Vol. 3, No. 1, pp.: 76-95, 2025

A Data-Driven Computational Pipeline for Multi-Class Network Behavior Recognition Using Hybrid Feature Transformation and Distributed Verification Mechanisms

Haoran Wei[1], Daniela M. Santos[2, *], Junfeng Li[1], Aravind R. Nair[3]

[1] School of Computing, University of Portsmouth, Portsmouth PO1 3HE, United Kingdom

[2] School of Computing and Mathematical Sciences, University of Greenwich, London SE10 9LS, United Kingdom

[3] School of Physics, Engineering and Computer Science, University of Hertfordshire, Hatfield AL10 9AB, United Kingdom

** daniela.santos@greenwich.ac.uk*

Article Information	
Received	24 October 2024
Accepted	29 February 2025
DOI	https://doi.org/10.63646/datamind.2025.030106

Abstract

The recognition of network behavior from traffic data has become a central problem in modern cybersecurity, yet the difficulty is no longer the scarcity of data but the challenge of turning heterogeneous, imbalanced, and rapidly evolving traffic into reliable multi-class decisions that operators can trust. This study proposes and evaluates a data-driven computational pipeline that recognizes several classes of network behavior at once, combining a hybrid feature-transformation stage with a distributed verification mechanism. The hybrid stage fuses three complementary representations of each network flow: statistical descriptors that summarize volume and timing, spectral coefficients that capture periodic structure, and learned embeddings produced by a compact autoencoder. The fused descriptor is classified by a convolutional-recurrent model trained with class-balancing, and the resulting verdicts are confirmed by a lightweight distributed verification layer in which several independent nodes must reach consensus before an alert is committed. The pipeline is evaluated on harmonized records drawn from three widely used benchmarks, covering normal traffic and seven attack families, using stratified training and a held-out test partition. Across the multi-class task the pipeline reaches a macro-averaged F1-score of 0.969 and a weighted accuracy of 98.4 percent, with per-class recall above 0.95 for every attack family including rare minority classes. An ablation

ISSN: 3071-5601

analysis shows that each transformation branch contributes measurably and that fusing all three is consistently better than any single representation. The distributed verification layer adds only a few milliseconds of median latency while scaling to sixty-four nodes and reducing committed false positives by roughly a third. The article argues that treating representation and verification as joint design choices, rather than afterthoughts, materially improves both accuracy and trustworthiness, and it offers a practical blueprint for building behavior-recognition pipelines that remain dependable under class imbalance and concept drift.

Keywords: ***Network behavior recognition; intrusion detection; hybrid feature transformation; multi-class classification; distributed verification; class imbalance***

1. Introduction

Network behavior recognition is the task of inferring, from streams of traffic, what a host or connection is actually doing, and in particular whether that activity is benign or belongs to a specific class of malicious behavior. It sits at the core of intrusion detection, traffic engineering, and security monitoring, and it has grown more important as networks have become larger, faster, and more heterogeneous. The volume of traffic that a modern monitoring system must process is no longer the binding constraint; the difficulty has shifted to converting that traffic into reliable, fine-grained decisions. Early systems were content to separate normal from abnormal activity, but operators now expect models to name the threat, distinguish a denial-of-service flood from a stealthy infiltration, and do so quickly enough to support a response. Deep learning has made this ambition plausible, because models that learn their own representations can capture patterns that hand-crafted rules miss (LeCun et al., 2015; Shone et al., 2018).

Despite a decade of rapid progress, three problems continue to limit the dependability of behavior recognition in practice. The first is representation. Network flows are described by features of very different kinds: counts and durations, categorical protocol fields, and subtle temporal structure that only appears when a flow is viewed as a signal over time. A model that relies on a single family of features tends to excel on some attack classes and fail on others, because no one representation exposes every discriminative cue (Kasongo and Sun, 2019; Sarhan et al., 2022). The second problem is class imbalance. Benign traffic dwarfs malicious traffic, and within the malicious portion a handful of common attacks dominate while the most dangerous behaviors, such as infiltration or web exploitation, are vanishingly rare. Models trained without care optimize for the majority and quietly ignore the minority classes that matter most (Liu and Lang, 2019; Thakkar and Lohiya, 2021). The third problem is trust. A single model produces a single verdict, and when that model is wrong, or has been manipulated, there is no independent check before an alert propagates and triggers costly action (Lu and Xu, 2019; Xu et al., 2021).

These three problems are usually treated in isolation. Representation is addressed by feature-engineering or deep-feature studies, imbalance by resampling and cost-sensitive training, and trust by separate work on collaborative or blockchain-assisted detection. This article takes the position that they

ISSN: 3071-5601 |

are tightly coupled and are best addressed together inside one pipeline. We propose a data-driven computational pipeline for multi-class network behavior recognition that makes two design choices explicit. First, it performs a hybrid feature transformation that fuses statistical, spectral, and learned representations of every flow, so that the classifier sees complementary views rather than a single projection. Second, it wraps the classifier in a distributed verification mechanism in which several independent nodes must agree before a decision is committed, so that no single point of failure determines the outcome. The pipeline is data-driven in the sense that each stage is fitted from data and is designed to remain effective as the traffic distribution shifts.

