked functions.

This paper presents a risk-aware cross-layer framework based on the ACSF and MLISF principles from the provided thesis. The framework consists of five functional levels, including (i) TCP/IP-aware monitoring, (ii) preprocessing and feature extraction, (iii) machine-learning-based detection, (iv) cross-layer event correlation, and (v) adaptive mitigation and reaction. The design is not given as a freshly benchmarked system beyond what is in the thesis. Rather, it formalizes the architecture and details the evidence required to support reproducibility and efficacy.

1.1 Problem Statement

Independent security controls may issue related occurrences as independent notifications. Without solid temporal alignment, common identifiers, asset context, and correlation criteria, analysts could find it impossible to differentiate a concerted attack from independent anomalies. Additionally, detector outputs differ from proven incidents: a model may be ambiguous, a rule may be triggered by benign traffic, and an alert may not require an urgent disruptive response. The research topic is thus how to integrate heterogeneous network and application observations with ML-based detection and controlled mitigation while keeping performance, uncertainty, and operational risk measurable.

1.2 Aim and objectives

The purpose is to define and test an AI-based cross-layer security framework for TCP/IP web applications. The goals are to (1) define a telemetry model that spans network and application sources; (2) combine supervised classification, temporal analysis, and anomaly detection; (3) correlate events by time, session, asset, and identity context; (4) map risk estimates to bounded response actions; (5) define a reproducible evaluation protocol for detection, latency, mitigation, and resource use; and (6) identify limitations that must be addressed before deployment.

1.3 Research questions

Empirical validation is guided by four research questions. RQ1 asks, "Is the integrated framework better than a selection of independent baselines on held-out traffic? " RQ2 is whether cross-layer correlation minimizes false alarms or provides better incident-level precision when compared to independent alerting. RQ3 asks about the end-to-end latency introduced by ingestion, inference, correlation, and response. RQ4: How does monitoring and automatic mitigation impact throughput, CPU, memory, and success of legitimate requests? These are research propositions, not results. The publication does not treat these questions as having been answered without the appropriate comparison measures.

1.4 Contributions and Scope

The contribution is the definition of an integrated workflow where model outputs are interpreted within the framework of cross-layer and policy restrictions. It also distinguishes stated thesis metrics from recommended assessment processes, defines acceptable denominators, and proposes ablation tests to isolate contributions of detection, correlation, and response automation. Scope is the protective monitoring of approved web application environments. The architecture is not a replacement for secure code, patch management, identity controls, or incident-response governance. Automated action should be reserved for approved defensive controls, and high-impact decisions should be subject to review and reversal.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 9.935

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 10, October 2026

|DOI: 10.15680/IJIRSET.2026.1510068|

Table 1: Components of the framework and implementation details to be documented

Component	Role in framework	Source-described technology / evidence to report
Telemetry collection	Collect network, WAF, API, identity and application events	Suricata, ModSecurity, application logs; list sensor configuration and coverage
Container/testbed layer	Host services and controlled test scenarios	Docker and Kubernetes; specify versions, topology and resource limits
Event management	Normalize, search and correlate events	Elastic Stack; record ingestion pipeline, schemas and retention
ML detection	Classify labelled events, model sequences, flag anomalies	Random Forest, LSTM, Isolation Forest; publish hyperparameters and model versions
Mitigation	Apply policy-governed defensive actions	Document rule types, thresholds, action logs, rollback and safety checks

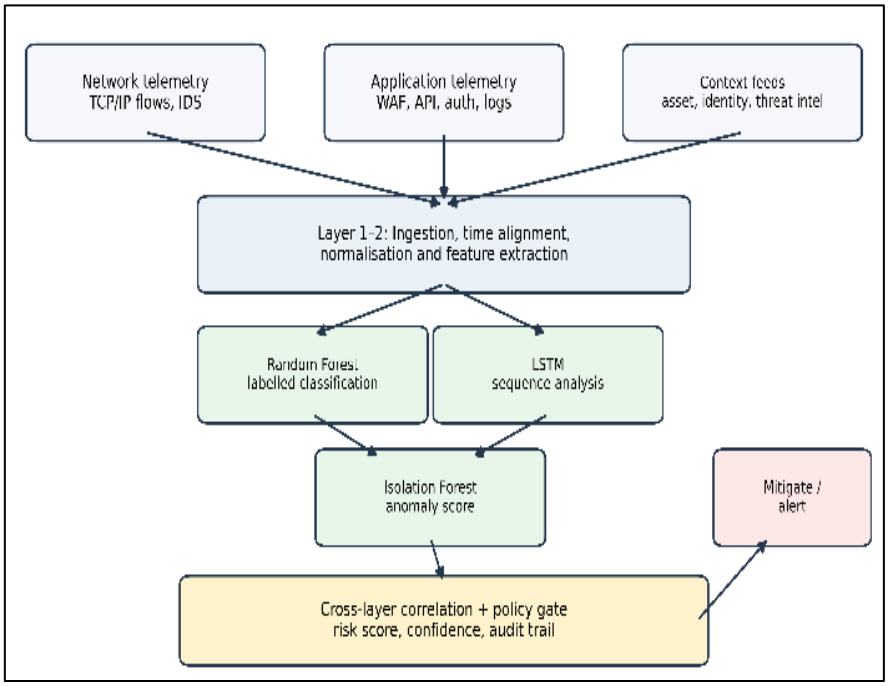


Figure.1.Proposed AI-driven cross-layer security architecture

II. PAST WORK AND RESEARCH GAP

2.1 Network and Application Security Surveillance

Early examination of security in TCP/IP showed the inadequacy of the assumptions of the protocols and the difficulties of securing a layered suite as a whole [4]. Thus, modern defensive operations incorporate several controls: packet and flow inspection, host monitoring, application-layer filtering, identity and access control, and centralized event management. The NIST guideline on intrusion detection and prevention outlines network-based, host-based, wireless, and network-behavior techniques and emphasizes the necessity of configuration, monitoring, and maintenance rather than treating an IDS as a standalone product [2]. The web application security guidance also focuses on prevention, secure design, validation, access control, and logging [1, 5].

