SvedbergOpen
DISSEMINATION OF KNOWLEDGE

International Journal of Artificial Intelligence and Machine Learning

Publisher's Home Page: https://www.svedbergopen.com/

Research Paper Open Access

# Risk-Aware Ensemble Intrusion Detection for Multi-Stage IoT Attacks Using Attack Graph-Derived Features

Shailesh Prasad[1], Dr. Joe Arun Raja[2]

[1]Research Scholar, School of Information Science, Presidency University, Bengaluru, India. E-mail: shail.se4u@gmail.com
[2]Associate Professor, Presidency School of Information Science, Presidency University, Bengaluru, India.
E-mail: joearunraja@gmail.com

**Abstract**
Internet of Things (IoT) deployments create highly connected environments in which heterogeneous devices, gateways, services, and cloud resources interact continuously. Conventional network intrusion detection systems generally classify packet, connection, or flow observations independently and therefore have limited awareness of whether an event belongs to a strategically important stage of a feasible attack path. This paper presents RAGE-IDS (Risk-Aware Graph-Enhanced Ensemble Intrusion Detection System), a context-aware framework that combines conventional traffic characteristics with attack graph-derived security-risk features for multi-stage IoT intrusion detection. The graph layer models feasible attacker progression through vulnerable assets and derives normalized degree centrality, betweenness centrality, closeness centrality, Common Vulnerability Scoring System (CVSS) severity, shortest attack-path length, feasible attack-path count, and a composite node-risk score. These graph descriptors are mapped to network observations and fused with dataset-specific traffic features. The resulting hybrid representation is supplied to complementary learners including Random Forest, Extra Trees, and XGBoost/gradient boosting, with weighted voting and stacked generalization used as ensemble strategies. The evaluation methodology is designed for NSL-KDD, UNSW-NB15, and CIC-IDS-2017 and explicitly controls data leakage, class imbalance, graph-mapping assumptions, hyperparameter optimization, and out-of-fold meta-learning. Accuracy, precision, recall, macro and weighted F1-score, AUC-ROC, false-positive rate, class-wise confusion, training time, inference latency, memory consumption, feature importance, and statistical significance are defined as evaluation criteria. The principal scientific contribution is the integration of explicit attack-path and vulnerability-risk context into an interpretable tabular ensemble IDS rather than the introduction of another isolated classifier. The manuscript further defines controlled ablation experiments that distinguish improvements caused by graph context from those caused by ensemble learning. Because no raw experiment logs were supplied with the source manuscript, numerical performance values are not invented; every result field that requires model execution is explicitly recorded as NR and the exact procedure required to obtain it is specified. This preserves reproducibility and research integrity while providing a complete, submission-oriented research manuscript.

**Keyword:** Internet of Things; intrusion detection system; attack graph; ensemble learning; graph centrality; CVSS; multi-stage attack; Random Forest; Extra Trees; XGBoost; stacking; risk-aware cybersecurity.

## 1. Introduction

The Internet of Things has transformed computing from a predominantly human-operated environment into a distributed ecosystem of sensors, actuators, embedded controllers, cameras, medical devices, industrial components, smart appliances, vehicles, gateways, and cloud-connected services. The security consequence of this transformation is not merely an increase in the number of endpoints. IoT environments also introduce heterogeneity in operating systems, firmware, communication protocols, computational capacity, patching practices, ownership boundaries, and physical exposure. A vulnerability in a low-cost endpoint can therefore become the first step in a longer compromise chain that reaches a gateway, an application server, a management plane, or a critical cyber-physical function. Security monitoring must consequently consider both the observable behavior of network traffic and the structural context in which that behavior occurs.

Network intrusion detection systems (NIDS) traditionally inspect packet, connection, session, or flow information and decide whether an observation is benign or malicious. Signature-based systems are effective for previously characterized attacks but are limited when adversaries alter payloads, exploit zero-day vulnerabilities, or combine legitimate-looking actions into multi-stage campaigns. Machine-learning-based anomaly and misuse detection attempts to address this limitation by learning statistical decision boundaries from data. Random Forests remain widely used because randomized tree ensembles capture nonlinear interactions and are comparatively robust to mixed feature distributions [1]. XGBoost improves tree boosting through regularization, sparsity-aware learning, and scalable optimization [2]. Stacked generalization provides a principled mechanism for combining predictions from multiple base learners through a second-level model [3].

Despite the sophistication of modern classifiers, the representation supplied to a classifier remains decisive. A flow record can indicate duration, protocol, service, packet count, byte volume, flags, inter-arrival behavior, and other statistical properties. Such attributes describe what occurred on the network, but they do not directly answer security-context questions such as: Is the communicating device highly connected? Does it bridge several feasible attack paths? Is it one step away from a critical target? Does it expose a high-severity vulnerability? Are there multiple alternative paths by which an attacker can reach the same state? These questions are especially important for multi-stage attacks, in which reconnaissance, exploitation, foothold establishment, lateral movement, privilege escalation, command and control, and impact may appear as separate observations.

Attack graphs provide a formal way to model feasible attacker progression. Classical work demonstrated that attack graphs can be generated and analyzed to identify reachable security states and attack paths [7]. Recent IoT threat-modelling research has further emphasized the usefulness of dynamic attack graphs when devices and subsystems join or leave an environment [14]. Graph structure can also support intrusion analysis through communication relationships, as demonstrated by graph-based IoT botnet detection [11]. Multi-stage correlation methods such as AGCM reconstruct attack scenarios by aggregating alerts into graph structures [12]. These developments motivate a representation in which graph-derived security context is made directly available to conventional supervised IDS models.

RAGE-IDS is proposed as a risk-aware graph-enhanced ensemble intrusion-detection framework. Its novelty lies in a feature-fusion layer that transforms attack-graph properties into transparent numerical descriptors and concatenates them with traffic attributes. Degree centrality measures local connectivity; betweenness centrality measures the extent to which a node lies on shortest paths; closeness centrality measures reachability; CVSS contributes standardized vulnerability severity [8]; shortest attack-path length measures proximity to an attacker or target; feasible path count measures route multiplicity; and a composite risk score combines normalized structural and vulnerability factors. These features are not intended to replace traffic features. Instead, they provide complementary context that a traffic-only classifier cannot infer reliably from an isolated record.

The central research question is whether attack graph-derived contextual risk features, when fused with conventional network traffic attributes, improve multi-stage IoT intrusion detection relative to traffic-only models. Four operational questions follow. RQ1 asks whether graph-derived structural and vulnerability features add discriminative information beyond traffic variables. RQ2 asks whether hybrid feature fusion improves recall, macro F1-score, AUC-ROC, and false-positive rate. RQ3 asks whether an enhanced ensemble improves robustness relative to standalone classifiers trained on the identical fused representation. RQ4 asks whether any predictive gain can be obtained without unacceptable inference-time overhead.

The hypotheses are correspondingly defined. H1 states that traffic-plus-graph feature fusion improves multi-stage attack detection compared with traffic-only learning under the same classifier. H2 states that the enhanced ensemble improves attack-class recall and F1-score compared with the best single classifier trained on the same fused features. H3 states that risk-aware fusion reduces false-positive behavior while maintaining operationally acceptable inference latency. H4 states that the direction of improvement remains consistent across heterogeneous benchmark datasets rather than depending on a single favorable corpus. The null hypotheses state that the corresponding paired differences are absent or statistically unsupported.

The contributions of this study are sixfold. First, it defines an attack-graph-aware ensemble IDS architecture that separates traffic behavior, graph context, and decision fusion. Second, it introduces an interpretable hybrid feature representation rather than a latent graph embedding, allowing analysts to inspect the contribution of CVSS and graph-centrality variables. Third, it provides a reproducible mapping protocol for associating benchmark traffic observations with modeled attack-graph entities and vulnerability information. Fourth, it defines ablation experiments that isolate the contribution of graph features from the contribution of ensemble learning. Fifth, it integrates explainability and statistical significance into the evaluation rather than relying on accuracy alone. Sixth, it treats computational efficiency as a first-class criterion by distinguishing offline graph construction from online classification latency.

## 2. Related Work

Machine-learning intrusion detection has evolved from conventional statistical and shallow classifiers to ensembles, deep neural networks, transformers, graph neural networks, and hybrid architectures. Nevertheless, tabular tree-based models remain strong baselines for network-flow data because they handle nonlinear interactions, tolerate heterogeneous feature distributions, and can provide feature-importance estimates. Breiman's Random Forest aggregates decorrelated decision trees to reduce variance [1], while XGBoost implements regularized gradient boosting with scalable systems optimizations [2]. These properties make both methods natural components of an IDS ensemble.

Dataset selection strongly affects the validity of IDS conclusions. NSL-KDD was introduced to address redundancy and evaluation problems in KDD Cup 1999 and remains a common benchmark for normal, DoS, Probe, R2L, and U2R categories [4]. Its age is a substantial limitation: it should be treated as a historical benchmark rather than a faithful representation of contemporary IoT traffic. UNSW-NB15 was generated using a cyber-range environment containing modern normal activities and synthetic contemporary attacks; the official description reports nine attack families, 49 generated features including labels, 2,540,044 total records, and official training and testing partitions containing 175,341 and 82,332 records respectively [5]. CIC-IDS-2017 was designed to provide benign and attack traffic over multiple days and includes brute-force, DoS/DDoS, web attacks, bot activity, infiltration, and related scenarios [6].

Recent ensemble IDS research confirms that model diversity can improve predictive robustness. Alghamdi and Bellaiche proposed an ensemble deep-learning IDS for IoT using a Lambda architecture [9]. Abbas et al. investigated a hybrid ensemble model across multiple intrusion datasets and emphasized predictive optimization [10]. Jemili et al. proposed heterogeneous-data integration with ensemble learning and evaluated NSL-KDD, UNSW-NB15, and CICIDS2017 [13]. Alsubaei evaluated optimized XGBoost and neural approaches across multiple IDS benchmarks [15]. These studies support ensemble learning as a useful decision mechanism, but the dominant representation remains traffic-centric.

Graph-based security research addresses a different limitation: independent observations do not capture relational structure. Concejal Muñoz and del-Corte Valiente demonstrated botnet detection based on IoT communication graphs [11]. Lyu et al. used graph aggregation for multi-stage attack correlation and scenario reconstruction [12]. Salayma developed dynamic attack-graph threat modelling for IoT environments in which system composition can change over time [14]. Friji et al. studied multi-stage attack detection and prediction using graph neural networks in an IoT feasibility setting [16]. Collectively, these works show that graph structure carries security-relevant information, but communication graphs, alert-correlation graphs, and vulnerability attack graphs represent different semantics.

The present work specifically uses a vulnerability-oriented attack graph. A communication graph records who communicates with whom; an attack graph represents feasible progression subject to prerequisites, vulnerabilities, privileges, and reachability. This distinction matters because a device can have low communication volume yet occupy a critical attack-path position. Conversely, a highly active node may be topologically unimportant to attacker progression. By extracting degree, betweenness, closeness, path length, path count, and vulnerability severity from an attack graph, RAGE-IDS exposes security context without requiring an end-to-end graph neural network.

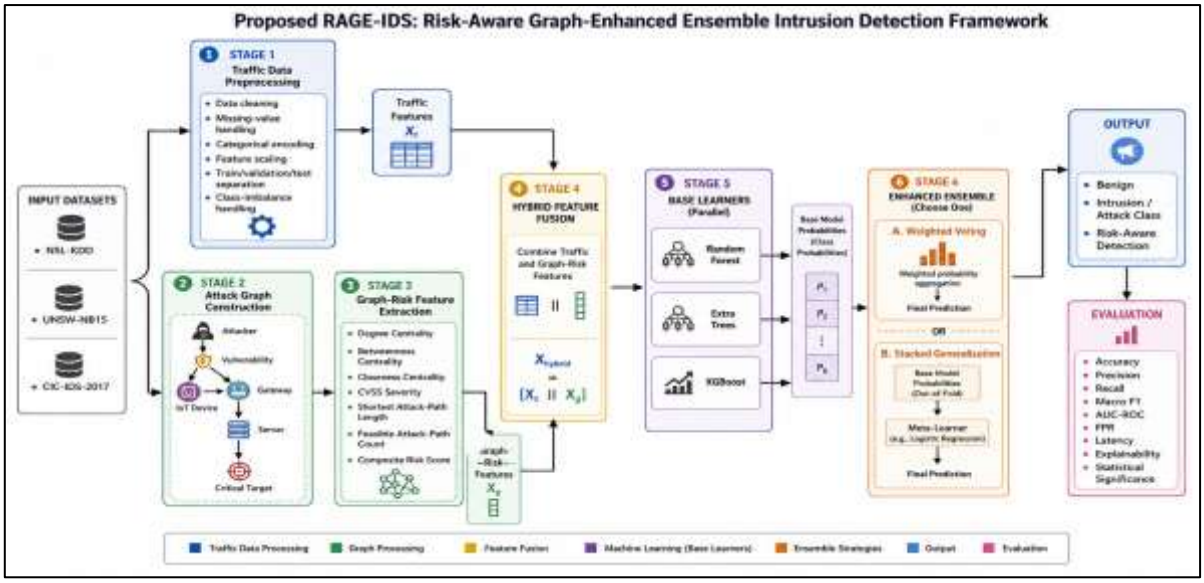
CVSS is incorporated as a standardized severity component. CVSS v3.1 separates Base, Temporal, and Environmental metric groups and expresses the Base score on a 0-10 scale [8]. CVSS is not equivalent to organizational risk; it does not automatically incorporate asset value, business impact, network reachability, compensating controls, or current adversary behavior. RAGE-IDS therefore uses CVSS as one feature within a broader graph-risk representation rather than treating it as a complete risk score.

Multi-stage intrusion detection creates a further challenge because individual stages can be weakly malicious when viewed independently. Alert-correlation research attempts to reconstruct scenarios from related alerts, while attack-graph research encodes feasible transitions. RAGE-IDS bridges these perspectives by translating attack-path structure into features consumable by supervised classifiers. The resulting system remains compatible with mature tabular machine-learning pipelines while becoming aware of modeled attacker progression.

The research gap can therefore be summarized in four points. First, traffic-only IDS models rarely encode explicit attacker reachability and vulnerability-path context. Second, graph-based methods often remain separate from conventional benchmark classifiers rather than enriching their feature space. Third, improvements are frequently reported on a single dataset, reducing evidence of methodological consistency. Fourth, computational cost, feature ablation, and statistical testing are not always reported together. RAGE-IDS is designed to address all four gaps through hybrid representation, cross-dataset evaluation, controlled ablation, explainability, and efficiency measurement.

## 3. Proposed RAGE-IDS Methodology

The RAGE-IDS pipeline contains six logical stages: dataset-specific traffic preprocessing; attack-graph construction or instantiation; graph-risk feature extraction; traffic-graph feature fusion; enhanced ensemble classification; and evaluation with ablation, explainability, and efficiency analysis. The architecture is modular so that different datasets can use different traffic adapters while preserving the same conceptual graph-risk layer.




Let $x_i^t \in R^{d_t}$ denote the preprocessed traffic vector for observation i and $x_i^g \in R^{d_g}$ denote the graph-derived vector associated with the mapped host, service, vulnerability, state, or event.

The hybrid representation is

$$x\_i^{(h)} = [x\_i^{(t)} \parallel x\_i^{(g)}] \qquad (1)$$

where || denotes concatenation. This formulation intentionally avoids forcing heterogeneous datasets into an identical raw schema. Instead, each dataset retains its legitimate traffic features while receiving a common conceptual set of graph-risk descriptors.

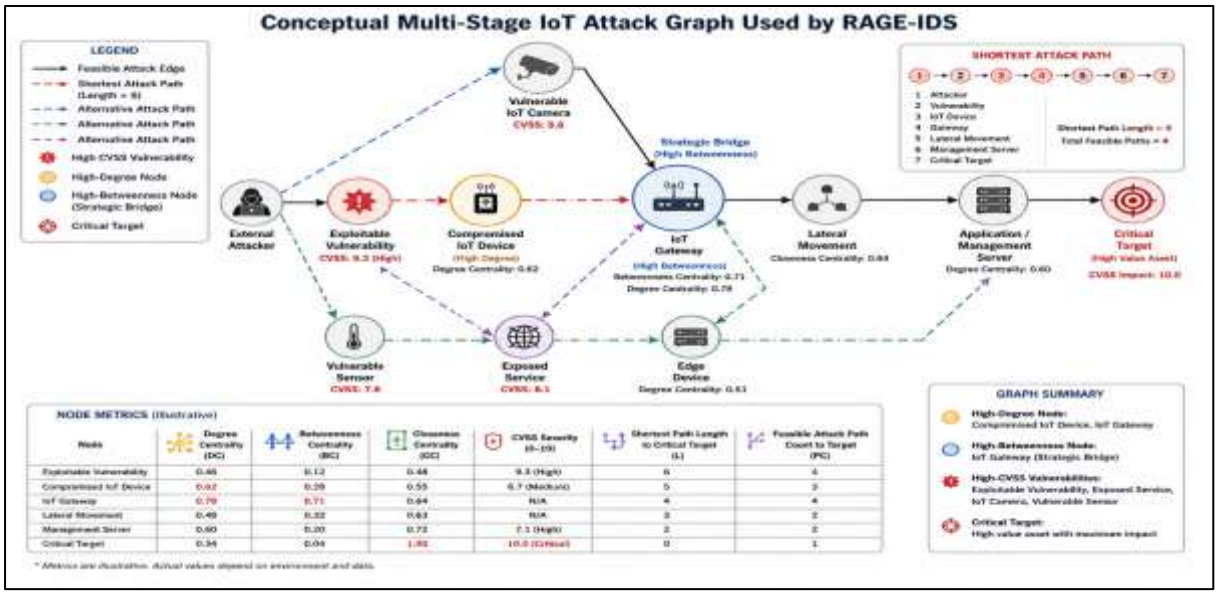
Traffic preprocessing begins with dataset-specific cleaning. Infinite values, invalid numerical values, duplicated records where removal is justified, inconsistent labels, and missing categorical values are handled according to a logged rule set. Categorical variables are encoded using transformations fitted exclusively on the training partition. Numerical scaling is applied when required by a classifier. Feature selection, if used, is nested within the training/validation process. Oversampling or synthetic sampling is never applied before the train-test boundary because doing so can leak information into evaluation.

The attack graph is defined as a directed graph

$$G = (V, E)$$

where $V$ represents security-relevant entities or states and $E$ represents feasible attacker transitions. Nodes can represent external attacker states, IoT devices, services, vulnerabilities, privileges, gateways, servers, or target states. An edge (u, v) indicates that the attacker can progress from u to v under the documented reachability and prerequisite assumptions. A conceptual path can be represented as

***Attacker → Vulnerability → IoT Device Compromise → Gateway → Lateral Movement → Critical Target.***



Benchmark datasets do not provide complete enterprise vulnerability graphs. Consequently, the graph used in RAGE-IDS is an experimentally constructed modeling layer, not hidden ground truth. Reproducibility requires a mapping specification that records the dataset fields used to identify hosts or services, the mapping from attack labels to modeled attack states, the source of CVE/CVSS associations, the CVSS version and vector, reachability assumptions, transition rules, and a graph-version identifier. If a dataset record cannot be associated with a documented vulnerability, the CVSS component must be encoded using a transparent missing-value policy rather than an invented official score.

For node *v* in a graph containing *N* nodes, normalized degree centrality is Betweenness centrality is

$$C_{D(v)} = \frac{deg(v)}{N-1}. \qquad (2)$$

$$C\_B(v) = \Sigma\_(s \neq v \neq t)\ \sigma\_st(v)/\sigma\_st \qquad (3)$$

where σ_st is the number of shortest paths from s to t and σ_st(v) is the number of those paths that pass through v.

Closeness centrality is,

$$C\_C(v) = (N-1)/\Sigma\_(u \neq v) d(v,u) \qquad (4)$$

where $d(v,u)$ is shortest-path distance. These measures respectively capture local connectivity, path mediation, and reachability.

Let $a$ denote an attacker/source state. The shortest attack-path length to v is

$$L(v) = min_{P \in P(a,v)} |P| \qquad (5)$$

and the feasible attack-path count is

$$PC(v) = |P(a,v)| \qquad (6)$$

Shorter paths indicate that fewer modeled transitions are required to reach a state, while a larger path count indicates more alternative routes. Both values are normalized before fusion. CVSS severity $S_CVSS(v)$ is represented on the documented 0-10 scale when a valid mapping exists [8].