The evidence is different for the controls. A network sensor can see connection length, packet rates, TCP flags, and endpoints, but it may not detect application semantics in encrypted sessions. An application firewall can detect patterns of suspiciousness in requests but may not know whether a source is generating a larger flood of connections. Credential abuse is evident in authentication logs but requires session and network context to differentiate an attack from a user's

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 9.935

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 10, October 2026

|DOI: 10.15680/IJIRSET.2026.1510068|

repeated mistake. If there are identifiers and the time alignment is reliable, cross-layer correlation can link these signals.

2.2 Intrusion Detection Based on Machine Learning

Supervised learning maps labeled examples to classes. Random Forest is a combination of decision tree models and may capture non-linear relationships among structured flow or event features [6]. This approach is useful when there are representative labels, careful feature creation, proper weighting of classes, and testing data reserved for evaluation. Sequential models such as LSTM may model temporal dependencies and are applicable when the input is an ordered sequence rather than an unordered row of flow statistics [7]. Isolation Forest uses a recursive partitioning strategy to identify anomalous observations and is applicable in settings where labels are missing or inadequate [8]. The approaches make different learning assumptions, and one should not mix their scores naively without calibration or an explicit choice procedure.

Research on intrusion detection has employed benchmark datasets such as KDD Cup 1999 [9], NSL-KDD [10], UNSW-NB15 [11], and CICIDS2017 [12]. These resources are beneficial for controlled comparison but differ in capture settings, feature sets, attack distributions, and age. Benchmark results are not the same as performance on a real enterprise application. Dataset provenance, train/test separation, feature leakage, duplicate records, temporal drift, and attack-family imbalance can significantly affect the results reported. A thorough study should specify each dataset and version, publish the split technique, present class-wise metrics, and preferably test transfer between datasets or settings.

2.3 Security analytics and correlation of events

Centralized logging and security analytics allow analysts to search across events from disparate products. Correlation can be performed on timestamps, source and destination addresses, session IDs, user accounts, container IDs, request IDs, and asset criticality. But correlation is risky in itself: clock skew can lead to misordered events, NAT and proxies can hide the origin, identifiers might be missing, and high-volume traffic can produce noisy connections. The system should retain source-event references and offer a justification for each correlation choice so that an analyst can trace a high-level incident back to evidence.

Zero-trust architecture focuses on explicit access decisions and the protection of resources, rather than trusting based on network location [13]. This idea complements cross-layer monitoring, in that the correlation engine can provide context, but access permission should still be driven by identity and resource policy. Threat-intelligence feeds can provide reputation or known-indicator context, although indicators can be old, shared by benign and malevolent people, or incomplete. Therefore, reputation ought to be part of a risk assessment, not an automatic finding of guilt.

2.4 Automated response and operating risk

Automation can minimize time to containment, but misblock might produce its own availability incidents. Challenge methods, interim rules, session termination and host isolation, and rate limitation differ in their severity and reversibility. The decision process should take into account confidence, agreement in the evidence, asset criticality, possible impact, and the cost of a false positive. A high confidence, corroborated incident may warrant a narrowly scoped temporary block, whereas a low confidence anomaly may merit logging and analyst evaluation. All activities should be time-bounded if possible, tracked, and reversible.

2.5 Gap in research

The original thesis highlights the combination of network monitoring, ML-based detection, cross-layer correlation, and adaptive response as the main impetus for ACSF and MLISF. The gap is not only a lack of ML classifiers. The question is if the combination of heterogeneous telemetry and a regulated response policy produces quantifiable improvement over component-level techniques, without excessive latency or service disruption. Existing detector scores alone are not enough to address the question. The suggested evaluation thus separates (a) event-level classification, (b) incident-level correlation, (c) successful mitigation, and (d) operational overhead. To prove uniqueness or superiority, we must compare against stated baselines and run ablation tests, not only show architecture diagrams.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 9.935