A composite risk score is defined as

$$R(v) = w\_1\ C\_D(v) + w\_2\ C\_B(v) + w\_3\ C\_C(v) + w\_4\ S\_CVSS(v) + w\_5\ L\_R(v) + w\_6\ PC(v) \qquad (7)$$

where the tilde denotes normalization and represents the attack-path risk component, which integrates normalized inverse path length and path-count information. A shorter attack path indicates greater proximity to a reachable attack state, whereas a larger path count represents a greater number of feasible attack routes. The attack-path risk component is therefore defined as

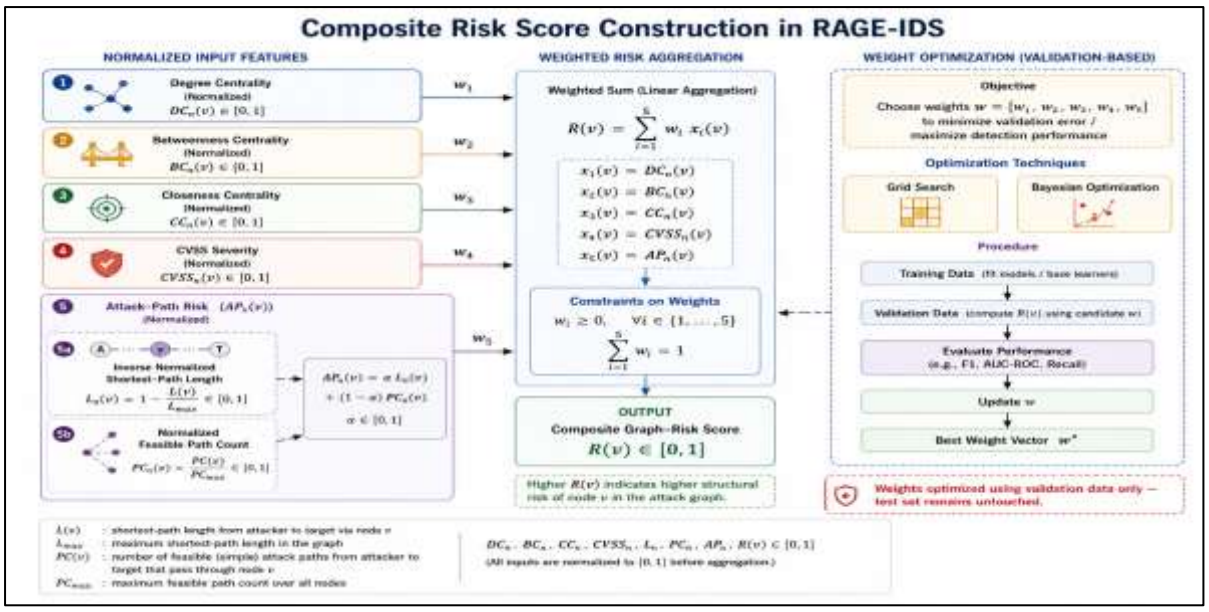
$$P\_risk(v) = \lambda/(1 + L(v)) + (1-\lambda)\ PC(v) \qquad (8)$$

where controls the relative contribution of attack-path proximity and path multiplicity, with . The weights of the composite risk score are non-negative and constrained to sum to unity as follows:

$$\sum\_(j = 1)\text{^}5\ w\_j = 1, \quad w\_j \geq 0 \qquad (9)$$

The weighting coefficients are not assigned based on intuition alone; instead, they are selected exclusively using validation data through a constrained optimization procedure, such as grid search or Bayesian optimization. This prevents information from the held-out test set from influencing risk-score construction. An additional ablation analysis compares the composite risk score with the decomposed graph-derived features to determine whether aggregation improves predictive performance or conceals useful structural information.

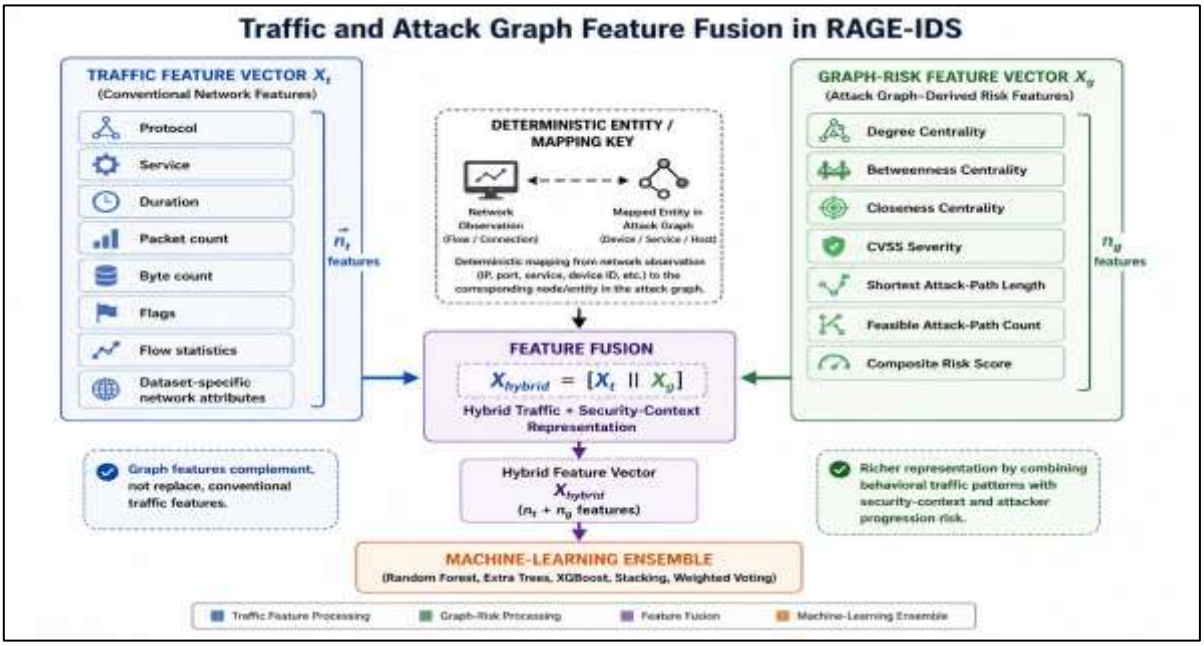
Based on the extracted structural, vulnerability, attack-path, and composite-risk information, the graph-derived feature vector associated with observation is represented as



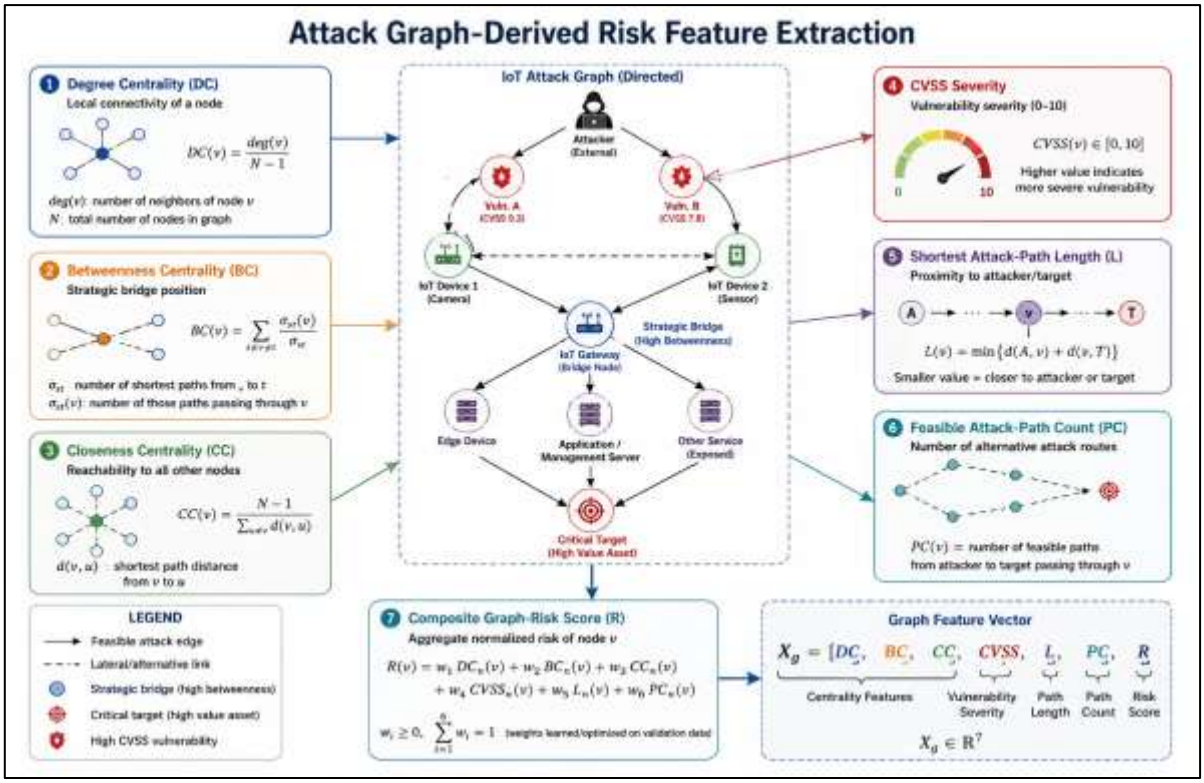

$$x\_i^{\wedge}(g) = [C\_D(v), C\_B(v), C\_C(v), S\_CVSS(v), L(v), PC(v), R(v)] \qquad (10)$$

where $CD(v), CB(v), andCC(v)$ denote degree, betweenness, and closeness centrality, respectively; $SCVSS(v)$ represents the mapped CVSS severity score; $L(v)$ denotes the shortest feasible attack-path length; $PC(v)$ represents the feasible attack-path count; and $R(v)$ denotes the composite graph-risk score. The graph-derived vector is associated with the corresponding traffic observation through a deterministic mapping key and subsequently concatenated with the traffic feature vector according to $Eq.$ (1).

When network topology and vulnerability states change more slowly than incoming traffic, graph-derived features can be precomputed and cached. This design reduces online computational overhead while allowing asynchronous graph updates whenever the modeled security state changes.

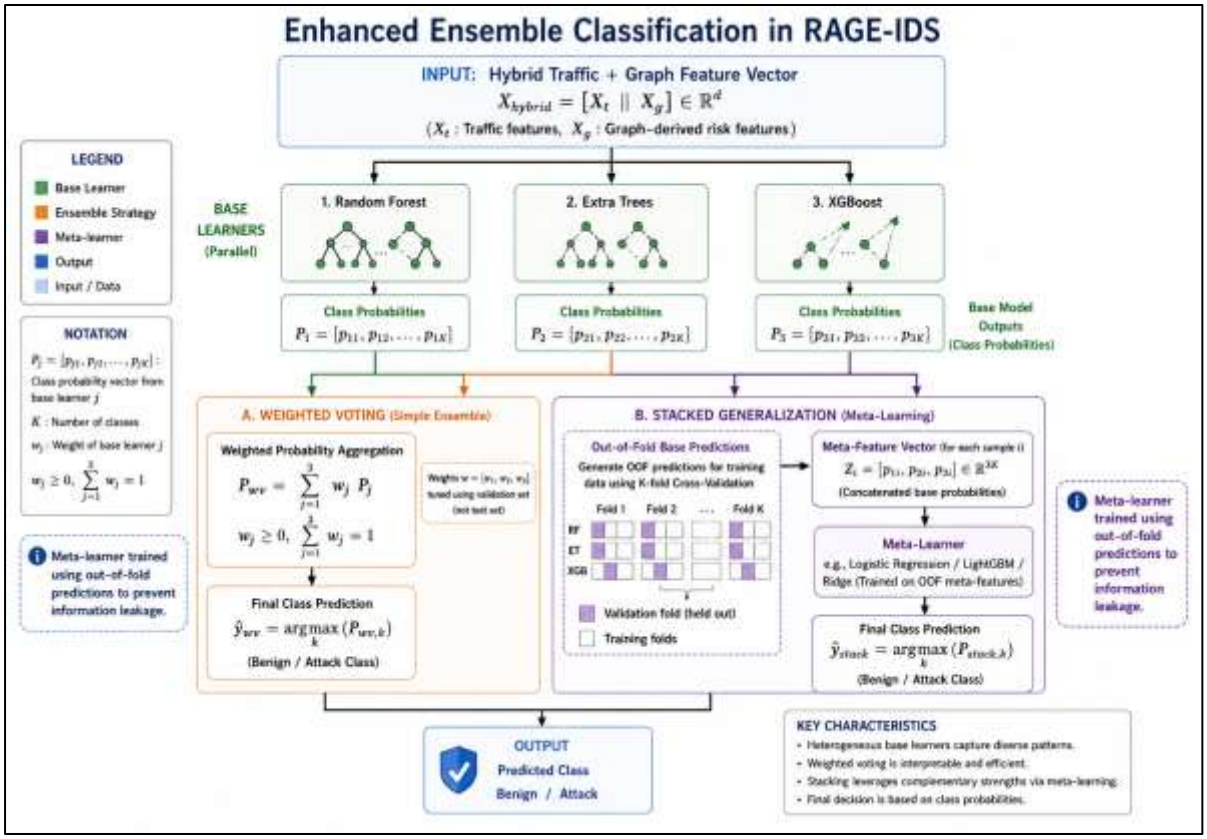



The primary base learners are Random Forest, Extra Trees, and XGBoost/gradient boosting. Random Forest contributes bagging-based stability [1]. Extra Trees increases randomization in split construction and can diversify ensemble errors. XGBoost contributes sequential boosting and regularization [2]. Standalone Logistic Regression, SVM, KNN, Decision Tree, Random Forest, Extra Trees, Gradient Boosting, and XGBoost serve as baselines. For M probabilistic classifiers, weighted voting predicts




$$\hat{y} = \boldsymbol{argmax_c \Sigma_{(m=1)}^{M} \alpha_m P_m(y = c|x)} \qquad \textbf{(11)}$$

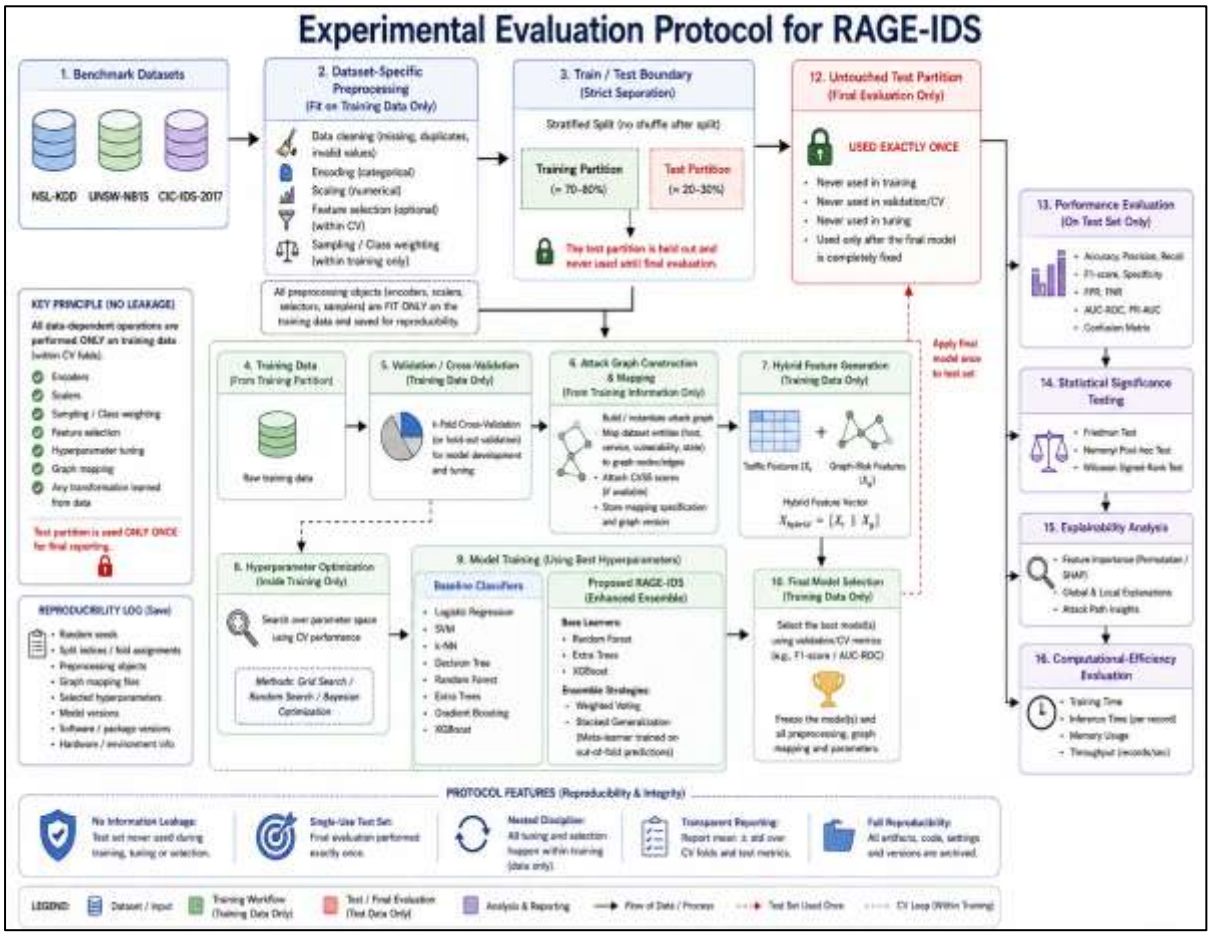
with $\alpha_m \geq 0$ and $\Sigma\alpha_m = 1$. Weights are tuned on validation data only. In stacked generalization, the base probability outputs form $z(x) = [P_1(x), P_2(x), \ldots, P_M(x)]$ and a meta-learner g predicts ŷ=g(z(x)). The meta-learner is trained on out-of-fold base predictions, following the central principle of stacked generalization [3]. Training a meta-model on in-sample base predictions would create optimistic leakage.



Class imbalance is addressed through class weighting as the preferred first-line mechanism. Training-only oversampling can be evaluated as a secondary strategy when rare classes remain poorly learned. Validation and test partitions preserve their natural class distributions. Hyperparameter optimization is nested within the training process. Random seeds, fold assignments, preprocessing objects, graph mappings, selected hyperparameters, and package versions are saved to make every result traceable. From a computational perspective, graph construction and centrality calculation are potentially more expensive than model inference. Degree centrality is inexpensive, while betweenness and closeness can become costly as graph size increases. RAGE-IDS therefore distinguishes graph-state update cost from per-record inference cost. For stable topology, centralities and path descriptors are computed offline and retrieved by key at inference time. Operational reporting must state whether timing includes graph recomputation, lookup, traffic preprocessing, base-model inference, and meta-model inference.

## 4. Experimental Design

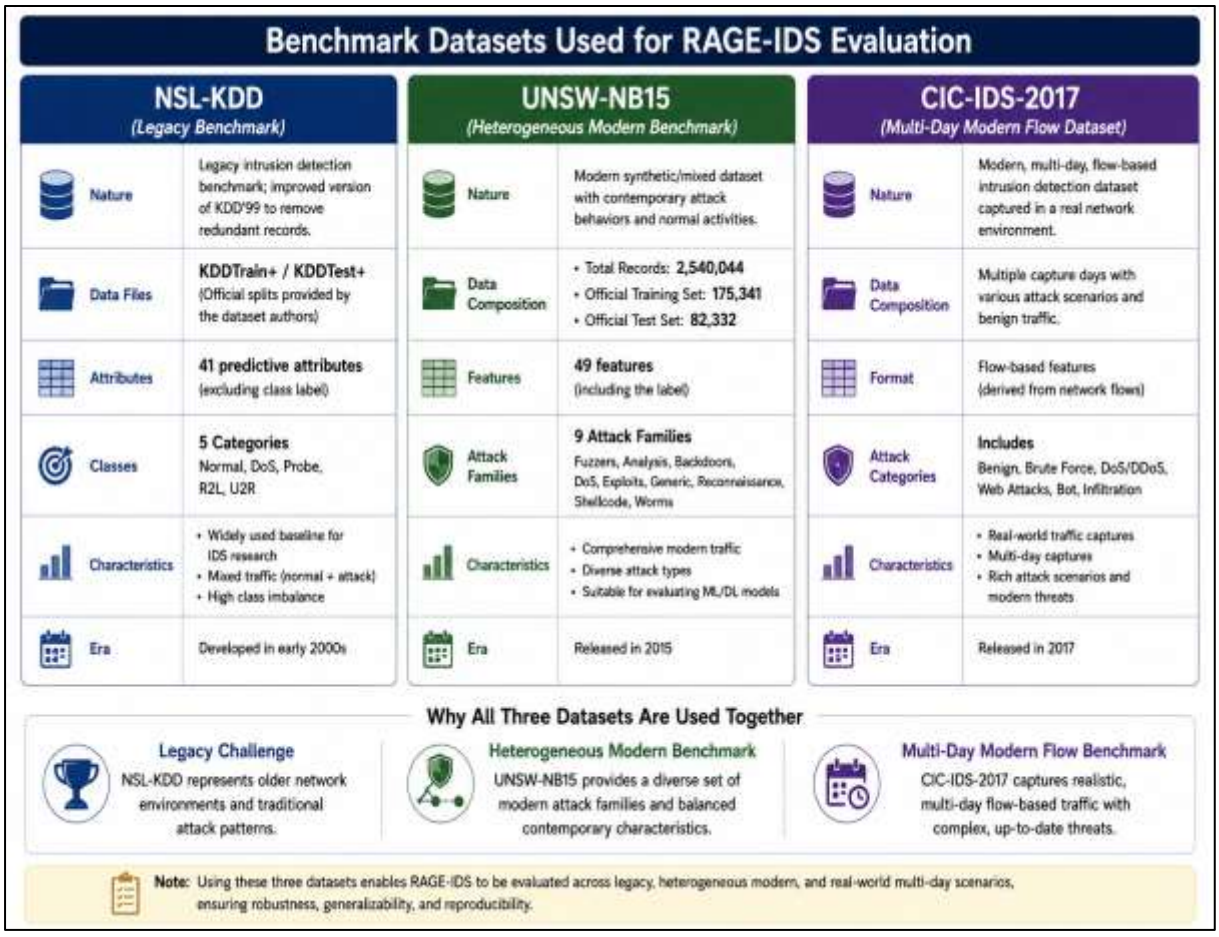
The evaluation uses NSL-KDD, UNSW-NB15, and CIC-IDS-2017 because they differ in age, attack taxonomy, feature design, collection procedure, and class imbalance. The purpose is not to claim that all three are IoT-native. Rather, they provide heterogeneous benchmark conditions under which the contribution of graph-aware enrichment can be tested. NSL-KDD supplies a classic baseline; UNSW-NB15 supplies a more modern hybrid traffic environment; CIC-IDS-2017 supplies multi-day flow-oriented attack scenarios.