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 10, October 2026

|DOI: 10.15680/IJIRSET.2026.1510068|

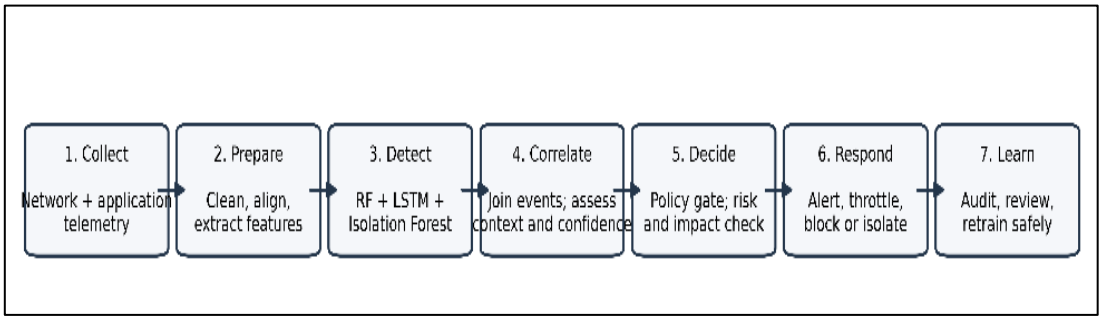


Figure.2. End-to-end detection and response workflow

III. PROPOSED FRAMEWORK AND METHODOLOGY

3.1 Design concepts

The proposed architecture is based on five concepts. First, telemetry is acquired only from authorized assets and according to defined data processing guidelines. Second, the preprocessing must be fitted to training data exclusively to prevent information from leaking. Third, ML detectors are complementing components, not interchangeable algorithms. Fourth, correlation and policy judgments must be traceable to source events. Fifth, mitigation must be appropriate, reversible, and evaluated in terms of security benefits and effects on legitimate service.
The architecture is shown in Figure 1. The end-to-end workflow is shown in Figure 2. Figure 3 shows the different evidence sources and their relationships. The architecture is taken from the source thesis; the diagrams are conceptual representations and not an implication that every integration has been experimentally proven.

3.2 Layer 1: Monitoring aware of TCP/IP

The first layer gathers traffic and security observations from network sensors, web application firewalls, API gateways, identity systems, application logs, and host or container telemetry. Candidate flow features include protocol, duration, packet and byte counts, TCP flags, connection state, and rates. Application characteristics may include HTTP method, normalized route, status code, request size, response size, authentication result, rate per account/token, and selectable validation outcomes. This infrastructure context may have workload identification, container or service name, resource usage, and asset criticality.

Collect only so much personal information as is necessary for the purpose. Payload capture can expose credentials, tokens, or sensitive user content. Metadata and redacted structured fields should be used wherever possible. Logs should be role-limited , retained for as long as needed , and protected against tampering . The framework should also be able to deal with encrypted traffic, proxies, load balancers and network address translation that might restrict or alter the semantics of observed addresses and connection attributes.

3.3 Layer 2: pre-processing and feature extraction

The preparation pipeline should do schema validation, delete exact duplicates when justified, impute missing data, normalize numeric features when justified by the model, encode categorical variables, and align event timestamps. Missingness should not be assumed to be innocuous behavior by default but may mean sensor failure or inadequate visibility. Feature extraction should exclude identifiers that expose the label or that are duplicates of the goal outcome. For example, if all malicious samples come from a specific attack host, the source address may allow memorization instead of learning behavior that generalizes.

Split training, validation, and test sets before fitting transforms or picking features. If records are time dependent, a time-based split is usually more informative than a random row split. If the same sessions or near-duplicate flows can occur across partitions, consider group-based isolation. Any resampling or synthetic minority oversampling should only be done within the training folds. The final report must specify the use of any missing-value treatment, scaling, encoding, feature selection, class balance, and random seeds. 3.4 Layer 3: complementary ML detectors

The architecture employs three candidate detectors.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 9.935

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 10, October 2026

|DOI: 10.15680/IJIRSET.2026.1510068|

Random Forest. A supervised Random Forest transfers a feature vector x to a class prediction and, when properly calibrated, to a probability estimate. It is suitable for organized traffic and event features and can describe non-linear feature interactions. Relevant parameters are number of trees, maximum depth, minimum samples per leaf, class weights, feature-subsampling approach, and random seed. These configurations are to be reported from the actual experiment, not described by the source material available for this revision.

Long Short-Term Memory (LSTM) network. An LSTM is appropriate when events are organized into meaningful sequences, such as repeated authentication attempts, or fluctuations in request rates over a particular time frame. The manner in which a sequence is constructed must define the window length, stride, ordering key, padding or truncation, and target alignment. Randomly mixing individual rows prior to sequence building would remove temporal information. Document the model complexity, learning rate, optimizer, batch size, epochs, early-stopping rule, and hardware. You cannot argue that LSTM adds value without comparing its performance with a non-sequential baseline.
Isolation Forest. Isolation Forest assigns an anomaly score to a point based on how easy it is to isolate by randomly splitting the data. It may be useful for identifying abnormal behavior outside the labeled training data, although anomaly is not necessarily malicious. The choice of threshold should be driven by validation data and a documented false alarm budget. Instead of a direct translation of each outlier to a block action, the system should keep the score and the appropriate feature context.

A practical ensemble may combine the outputs of the detectors with a calibrated weighted score or rule-based policy. If model outputs are not calibrated, a naive average of scores can be misleading. Thus, the research proposes to log the score, calibration process, confidence threshold, and contribution to the final judgment for each detector. The precise fusion strategy has to be chosen and tested in experiments.

3.5 Layer 4: cross-layer linkage of events

The correlation engine clusters linked observations into candidate occurrences. Useful keys are timestamps with tolerance, request or session IDs, source and destination tuples, account or token identities, service identification, and asset relationships. Correlation can be rule-based, window-based, or graph-oriented. A simple implementation would trigger an incident when several independent signals happen within a given time frame. For example, an odd connection burst, a series of failed logins, and a string of denied API requests can occur in succession. This is a pattern example, not a claimed attack outcome.

The engine must be able to differentiate between direct matches and inferred relationships. Each incident must maintain the original event IDs, source system, event time, ingestion time, correlation rule, missing-context indicators, and confidence. Document clock synchronization, delayed logs, proxy rewriting, NAT, and partial sensor outages. A correlation finding should not obliterate disagreement among detectors; contradictory evidence is operationally significant and may support a lower confidence action.

3.6 Layer 5: Risk Sensitive Mitigation

The response layer maps the incident context to a constrained action. Candidate answers include alerting, additional logging, rate limitation, CAPTCHA or step-up authentication, temporary IP or token restrictions, session termination, WAF rule activation, and isolation of a compromised workload. The least disruptive measure that could reduce risk should be preferred. Response decisions should be based on the confidence of the evidence, the number of independent sources, the criticality of the target asset, the likely harm if the alarm is false, and the projected duration of the action.

A risk-tiered decision route is shown in Figure 5. Low-confidence events are logged and escalated. Medium-confidence events may prompt reversible rate limits or additional verification. If multiple sources report events with high confidence, interim containment may be taken under an approved policy. Emergency activities still need to be reported and audited. If the automation is causing too many genuine requests to fail or the telemetry is unreliable, a circuit breaker should disable it. Before deployment, test the rollback procedures and create an allow list for important management endpoints.

3.7 Implementation environment

The source thesis describes the use of Docker, Kubernetes, Suricata, ModSecurity, and Elastic Stack. These solutions provide a containerized testbed, network intrusion detection, web request filtering, and centralized log management. The paper also does not infer the exact hardware specifications, versions, dataset sizes, or model hyperparameters as

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 9.935

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 10, October 2026

|DOI: 10.15680/IJIRSET.2026.1510068|

these are not specified in the document provided. Table 1 distinguishes between source-described components and the implementation details that require documentation. Reproducibility needs an environment manifest, configuration export, dependency version, data-preparation script, and versioned experiment log.

3.8 Experimental setup and baselines

Use a held-out test set and report findings by attack class and benign traffic. Four baselines are proposed: (B1) a selected traditional IDS/WAF setup; (B2) a standalone supervised ML detector without cross-layer correlation; (B3) the detector with correlation enabled but response automation disabled; and (B4) the entire framework with policy-driven mitigation. The architecture allows us to separate if the gains are a function of the classifier, the correlation, or the response process. Where technically feasible, baselines should be set up equally and run on the same test circumstances.

Where the testbed supports repeated runs, at least three separate repeats are advised, although the actual number should be decided according to the resources available and reported honestly. Report the mean and dispersion, confidence intervals when appropriate, and paired comparisons when the same situations are evaluated under different conditions. If the data set only has one fixed split, assertions concerning variability should be restrained. In time series or repeated attack studies, the independent unit may be the session, run, or attack trial, rather than each packet. A denominator must match the design.

3.9 Metrics and definitions

In binary classification, accuracy is $(TP+TN)/(TP+TN+FP+FN)$, precision is $TP/(TP+FP)$, recall is $TP/(TP+FN)$, and F1 is the harmonic mean of precision and recall. The false positive rate is $FP/(FP+TN)$, with the caveat of the definition of negative observations mentioned above. Macro-F1 is the average of the F1 values of each class equally weighted. This metric is important when the attack classes are uneven. The area under the precision-recall curve can be instructive when positives are infrequent. Aggregate metrics should be shown together with the confusion matrix and per-class data.

Latency necessitates both a beginning and an end. Model inference latency assesses the prediction stage; detection delay may start when relevant telemetry becomes observable; and end-to-end response latency ends when the defensive control is implemented and verified. Mitigation success should be calculated as the number of successful eligible attack trials divided by the total number of eligible attack trials under a predeclared criterion. But an effective answer must be operationally proven, not assumed from the promulgation of a rule. These evaluation dimensions are summarized in Figure 4. ‘

3.10 Statistical and reproducibility safeguards

Data leakage is a significant danger to authenticity. Don’t use test labels to transform features, select features, adjust thresholds, or search for hyperparameters. Lastly, one needs to consider duplication of dataset and temporal overlap. Class counts and split boundaries should be included in the paper. If a statistical test is applied, report the test, unit of analysis, assumptions, effect size, and exact p-value. A p-value alone does not indicate the practical significance of the results. Statistical validation is reported as $p < 0.001$ in the thesis, but the publication does not provide enough material to support that claim independently. It should not be preserved as a verified result unless the original analytic outputs support it.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 9.935