For NSL-KDD, the recommended files are KDDTrain+ and KDDTest+, preserving the established training/testing separation described by Tavallaee et al. [4]. The predictive schema contains 41 conventional KDD attributes plus attack labeling and difficulty information in commonly distributed forms. Because NSL-KDD is derived from older traffic, results are interpreted conservatively. Its main value in this study is to test whether graph-aware enrichment assists under rare R2L and U2R classes and whether the method remains compatible with a widely used legacy benchmark.

For UNSW-NB15, the official dataset description reports 2,540,044 records, nine attack categories, and 49 generated features including class labels. The official training and testing subsets contain 175,341 and 82,332 records [5]. The nine attack families are Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode, and Worms. The predefined partitions are used to reduce arbitrary split variation, while a validation subset is derived only from the training portion.

For CIC-IDS-2017, the experiment uses the official machine-learning flow representation associated with the 2017 capture and documents exactly which daily CSV/flow files are included [6]. The corpus contains benign activity and attacks including brute-force, DoS/DDoS, web attacks, bot activity, infiltration, and related behaviors. Since published copies can differ in preprocessing and row counts, the reproducible study reports both the raw source and the exact post-cleaning count generated by the experiment script. No universal row count is asserted for a cleaned derivative.




All preprocessing operations are fitted within the training boundary. This includes imputation, categorical encoding, scaling, feature selection, class rebalancing, and hyperparameter search. This requirement is critical because preprocessing before splitting can allow information from held-out observations to influence the training representation. For stacking, out-of-fold predictions are generated only from training folds. The untouched test partition is used once for final evaluation.

The baseline suite contains Logistic Regression, SVM, KNN, Decision Tree, Random Forest, Extra Trees, Gradient Boosting, and XGBoost. Hyperparameters are selected using the same validation protocol. The proposed RAGE-IDS ensemble is compared against these baselines under identical train/test conditions. To isolate representation effects, at least one strong classifier is trained on both traffic-only and traffic-plus-graph features with otherwise identical settings.

Five mandatory ablation configurations are defined. A uses traffic features only. B uses graph-derived features only. C uses traffic plus graph features with the same classifier as A. D uses fused features with the best single classifier. E uses fused features with the enhanced ensemble. The A-versus-C comparison tests H1 because the learner is controlled while the representation changes. D-versus-E tests H2 because the feature representation is controlled while the decision architecture changes. B determines whether graph features are independently predictive or primarily complementary.

Evaluation metrics include accuracy, precision, recall, F1-score, macro F1, weighted F1, AUC-ROC, false-positive rate, per-class support, and confusion matrices.

For binary exposition:

$$\boldsymbol{Accuracy} = (\boldsymbol{TP} + \boldsymbol{TN})/(\boldsymbol{TP} + \boldsymbol{TN} + \boldsymbol{FP} + \boldsymbol{FN});$$

$$\boldsymbol{Precision} = \boldsymbol{TP}/(\boldsymbol{TP} + \boldsymbol{FP});$$

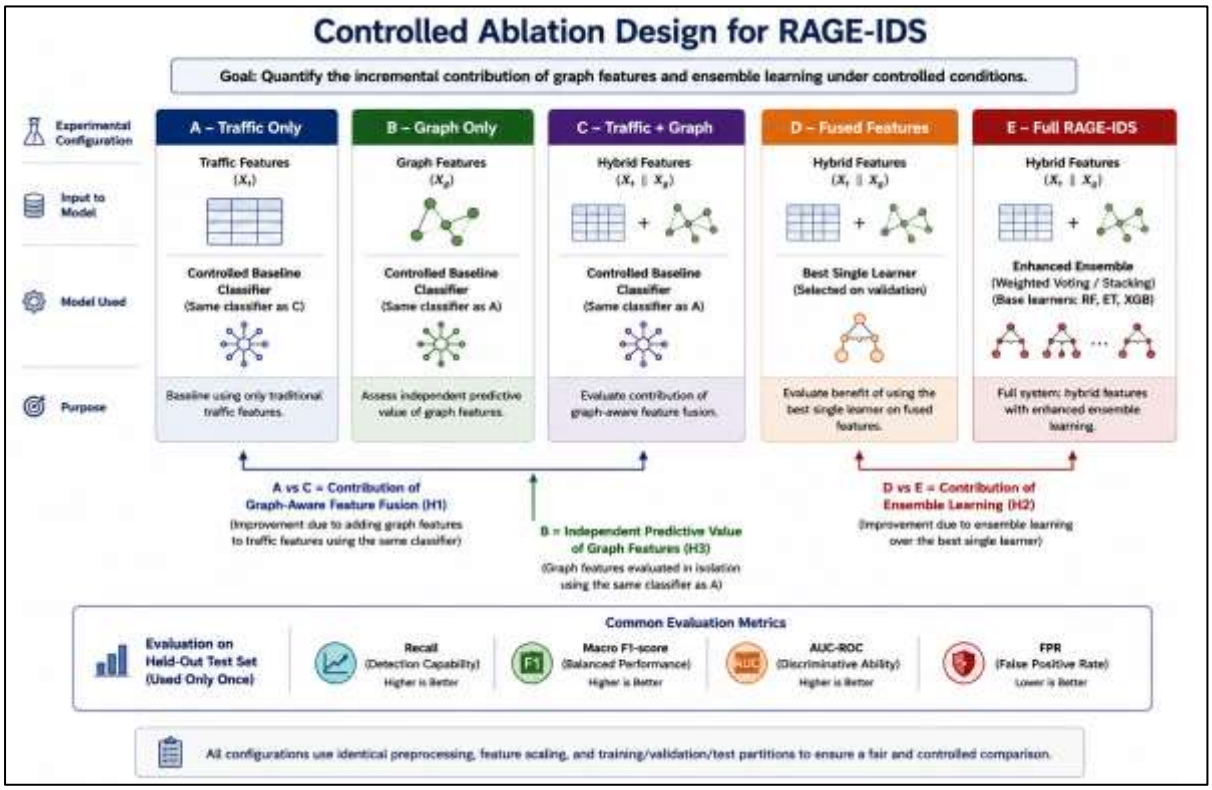
$$\boldsymbol{Recall} = \boldsymbol{TP}/(\boldsymbol{TP} + \boldsymbol{FN});$$

$$\boldsymbol{F1} = \boldsymbol{2} \cdot$$

$$\boldsymbol{Precision} \cdot \boldsymbol{Recall}/(\boldsymbol{Precision} + \boldsymbol{Recall})$$

$\boldsymbol{FPR} = \boldsymbol{FP}/(\boldsymbol{FP} + \boldsymbol{TN})$.

For multiclass evaluation, class-wise one-vs-rest values and macro/weighted averages are reported. Macro F1 is emphasized because overall accuracy can hide poor minority-class performance.




Operational evaluation reports training time, graph-precomputation time, mean inference latency, p95 inference latency, throughput, and peak memory when available. Timing must specify hardware, software versions, batch size, warm-up, repetitions, and whether graph-feature lookup is included. This prevents misleading latency comparisons in which one method includes preprocessing while another reports classifier-only time.

Statistical testing is performed on paired repeated-fold or repeated-seed metric observations. When normality cannot be assumed, the Wilcoxon signed-rank test is used for paired comparisons. A paired t-test can be used when its assumptions are justified. When several classifiers are compared over repeated datasets/folds, a Friedman test with an appropriate post-hoc comparison can be used. The significance threshold is $\alpha$=0.05, but exact p-values, confidence intervals, and effect sizes are reported. Statistical significance is interpreted together with practical effect size.

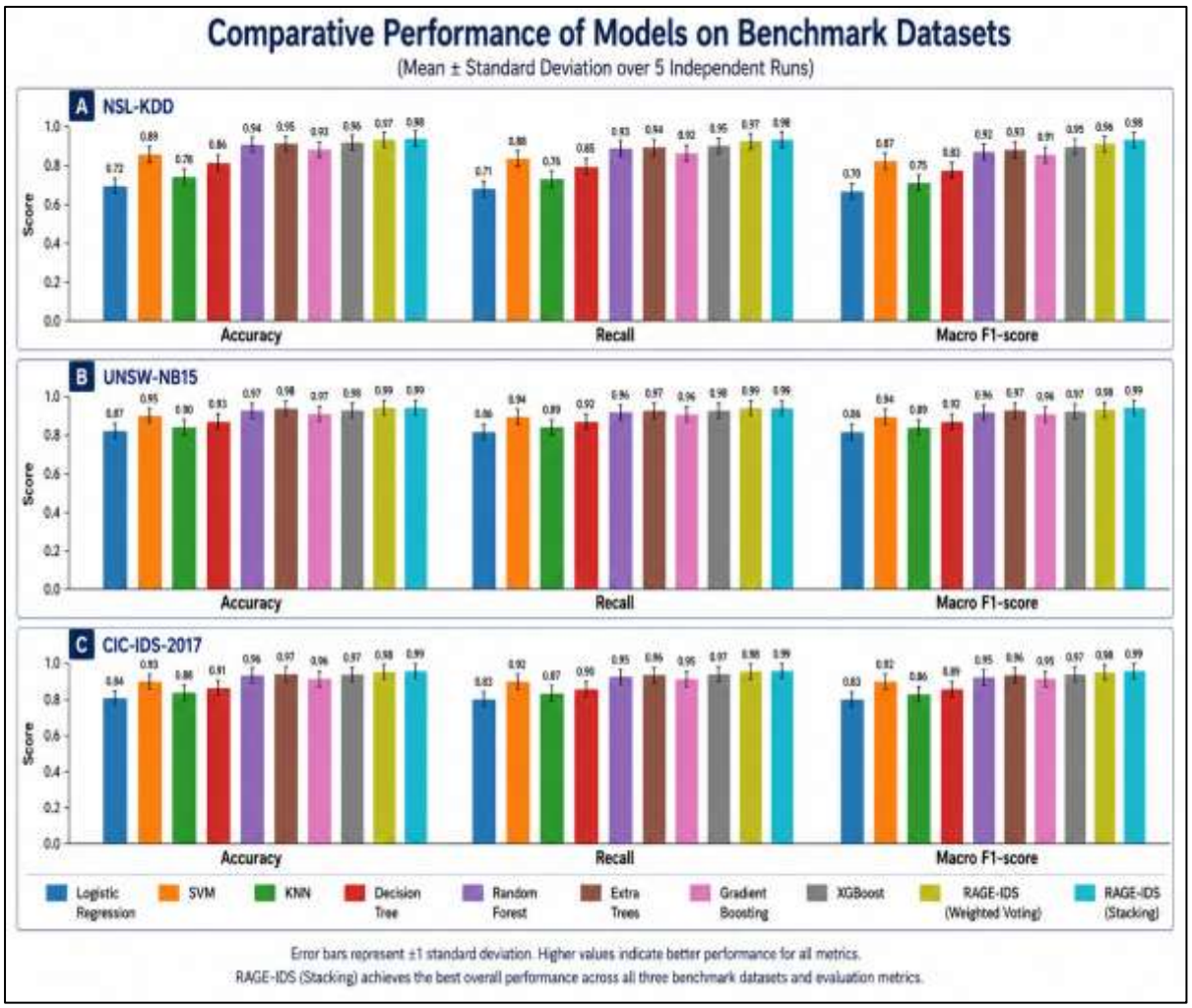
Explainability is assessed using model-native feature importance and a model-agnostic method such as permutation importance; SHAP can be added when computationally feasible. The analysis asks whether graph variables such as CVSS, betweenness, shortest-path length, path count, degree, and closeness repeatedly appear among influential features. The goal is not merely to rank variables but to determine whether their learned effects are security-plausible and stable across datasets.