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 10, October 2026

|DOI: 10.15680/IJIRSET.2026.1510068|

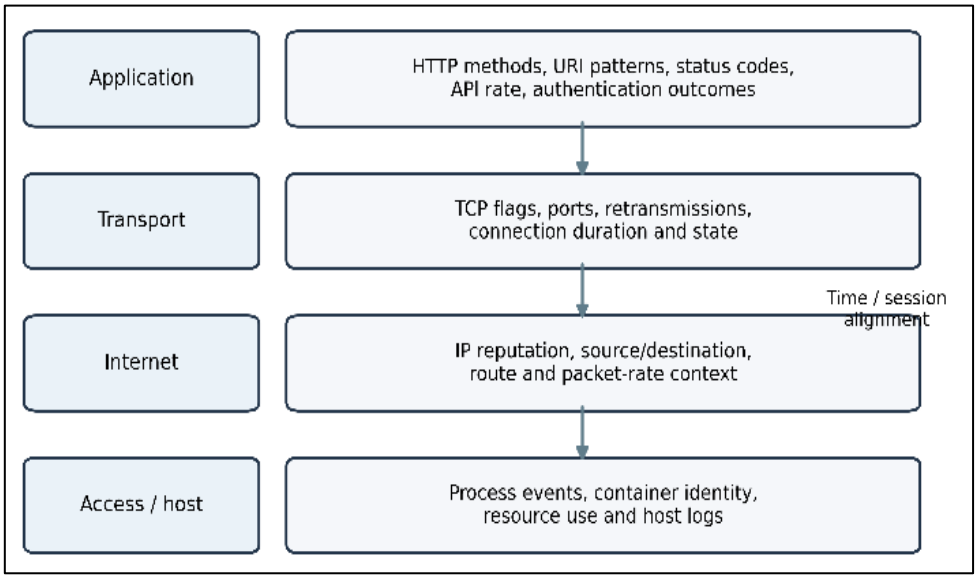


Figure.3.Cross-layer telemetry and contextual correlation

Table.2. Candidate telemetry and derived feature groups

Source layer	Candidate features	Key caveat
Network / transport	Protocol, duration, packet and byte rates, TCP flags, connection state	Encryption, NAT and proxies may limit endpoint attribution
Application / API	Method, normalized route, status code, request size, response size, request rate	Avoid retaining secrets or unnecessary payload content
Identity / session	Authentication result, failed-login rate, token/session identifier, privilege change	Pseudonymize identifiers and restrict access
Host / container	Workload identity, CPU/memory, process events, service identity	Sensor coverage and clock skew affect completeness
Context	Asset criticality, reputation, change window, known maintenance	Threat-intelligence indicators may be stale or shared

Table.3. Machine-learning components and validation requirements

Model	Intended role	Input assumptions	Validation requirement
Random Forest	Supervised classification	Structured labelled examples	Per-class metrics, feature leakage checks, calibration
LSTM	Temporal sequence modelling	Ordered events with defined windows	Compare with non-sequential baseline; report window and training settings
Isolation Forest	Anomaly scoring	Reference distribution of normal or mixed observations	Threshold selection on validation data; measure false-alert burden
Fusion / policy	Combine evidence for incident decision	Comparable or calibrated scores plus event context	Ablation tests; document weighting, thresholds and confidence handling

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 9.935

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 10, October 2026

|DOI: 10.15680/IJIRSET.2026.1510068|

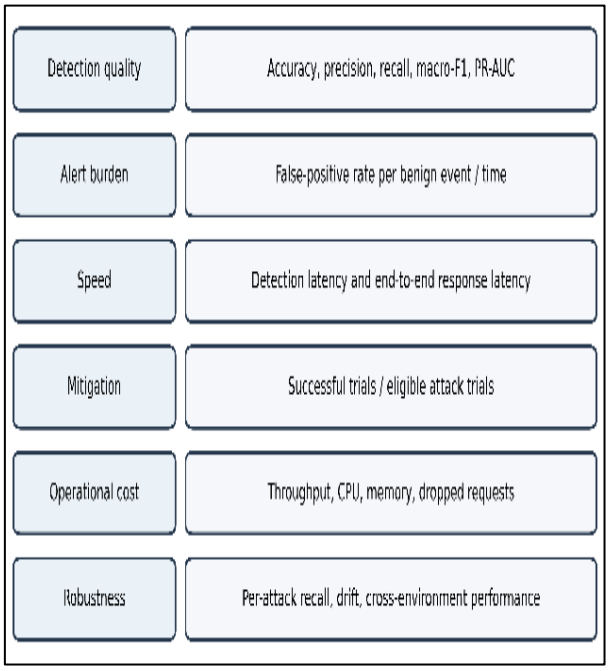


Figure.4. Evaluation dimensions for a reproducible security study.

Table.4. Performance figures reported in the source thesis (not independently recalculated)

Indicator	Value in source thesis	Interpretation / verification needed
Overall detection accuracy	97.1%	Recalculate from held-out predictions and confusion matrix
False-positive rate	2.1%	Confirm negative-class definition and denominator
Mitigation success	Greater than 96.9%	Report successful trials / eligible trials and success criterion
Average detection latency	Abstract: under 110 ms; conclusion: approximately 110–150 ms	Inconsistent across thesis sections; reconcile raw logs and stage definitions

IV. RESULTS AND DISCUSSION

4.1 Performance reported in thesis

The thesis at hand claims an overall accuracy of 97.1% in detecting threats, a false-positive rate of 2.1%, and a mitigation success rate of over 96.9%. The abstract says < 110 ms average detection latency, while the conclusion says around 110-150 ms. The values given in Table 4 are those published in the source, and no recalculation is done. The publication does not contain raw prediction files, confusion matrices, counts of attack trials, traffic captures, or response logs that would allow for independent recalculation.

The accuracy result suggests a high aggregate classification rate under the thesis's experimental settings, but it does not by itself demonstrate generalization. If benign traffic is dominant in the test set, a high accuracy can coexist with a poor recall for less frequent attacks. The 2.1% false positive rate may be informative of alert burden; however, how we define a negative example and how a denominator was created influences the conclusion. Also, a per-event false positive rate doesn't simply translate into the number of instances an analyst might analyze every hour or day. Therefore, operational reporting should contain classification metrics and alert volume at the rate of target traffic.

The 96.9% mitigation success indicated above is a separate mitigation success claim, not a detection accuracy claim. A definition of a successful trial, the number of qualifying trials, the assault scenario, the response performed, and proof

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 9.935

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 10, October 2026

|DOI: 10.15680/IJIRSET.2026.1510068|

that the attack was materially curtailed are required. If the determinant is just successful or full trials, the proportion may exaggerate effectiveness. The publication should publish counts, e.g., successful trials / total eligible trials, along with the percentage. Here no count is implied, as the provided document does not offer it.