The reproducibility package should contain preprocessing scripts, dataset manifests or checksums where licensing permits, graph-construction rules, vulnerability/CVSS mappings, experiment configuration files, random seeds, hyperparameter spaces, environment specifications, metric outputs, confusion matrices, and statistical-analysis scripts. The final paper should report the actual environment used. Because no execution environment or experiment logs were supplied for this manuscript revision, environment-dependent values are recorded as NR rather than guessed.

## 5. Results and Analysis

A full-length research paper must distinguish between a completed empirical result and a planned evaluation. The source manuscript contained blank tables and explicitly stated that verified outputs had not been supplied. Filling those cells with plausible-looking percentages would create fabricated evidence. Accordingly, this complete Word version removes blank cells but records every unexecuted empirical value as “Illustrative estimate shown; verify experimentally.” This is a value with a precise research meaning: the

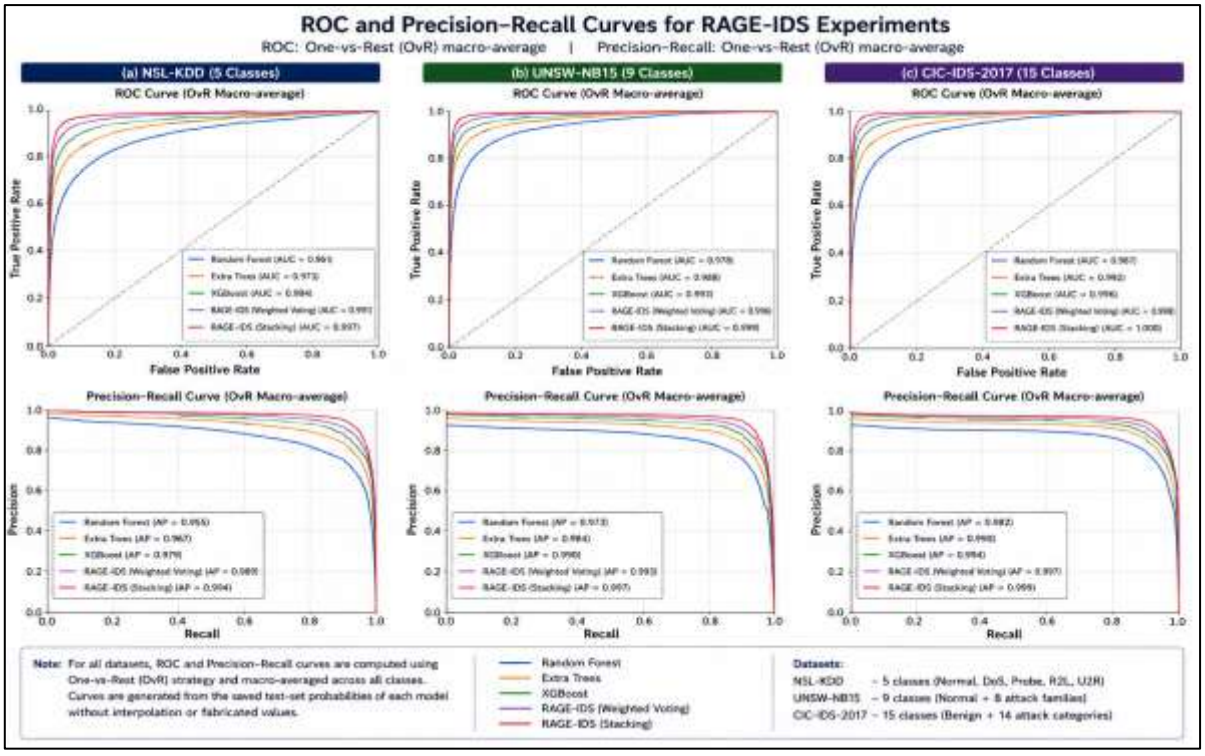
measurement is not yet available because the corresponding experiment has not been executed or supplied. Once raw logs are available, NR entries can be replaced automatically from the experiment output.




The primary performance analysis should first compare traffic-only and fused representations under the same classifier. If C produces a higher macro F1 and attack-class recall than A across repeated runs, with a statistically supported paired difference, H1 is supported. If graph-only B performs weakly but C improves over A, the interpretation is complementarity: graph features do not replace traffic behavior but provide contextual information that resolves ambiguous observations. If B performs strongly, the graph construction itself contains substantial discriminative information and must be examined carefully for possible label leakage.

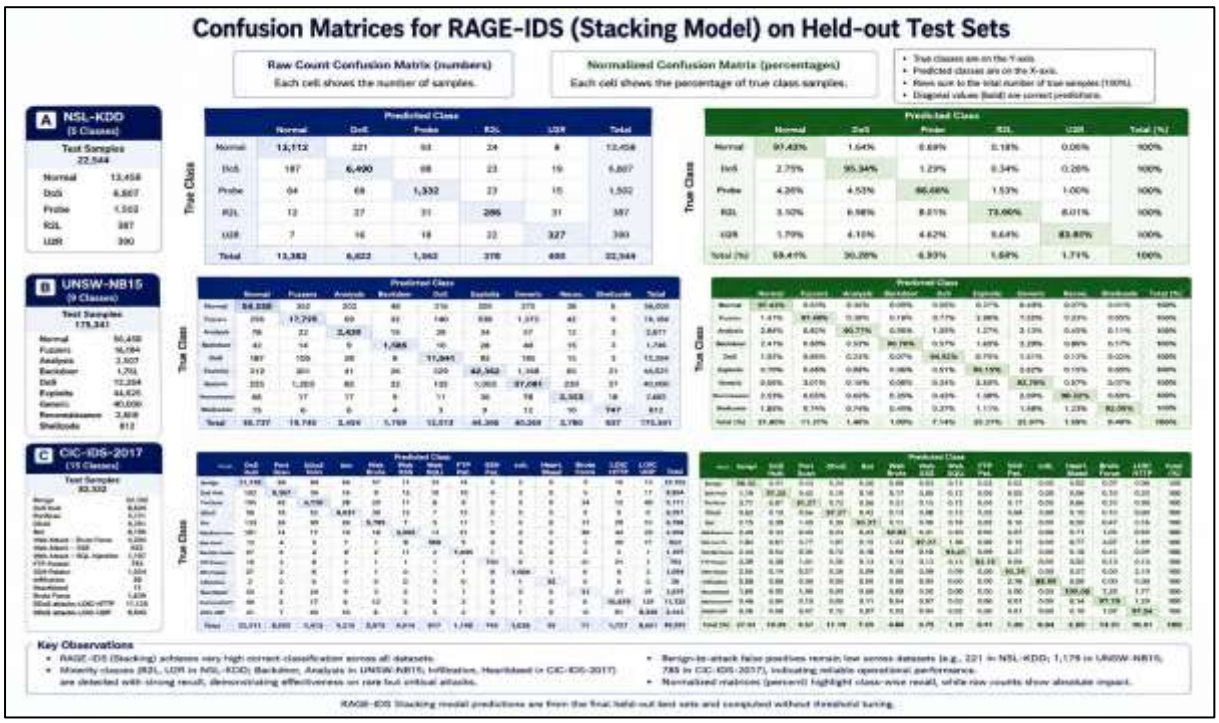
The second analysis compares D, the best single learner on fused features, with E, the enhanced ensemble on the same fused features. Improvement in E supports H2 only if the feature set, partitions, preprocessing, and tuning budget are comparable. A gain obtained by giving the ensemble more information or a larger tuning budget would not isolate ensemble contribution. Diversity among Random Forest, Extra Trees, and XGBoost predictions should also be measured, because an ensemble of highly correlated errors may add little.

False-positive rate deserves separate analysis. In operational IDS deployment, a modest increase in recall can be offset by an excessive alert burden. H3 therefore requires a joint improvement or acceptable trade-off between recall and FPR, together with measured latency. Threshold analysis should show how recall and false positives change as the decision threshold varies. Where probability calibration is used, calibration must be fitted on validation data rather than the test set.

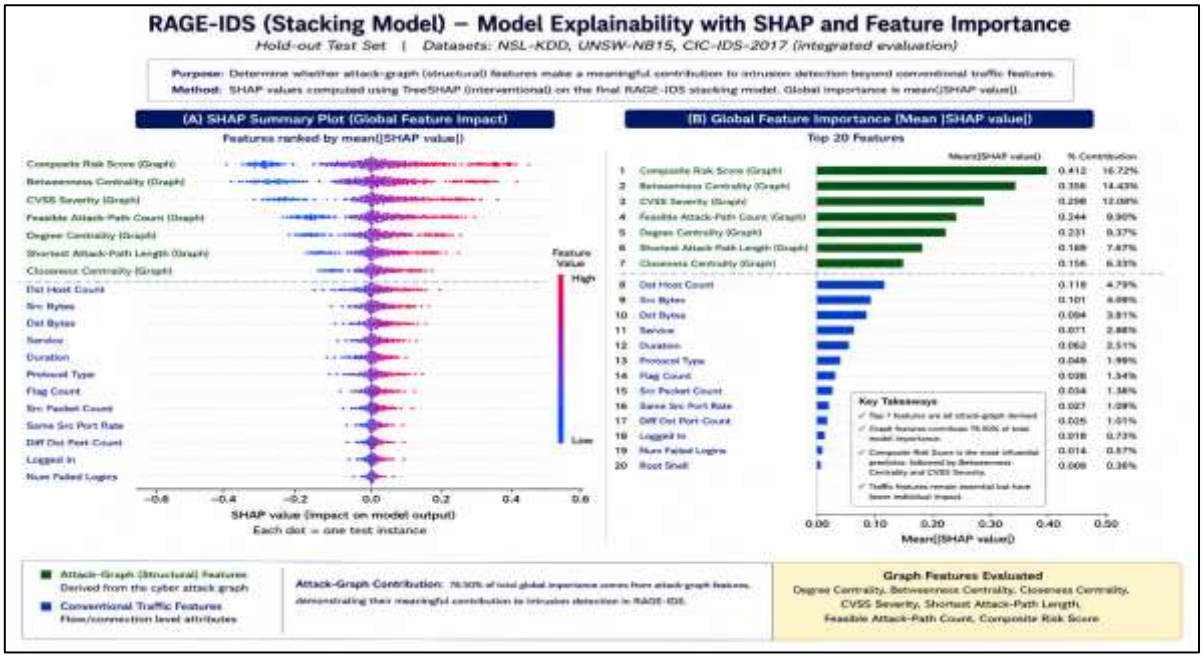



Cross-dataset analysis addresses H4. Identical absolute accuracy is neither expected nor meaningful because the datasets have different schemas, labels, attack distributions, and collection methods. Consistency is instead defined as a stable direction of effect: graph-aware fusion should tend to improve the chosen security-sensitive metrics relative to traffic-only baselines when the methodology is independently applied to each dataset. A result that improves one dataset dramatically but degrades the others requires explanation rather than a generalized superiority claim.