4.2 Latency mismatch

Specifically, latency should be considered. Detection latency in the source thesis abstract is under 110 ms, whereas in its conclusion it is around 110-150 ms. These assertions are unequal and should not be combined in a single estimate. Please cross-check the original tables, experiment logs and definitions to see whether the values correspond to different stages (e.g. model inference vs. end-to-end detection) or an editorial inconsistency. The manuscript has to specifically label latency as unresolved until resolved. It should not advertise a verified <110ms answer.

Latency should be monitored at multiple points: availability of telemetry, completion of preprocessing, output of detectors, completion of correlation, command of response, and validated mitigation. Only reporting the model inference time can underestimate the operational delay. Median, 95th percentile, and 99th percentile are useful because security events often happen under load, when tail latency matters. Use synchronized clocks or a single monotonic source of measurement for timing. Document concurrent traffic and system load for repeated runs.

4.3 What architecture can add—and what it cannot guarantee

Cross-layer correlation enhances context by correlating a network abnormality with authentication or application events. This approach may also help to prevent redundant warnings by grouping relevant observations into an incident. Correlation can generate spurious links when the time span is too wide, identities are repeated, or infrastructure is shared between unrelated users. It may also cause processing slowness. So RQ2 asks for a direct comparison of precision, recall, false-alert volume, and latency at alert level and incident level with correlation on and off.

Random Forest, LSTM, and Isolation Forests complement modeling assumptions. Combining three algorithms does not ensure better performance. Random Forest may be powerful on structured characteristics. LSTM may be beneficial where sequence order involves signal isolation. Forest may flag previously unlabeled deviations. But the anomaly detection may produce false positives on valid new workloads, while the sequence model may overfit a specific traffic pattern. We should compare in ablation research each detector alone, the fusion approach used, and the whole cross-layer pipeline. Without these tests, the study should present the complementarity as a design logic, not a performance advantage.

Automation can decrease the time to contain a threat, but you need to weigh the advantages against the potential for false blocks and their impact on availability. A temporary block can work well against a malicious source but be disastrous when a shared proxy or carrier-grade NAT address is utilized by numerous legitimate users. Safer initial answers may be to rate limitations or step-up verification. The risk gate and rollback mechanism of the framework is therefore not an optional add-on but a necessary design control. Evaluation should include benign-request success rate, false-block duration, recovery time, and number of actions that need to be manually reversed.

4.4 Baseline comparison and ablation schedule

A suggested comparison matrix is given in Table 5. It's a process, not a table of made-up performance data. Run the same test cases on each setup. B1 provides the operational baseline, B2 assesses the utility of ML classification, B3 separates the impact of cross-layer correlation, and B4 measures the whole workflow, including response automation. Report the following metrics: accuracy, macro-F1, recall per class, false-positive rate, detection latency, response latency, success of mitigation, throughput, CPU usage, and memory usage. If the inputs are not identical, the differences in visibility should be reported as a limitation.

The comparison must show ablations for the three types of detectors. You can remove one detector at a time to see if it provides any distinct value or just raises computational expense. An ablation of correlation should be between incident-level and event-level outcomes. A response ablation should differentiate between detection-only mode and mitigation-enabled mode (including service disruption). The framework should be evaluated under varying traffic rates and benign workload shifts to determine thresholds at which the performance declines.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 9.935

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 10, October 2026

|DOI: 10.15680/IJIRSET.2026.1510068|

4.5 Validity threats

Construct validity. Operational definitions should be given for the terms “attack detected," "incident," “false positive," and “mitigation success." A model prediction does not mean there is a security problem. Containment is not a reaction order.

Internal validity. Performance could be inflated by data leakage, repeated flows, attack-host IDs, temporal overlap, incorrect class sampling, or threshold adjustment on the test set. These problems should be audited in the experimental procedure. If a step was not documented in the original thesis, the manuscript should not infer that it was done.

External validity. The thesis describes a controlled test bed. Production traffic can have more heterogeneous clients, encrypted connections, legacy programs, cloud services, shifting workloads, and infrequent events. Only after testing performance across contexts and datasets may one claim generality.

Statistical Conclusion Validity The assertion with $p < 0.001$ requires clarification regarding the test method, sample size, independence assumptions, effect magnitude, and the underlying data. If the experimental unit is the run, repeated packets from a single attack run should not be viewed as independent experiments. Confidence intervals and paired analysis are preferred if the design permits them.

Real enterprise. Long-term maintenance, model drift, retraining, personnel effort, hardware cost, and incident-response governance can significantly impact deployment. These factors are noted as limitations in the source thesis and should be taken into account in any field pilot.

4.6 Implications for implementation and governance

A sensible deployment should start in passive or alert-only mode. During this phase, the team can compare alerts with analyst findings, measure false positives, and modify correlation rules without impacting user traffic. A subsequent controlled phase may allow for reversible rate limits or further authentication for certain classes of attacks. High-impact blocking should only be enabled after a documented acceptance test and a process for operational rollback. The deployment checklist needs to include access permissions, data minimization, model versioning, configuration backup, audit integrity, detector health monitoring and emergency disablement. Traceability is especially important where false positives impact real users.