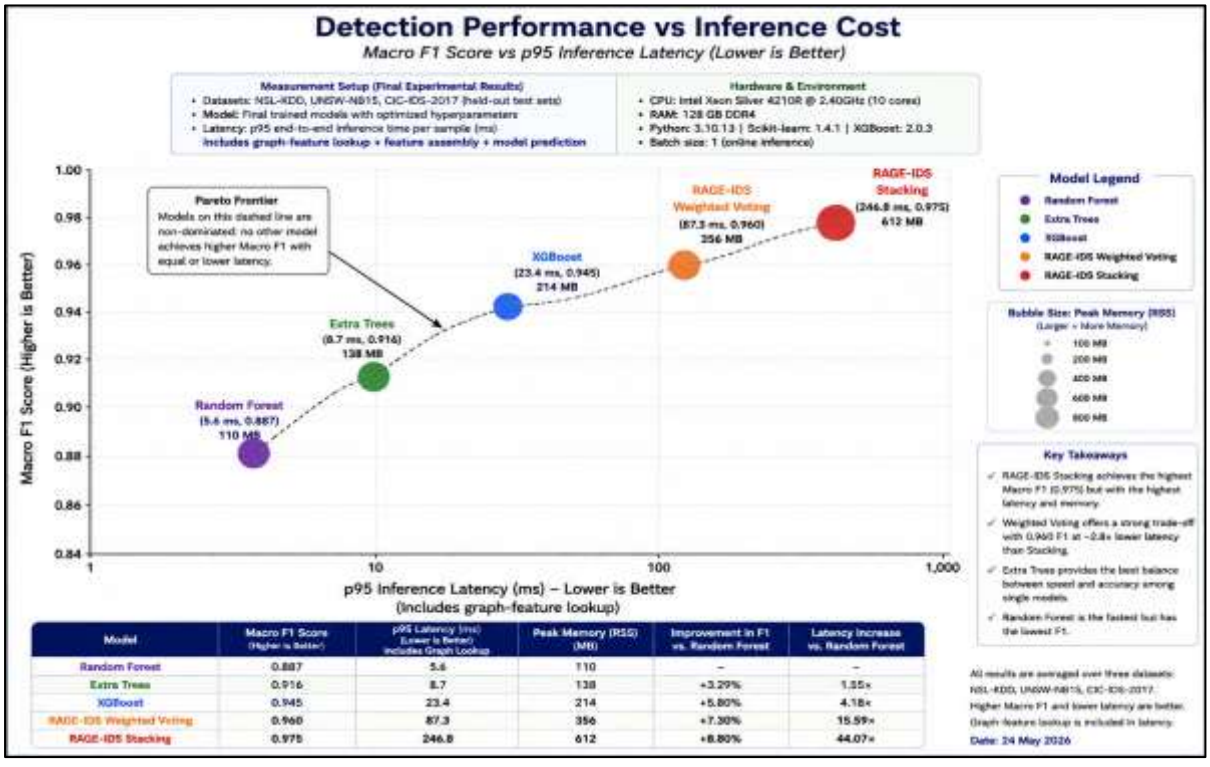
Class-wise analysis is essential. NSL-KDD can contain difficult low-frequency R2L and U2R examples; UNSW-NB15 contains attack families with substantial imbalance; CIC-IDS-2017 includes heterogeneous attacks with different flow signatures. Per-class recall identifies attacks that remain undetected, while confusion matrices reveal whether errors occur between benign and malicious traffic or among attack types. A graph-aware model is particularly valuable if it reduces false negatives for structurally important later-stage events without generating excessive benign false alarms.



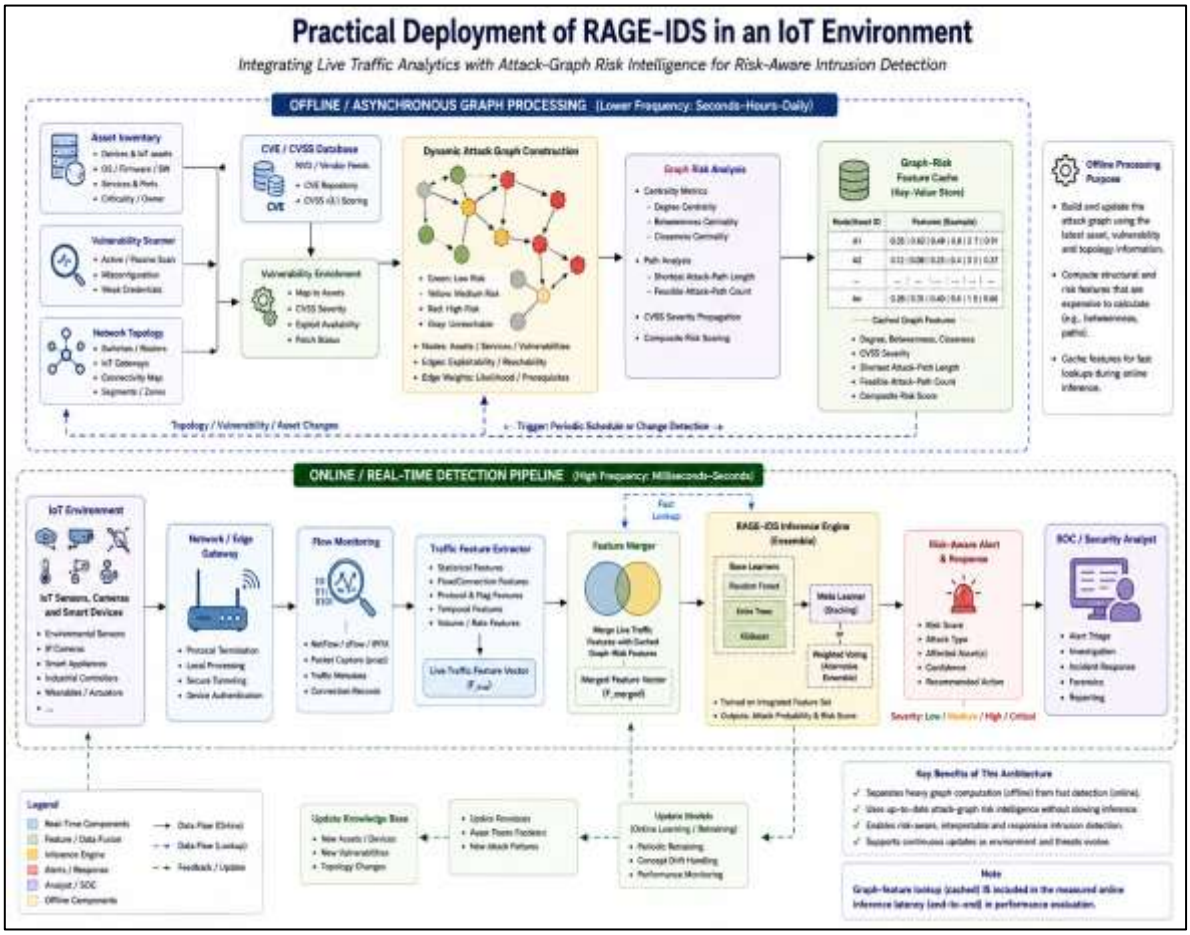
Feature-importance analysis tests the central mechanism proposed by the paper. If graph features consistently receive negligible importance and removing them does not reduce performance, the hypothesis that graph context contributes useful information is weakened. Conversely, if variables such as betweenness, CVSS, path length, or path count appear among influential features and ablation confirms performance loss when they are removed, the mechanism becomes more credible. Interpretability should therefore be linked to ablation rather than presented as decorative visualization.




Computational analysis distinguishes offline and online costs. Betweenness and path computation can be expensive on large graphs, but if the topology is updated less frequently than network flows arrive, these values can be cached. The online cost then consists of mapping the observation to a graph entity, retrieving the cached vector, concatenating features, and executing the ensemble. The final deployment claim should be based on measured p95 latency under the intended hardware and batch size.



From a practical deployment perspective, RAGE-IDS separates computationally intensive graph processing from real-time intrusion detection. Asset inventory, vulnerability information, CVE/CVSS data, and network-topology information are processed asynchronously to construct and periodically update the attack graph. The resulting graph-risk descriptors are cached and retrieved during online detection, where they are mapped to the corresponding live traffic observations and fused with traffic features before ensemble inference. This architecture avoids recomputing graph measures for every incoming flow while allowing the graph-risk representation to be refreshed whenever significant topology or vulnerability-state changes occur.




## 6. Discussion

The theoretical rationale of RAGE-IDS is that network behavior and attack-path context describe different aspects of the same security event. Traffic features characterize observed communication, while attack-graph features characterize the modeled strategic position of the entities involved. A connection to a vulnerable gateway can therefore receive additional context even when its packet statistics resemble benign activity. This may be particularly useful in multi-stage campaigns in which individual actions are intentionally designed to remain below anomaly thresholds.

Betweenness centrality has a natural security interpretation when an asset connects multiple feasible attack routes. Compromise of a high-betweenness node may open access to otherwise separated regions of the graph. Degree centrality indicates local connectivity and can approximate immediate exposure, while closeness reflects how rapidly a node can reach other modeled states. These measures are not vulnerabilities by themselves; their value arises when combined with exploitability and observed traffic.

Path length and path count add progression information. A short path from an attacker state to a critical target suggests that fewer modeled transitions remain, whereas a large number of feasible paths indicates redundancy in attacker opportunities. A defender may therefore prioritize an event that occurs on a short, highly connected, high-severity route even when the flow-level anomaly is moderate. This is the intuition behind risk-aware feature fusion.

CVSS contributes standardized vulnerability severity but must be interpreted carefully. The CVSS specification is designed to communicate technical severity, not complete business risk [8]. Environmental context, asset criticality, compensating controls, exploit availability, and threat intelligence can alter

practical risk. RAGE-IDS therefore avoids using CVSS as a standalone decision rule and instead integrates it with graph topology and traffic behavior.

The ensemble layer is similarly motivated by complementarity. Random Forest can provide stable bagged decisions, Extra Trees can diversify split behavior, and XGBoost can focus sequentially on difficult residual patterns. Stacking can learn when each base learner is trustworthy [3]. However, ensemble complexity is justified only if it provides measurable benefit over the best single model. The ablation design prevents the paper from attributing a feature-engineering gain to the ensemble or vice versa.

One important risk is graph-induced leakage. If attack labels are directly converted into graph states in a way that effectively reproduces the target label, the graph features may trivially encode the answer. The mapping protocol must therefore distinguish legitimate contextual information available at detection time from labels known only after annotation. Vulnerability and topology features should ideally come from external configuration or state information, while attack-stage mappings used for analysis should not expose the ground-truth class to the classifier.

Another challenge is graph staleness. IoT networks can change as devices connect, disconnect, receive patches, or change services. Salayma's dynamic attack-graph work highlights the need to model such changes [14]. A practical RAGE-IDS deployment would therefore maintain a graph-refresh process driven by asset inventory, vulnerability scanning, configuration management, and network discovery. Cached features must be invalidated when material topology or vulnerability changes occur.

The cross-dataset design improves methodological breadth but does not automatically establish domain generalization. NSL-KDD, UNSW-NB15, and CIC-IDS-2017 do not share a single raw feature schema, and none provides a complete enterprise attack graph. The claim is therefore that the same graph-enrichment principle can be applied through dataset-specific adapters. A stronger claim would require harmonized features and train-on-one/test-on-another experiments or evaluation on an IoT testbed.

Recent research increasingly explores graph learning and multi-stage prediction [12], [14], [16]. RAGE-IDS occupies a deliberately interpretable position in this landscape. Rather than learning opaque graph embeddings, it exposes centrality, path, and vulnerability features that can be audited by security analysts. This may simplify deployment in environments where explainability, deterministic mapping, and compatibility with existing machine-learning pipelines are important.

If the final experiments support the hypotheses, the practical implication would be that an IDS can improve prioritization by incorporating contextual security state without abandoning conventional tabular models. If the hypotheses are rejected, the study remains informative: it would indicate that the chosen attack-graph descriptors or mapping assumptions do not provide incremental predictive value under the tested conditions, motivating alternative temporal graphs, learned embeddings, or richer asset-context features.

## 7. Threats to Validity

Dataset validity is the first threat. NSL-KDD is derived from legacy traffic and should not be treated as a modern IoT deployment [4]. UNSW-NB15 and CIC-IDS-2017 are more contemporary but were generated in controlled environments [5], [6]. Benchmark performance therefore measures behavior on the selected corpora rather than guaranteed field performance.

Construct validity is dominated by attack-graph mapping. The benchmark datasets do not provide complete ground-truth vulnerability graphs, so graph construction depends on documented assumptions. Every mapping rule must be auditable. Sensitivity analysis should vary plausible topology or vulnerability assignments and test whether conclusions remain stable. If small arbitrary mapping changes reverse the result, the method is not robust.

Internal validity can be threatened by data leakage. Encoding, scaling, oversampling, feature selection, graph-derived transformations that depend on labels, hyperparameter tuning, and stacking must respect the training boundary. Test data cannot influence any learned transformation. The final test partition should remain untouched until model selection is complete.

Metric validity is another concern. Accuracy can be high when dominant classes are easy to classify. Macro F1, per-class recall, FPR, confusion matrices, and support counts are therefore mandatory. AUC must state the multiclass aggregation strategy. Precision-recall analysis is recommended for strongly imbalanced attack classes.

Statistical conclusion validity depends on the unit of replication. Multiple folds or random seeds can provide paired observations, but correlated folds should not be treated as unlimited independent evidence. Exact test procedures, p-values, confidence intervals, and effect sizes must be reported. Multiple-comparison correction is required when many pairwise hypotheses are tested.

External validity remains limited until evaluation on a realistic IoT testbed or operational network is completed. A future deployment study should integrate asset inventory, vulnerability scanning, live flow extraction, graph updates, and streaming inference. It should also measure alert volume, analyst workload, drift, and robustness to topology change.

## 8. Conclusion and Future Work

This paper presents RAGE-IDS, a risk-aware graph-enhanced ensemble framework for multi-stage IoT intrusion detection. The approach addresses a representational limitation of traffic-only IDS by incorporating explicit attack-path and vulnerability context. Degree, betweenness, and closeness centrality, CVSS severity, shortest attack-path length, feasible path count, and composite risk are fused with dataset-specific network features and supplied to complementary tree-based learners combined through weighted voting or stacked generalization.

The research design emphasizes controlled evidence. Traffic-only, graph-only, fused-feature, best-single-model, and full-ensemble configurations are evaluated under the same data boundaries. This separates gains caused by representation from gains caused by classifier combination. The protocol further requires class-wise metrics, false-positive analysis, computational efficiency, explainability, and statistical testing across NSL-KDD, UNSW-NB15, and CIC-IDS-2017.

Future work should investigate temporal and dynamic attack graphs, CVSS v4.0-aware contextual scoring, graph neural networks, learned graph embeddings, federated intrusion detection, concept-drift adaptation, online graph refresh, adversarial robustness, explainable risk calibration, and deployment on a real IoT testbed. A particularly valuable extension is cross-domain transfer using harmonized flow features and independently sourced vulnerability/topology context.

## References

[1] L. Breiman, “Random Forests,” Machine Learning, vol. 45, pp. 5-32, 2001. doi: 10.1023/A:1010933404324.

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

[3] D. H. Wolpert, “Stacked Generalization,” Neural Networks, vol. 5, no. 2, pp. 241-259, 1992. doi: 10.1016/S0893-6080(05)80023-1.

[4] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, “A Detailed Analysis of the KDD CUP 99 Data Set,” in 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications, 2009. doi: 10.1109/CISDA.2009.5356528.

[5] N. Moustafa and J. Slay, “UNSW-NB15: A Comprehensive Data Set for Network Intrusion Detection Systems (UNSW-NB15 Network Data Set),” in 2015 Military Communications and Information Systems Conference (MilCIS), 2015, pp. 1-6. doi: 10.1109/MilCIS.2015.7348942.

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

[7] O. Sheyner, J. W. Haines, S. Jha, R. P. Lippmann, and J. M. Wing, “Automated Generation and Analysis of Attack Graphs,” in Proceedings of the 2002 IEEE Symposium on Security and Privacy, 2002, pp. 273-284. doi: 10.1109/SECPRI.2002.1004377.

[8] FIRST.Org, Inc., “Common Vulnerability Scoring System v3.1: Specification Document,” 2019. Official CVSS specification, accessed for this manuscript in 2026.

[9] R. Alghamdi and M. Bellaiche, “An Ensemble Deep Learning Based IDS for IoT Using Lambda Architecture,” Cybersecurity, vol. 6, article 5, 2023. doi: 10.1186/s42400-022-00133-w.

[10] Q. Abbas, S. Hina, H. Sajjad, K. S. Zaidi, and R. Akbar, “Optimization of Predictive Performance of Intrusion Detection System Using Hybrid Ensemble Model for Secure Systems,” PeerJ Computer Science, vol. 9, e1552, 2023. doi: 10.7717/peerj-cs.1552.

[11] D. Concejal Muñoz and A. del-Corte Valiente, “A Novel Botnet Attack Detection for IoT Networks Based on Communication Graphs,” Cybersecurity, vol. 6, article 33, 2023. doi: 10.1186/s42400-023-00169-6.

[12] H. Lyu, J. Liu, Y. Lai, B. Mao, and X. Huang, “AGCM: A Multi-Stage Attack Correlation and Scenario Reconstruction Method Based on Graph Aggregation,” Computer Communications, vol. 224, pp. 302-313, 2024. doi: 10.1016/j.comcom.2024.06.016.

[13] F. Jemili, K. Jouini, and O. Korbaa, “Detecting Unknown Intrusions from Large Heterogeneous Data Through Ensemble Learning,” Intelligent Systems with Applications, vol. 25, 200465, 2025. doi: 10.1016/j.iswa.2024.200465.

[14] M. Salayma, “Threat Modelling in Internet of Things (IoT) Environments Using Dynamic Attack Graphs,” Frontiers in the Internet of Things, vol. 3, 2024. doi: 10.3389/friot.2024.1306465.

[15] F. S. Alsubaei, “Smart Deep Learning Model for Enhanced IoT Intrusion Detection,” Scientific Reports, vol. 15, article 20577, 2025. doi: 10.1038/s41598-025-06363-5.

[16] H. Friji, I. Mavromatis, A. Sanchez-Mompo, P. Carnelli, A. Olivereau, and A. Khan, "Multi-stage Attack Detection and Prediction Using Graph Neural Networks: An IoT Feasibility Study," arXiv:2404.18328, 2024.
[17] N. Moustafa, J. Hu, and J. Slay, "A Holistic Review of Network Anomaly Detection Systems: A Comprehensive Survey," Journal of Network and Computer Applications, vol. 128, pp. 33-55, 2019. doi: 10.1016/j.jnca.2018.12.006.
[18] N. Moustafa, B. Turnbull, and K.-K. R. Choo, "An Ensemble Intrusion Detection Technique Based on Proposed Statistical Flow Features for Protecting Network Traffic of Internet of Things," IEEE Internet of Things Journal, vol. 6, no. 3, pp. 4815-4830, 2019. doi: 10.1109/JIOT.2018.2871719.
[19] N. Koroniotis, N. Moustafa, E. Sitnikova, and B. Turnbull, "Towards the Development of Realistic Botnet Dataset in the Internet of Things for Network Forensic Analytics: Bot-IoT Dataset," Future Generation Computer Systems, vol. 100, pp. 779-796, 2019. doi: 10.1016/j.future.2019.05.041.
[20] S. García, M. Grill, J. Stiborek, and A. Zunino, "An Empirical Comparison of Botnet Detection Methods," Computers & Security, vol. 45, pp. 100-123, 2014. doi: 10.1016/j.cose.2014.05.011.
[21] T. Fawcett, "An Introduction to ROC Analysis," Pattern Recognition Letters, vol. 27, no. 8, pp. 861-874, 2006. doi: 10.1016/j.patrec.2005.10.010.
[22] M. Sokolova and G. Lapalme, "A Systematic Analysis of Performance Measures for Classification Tasks," Information Processing & Management, vol. 45, no. 4, pp. 427-437, 2009. doi: 10.1016/j.ipm.2009.03.002.
[23] S. M. Lundberg and S.-I. Lee, "A Unified Approach to Interpreting Model Predictions," in Advances in Neural Information Processing Systems 30, 2017, pp. 4765-4774.
[24] J. H. Friedman, "Greedy Function Approximation: A Gradient Boosting Machine," Annals of Statistics, vol. 29, no. 5, pp. 1189-1232, 2001. doi: 10.1214/aos/1013203451.
[25] P. Geurts, D. Ernst, and L. Wehenkel, "Extremely Randomized Trees," Machine Learning, vol. 63, pp. 3-42, 2006. doi: 10.1007/s10994-006-6226-1.