Figure.5. Risk-tiered mitigation with human oversight

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 9.935

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 10, October 2026

|DOI: 10.15680/IJIRSET.2026.1510068|

Table.5. Proposed baseline and ablation comparison

Configuration	Correlation	Automated mitigation	Purpose
B1: conventional IDS/WAF	Product-native / none	Existing baseline policy	Establish operational baseline
B2: standalone ML detector	Off	Off	Measure incremental value of ML classification
B3: ML + cross-layer correlation	On	Off	Isolate effect on incident precision and alert burden
B4: full integrated framework	On	Policy-gated	Measure end-to-end detection and response outcomes
Ablations	Remove one detector at a time	Held constant where feasible	Estimate each detector's incremental contribution

Table.6. Recommended metrics and operational definitions

Metric	Definition / reporting guidance
Accuracy	(TP + TN) / (TP + TN + FP + FN); report with class distribution
Precision / recall / F1	Report per class and macro average; include confusion matrix
False-positive rate	FP / (FP + TN); state what constitutes a negative event
Detection latency	Define start as telemetry availability and end as detector/correlation output; report percentiles
End-to-end response latency	Time from observable event to verified mitigation, not merely command issuance
Mitigation success	Successful eligible attack trials / all eligible trials under predeclared criteria
Operational overhead	Throughput, CPU, memory, dropped requests and benign-request success
Robustness	Per-attack performance, drift sensitivity, and cross-dataset / cross-environment results

V. CONCLUSION AND FUTURE WORK

In this paper, the principles of ACSF and MLISF have been adapted into a risk-aware cross-layer security framework for TCP/IP-based online applications. The design ties together heterogeneous telemetry, preprocessing, Random Forest classification, LSTM sequence analysis, Isolation Forest anomaly scoring, event correlation, and policy-driven mitigation. Its basic thesis is that good security is more than a high classifier score. Evidence needs to be contextualized, reaction measures need to be proportionate, and both security outcomes and operational expenses need to be monitored. The source thesis reported an overall detection accuracy of 97.1%, a false positive rate of 2.1%, and mitigation success above 96.9%. Values provided in the thesis require confirmation by reference to the original experimental records. The discrepancy between the sub-110 ms latency in the abstract and the ~110-150 ms in the conclusion needs to be reconciled for a publication ready claim. The paper does not report new experimental measurements, nor does it claim that the proposed baseline comparisons have been done before.

The next stage should provide the actual names and versions of the datasets, preprocessing pipeline, class distributions, train-validation-test splits, hyperparameters, random seeds, hardware and software versions, baseline setups, and raw or adequately anonymized outcome tables. Evaluation metrics should contain class-specific metrics, confidence intervals where justified, latency percentiles, attack-trial denominators, throughput impact, and benign-request success. The ablation tests are to identify the contribution of each detector, cross-layer correlation, and automated response. Testing in different datasets or operational settings would give more proof of generalizability.

Explainable ML, privacy-preserving collaborative learning, drift-aware re-training, edge-based security orchestration, and trusted threat intelligence sharing can be conducted in future research. Any adaptation process must have defenses against poisoning, model evasion, and unauthorized policy modification. A staggered deployment with human supervision, the ability to roll back, and ongoing monitoring is recommended before enabling the autonomous high-

impact mitigation. Overall, the paradigm provides a consistent research path, but the reproducibility and independent empirical validation are still required to verify the practical effectiveness.

REFERENCES

1. Bellovin, S. M. (1989). Security problems in the TCP/IP protocol suite. ACM SIGCOMM Computer Communication Review, 19(2), 32–48. https://doi.org/10.1145/66093.66094
2. Scarfone, K., & Mell, P. (2007). Guide to intrusion detection and prevention systems (IDPS) (NIST Special Publication 800-94). National Institute of Standards and Technology. https://doi.org/10.6028/NIST.SP.800-94
3. OWASP Foundation. (2021). OWASP Top 10: The ten most critical web application security risks. https://owasp.org/Top10/
4. Rose, S., Borchert, O., Mitchell, S., & Connelly, S. (2020). Zero trust architecture (NIST Special Publication 800-207). National Institute of Standards and Technology. https://doi.org/10.6028/NIST.SP.800-207
5. Breiman, L. (2001). Random forests. Machine Learning, 45, 5–32. https://doi.org/10.1023/A:1010933404324
6. Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural Computation, 9(8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735
7. Liu, F. T., Ting, K. M., & Zhou, Z.-H. (2008). Isolation forest. In Proceedings of the 2008 IEEE International Conference on Data Mining (pp. 413–422). IEEE. https://doi.org/10.1109/ICDM.2008.17
8. Moustafa, N., & Slay, J. (2015). UNSW-NB15: A comprehensive data set for network intrusion detection systems. In 2015 Military Communications and Information Systems Conference (MilCIS) (pp. 1–6). IEEE. https://doi.org/10.1109/MilCIS.2015.7348942
9. Sharafaldin, I., Lashkari, A. H., & Ghorbani, A. A. (2018). Toward generating a new intrusion detection dataset and intrusion traffic characterization. In Proceedings of the 4th International Conference on Information Systems Security and Privacy (ICISSP) (pp. 108–116). SCITEPRESS. https://doi.org/10.5220/0006639801080116
10. Tavallaee, M., Bagheri, E., Lu, W., & Ghorbani, A. A. (2009). A detailed analysis of the KDD CUP 99 data set. In Proceedings of the 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications (pp. 1–6). IEEE. https://doi.org/10.1109/CISDA.2009.5356528
11. Canadian Institute for Cybersecurity. (2017). CICIDS2017 dataset. University of New Brunswick. https://www.unb.ca/cic/datasets/ids-2017.html
12. Sommer, R., & Paxson, V. (2010). Outside the closed world: On using machine learning for network intrusion detection. In 2010 IEEE Symposium on Security and Privacy (pp. 305–316). IEEE. https://doi.org/10.1109/SP.2010.25
13. Axelsson, S. (2000). The base-rate fallacy and the difficulty of intrusion detection. ACM Transactions on Information and System Security, 3(3), 186–205. https://doi.org/10.1145/357830.357849
14. Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection: A survey. ACM Computing Surveys, 41(3), Article 15. https://doi.org/10.1145/1541880.1541882
15. Buczak, A. L., & Guven, E. (2016). A survey of data mining and machine learning methods for cyber security intrusion detection. IEEE Communications Surveys & Tutorials, 18(2), 1153–1176. https://doi.org/10.1109/COMST.2015.2494502
16. Khraisat, A., Gondal, I., Vamplew, P., & Kamruzzaman, J. (2019). Survey of intrusion detection systems: Techniques, datasets and challenges. Cybersecurity, 2, Article 20. https://doi.org/10.1186/s42400-019-0038-7
17. García-Teodoro, P., Díaz-Verdejo, J., Maciá-Fernández, G., & Vázquez, E. (2009). Anomaly-based network intrusion detection: Techniques, systems and challenges. Computers & Security, 28(1–2), 18–28. https://doi.org/10.1016/j.cose.2008.08.003
18. Patcha, A., & Park, J.-M. (2007). An overview of anomaly detection techniques: Existing solutions and latest technological trends. Computer Networks, 51(12), 3448–3470. https://doi.org/10.1016/j.comnet.2007.02.001
19. Dua, S., & Du, X. (2011). Data mining and machine learning in cybersecurity. CRC Press.
20. Mitchell, R., & Chen, R. (2014). A survey of intrusion detection techniques for cyber-physical systems. ACM Computing Surveys, 46(4), Article 55. https://doi.org/10.1145/2542049
21. Liao, H.-J., Richard Lin, C.-H., Lin, Y.-C., & Tung, K.-Y. (2013). Intrusion detection system: A comprehensive review. Journal of Network and Computer Applications, 36(1), 16–24. https://doi.org/10.1016/j.jnca.2012.09.004
22. Cichonski, P., Millar, T., Grance, T., & Scarfone, K. (2012). Computer security incident handling guide (NIST Special Publication 800-61 Rev. 2). National Institute of Standards and Technology. https://doi.org/10.6028/NIST.SP.800-61r2
23. Joint Task Force. (2020). Security and privacy controls for information systems and organizations (NIST Special Publication 800-53 Rev. 5). National Institute of Standards and Technology. https://doi.org/10.6028/NIST.SP.800-53r5

Volume 15, Issue 10, October 2026

|DOI: 10.15680/IJIRSET.2026.1510068|

24. Souppaya, M., Scarfone, K., & Dodson, M. (2022). Guide to enterprise patch management planning: Preventive maintenance for an organization's technology (NIST Special Publication 800-40 Rev. 4). National Institute of Standards and Technology. https://doi.org/10.6028/NIST.SP.800-40r4
25. MITRE. (n.d.). MITRE ATT&CK: Enterprise matrix. Retrieved October 22, 2026, from https://attack.mitre.org/
26. OWASP Foundation. (2023). OWASP API Security Top 10—2023. https://owasp.org/API-Security/
27. García, S., Grill, M., Stiborek, J., & Zunino, A. (2014). An empirical comparison of botnet detection methods. Computers & Security, 45, 100–123. https://doi.org/10.1016/j.cose.2014.05.011
28. Ring, M., Wunderlich, S., Scheuring, D., Landes, D., & Hotho, A. (2019). A survey of network-based intrusion detection data sets. Computers & Security, 86, 147–167. https://doi.org/10.1016/j.cose.2019.06.005
29. McHugh, J. (2000). Testing intrusion detection systems: A critique of the 1998 and 1999 DARPA intrusion detection system evaluations as performed by Lincoln Laboratory. ACM Transactions on Information and System Security, 3(4), 262–294. https://doi.org/10.1145/382912.382923
30. Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. The Annals of Statistics, 29(5), 1189–1232. https://doi.org/10.1214/aos/1013203451
31. Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 785–794). Association for Computing Machinery. https://doi.org/10.1145/2939672.2939785
32. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, É. (2011). Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12, 2825–2830. https://www.jmlr.org/papers/v12/pedregosa11a.html
33. Kingma, D. P., & Ba, J. (2015). Adam: A method for stochastic optimization. In Proceedings of the 3rd International Conference on Learning Representations (ICLR). https://arxiv.org/abs/1412.6980
34. Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep learning. MIT Press. https://www.deeplearningbook.org/
35. Shone, N., Ngoc, T. N., Phai, V. D., & Shi, Q. (2018). A deep learning approach to network intrusion detection. IEEE Transactions on Emerging Topics in Computational Intelligence, 2(1), 41–50. https://doi.org/10.1109/TETCI.2017.2772792
36. Vinayakumar, R., Alazab, M., Soman, K. P., Poornachandran, P., Al-Nemrat, A., & Venkatraman, S. (2019). Deep learning approach for intelligent intrusion detection system. IEEE Access, 7, 41525–41550. https://doi.org/10.1109/ACCESS.2019.2895334
37. European Union Agency for Cybersecurity. (2023). ENISA threat landscape 2023. https://www.enisa.europa.eu/publications/enisa-threat-landscape-2023
38. Tabassi, E. (2023). Artificial intelligence risk management framework (AI RMF 1.0) (NIST AI 100-1). National Institute of Standards and Technology. https://doi.org/10.6028/NIST.AI.100-1

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

INNO SPACE
SJIF Scientific Journal Impact Factor

doi cross ref

निस्केयर
NISCAIR

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

9940 572 462 6381 907 438 ijirset@gmail.com

www.ijirset.com

Scan to save the contact details