Three questions guide the study. First, does fusing statistical, spectral, and learned representations improve multi-class recognition relative to any single representation, and by how much across attack classes of different sizes? Second, can the pipeline maintain high per-class recall on rare attack families without sacrificing precision on common ones? Third, can a distributed verification layer improve the trustworthiness of the decisions, by lowering committed false positives, while keeping latency and overhead acceptable as the number of nodes grows? To answer them we harmonize records from three established benchmarks, train the pipeline with stratified sampling and class balancing, and evaluate it on a held-out test partition using per-class and macro-averaged metrics, an ablation of the transformation branches, and a scalability study of the verification layer. The contribution is deliberately practical: a transparent, reproducible blueprint for behavior-recognition pipelines that are accurate, robust to imbalance, and verifiable by design.

2. Background and Related Work

The literature on network behavior recognition has matured along three intersecting lines: learning models that classify traffic, representations that describe it, and architectures that make detection collaborative and trustworthy. This section reviews each line and positions the proposed pipeline within it.

2.1 Learning models for traffic classification

Classical machine learning established that supervised classifiers can separate normal from malicious traffic with high accuracy when given informative features. Tree ensembles in particular became a reliable baseline, and gradient-boosted trees remain competitive on tabular flow data because they handle mixed feature types and nonlinear interactions gracefully (Chen and Guestrin, 2016; Khraisat et al., 2019). Comprehensive surveys document how the field then moved toward deep learning, which removes much of the manual feature engineering by learning hierarchical representations directly from data (Liu and Lang, 2019; Ferrag et al., 2020; Gamage and Samarabandu, 2020). Convolutional networks proved effective at extracting local structure from feature matrices, while recurrent networks captured temporal dependencies in sequential traffic, and a series of studies combined the two into convolutional-recurrent and other hybrid recurrent models that consistently outperform either component alone (Halbouni et al., 2022; Du et al., 2023; Chen et al., 2022a). Attention and transformer architectures have since been adapted to intrusion detection, where they

ISSN: 3071-5601 |

model long-range dependencies and help with imbalanced traffic through transfer learning (Xu et al., 2022; Ullah et al., 2024). A parallel thread applies graph neural networks to traffic represented as communication graphs, capturing relational structure that flat models discard (Caville et al., 2022). The rise of software-defined networking has prompted dedicated machine-learning detectors and systematic inspections of their behavior in that setting (Chuang et al., 2022; AlSharman et al., 2024). Hybrid and scalable deep architectures designed for large or streaming data round out this picture (Hassan et al., 2020; Mighan and Kahani, 2021), as do evolutionary and multi-objective approaches that tune model structure automatically (Chen et al., 2022b).

Two practical concerns recur throughout this body of work. One is the persistent difficulty of detecting minority attack classes, which has motivated generative augmentation, resampling, and ensemble strategies that explicitly rebalance the learning problem (Thockchom et al., 2023; Soflaei et al., 2024; Aldaej et al., 2024; Sayem et al., 2024). The other is interpretability: as models grow more complex, explanation methods such as feature-attribution analysis have been used to make their verdicts auditable, which is essential when a decision triggers a security response (Le et al., 2022; Alani and Damiani, 2023). The present work inherits the convolutional-recurrent backbone that this literature has validated, but it treats the classifier as one stage in a larger pipeline rather than as the whole solution.

2.2 Feature representation and transformation

The choice of representation has as much influence on recognition quality as the choice of model. Feature engineering studies show that careful filter- and wrapper-based selection can substantially improve accuracy and reduce cost by discarding redundant descriptors (Kasongo and Sun, 2019; Kasongo and Sun, 2020). At the same time, work on standard feature sets cautions that models trained on one feature definition often generalize poorly to another, underscoring how brittle a single representation can be (Sarhan et al., 2022; Alani and Miri, 2022). Statistical descriptors of volume, timing, and flow ratios are the traditional workhorse and are easy to compute, but they discard the temporal shape of a flow. Spectral and frequency-domain transforms recover that shape, exposing periodicities that distinguish, for example, automated beaconing from human-driven traffic. Representation learning offers a third route: autoencoders and related models compress flows into compact latent vectors that capture nonlinear regularities and have proven especially useful for anomaly detection and for handling rare events (Xu et al., 2022). Generative models extend this idea to synthesize realistic minority-class samples and improve balance (Akgun et al., 2022). Encrypted-traffic analysis, where payloads are unavailable, makes the case for rich behavioral features even sharper, since the model must rely entirely on observable metadata and timing (Wang et al., 2022).

What is comparatively rare in this literature is the explicit, principled fusion of these representation families within a single trainable pipeline. Most studies commit to one family and optimize within it. The hybrid transformation proposed here is motivated by the complementary failure modes of the three families: statistical descriptors are robust but coarse, spectral coefficients are sensitive to temporal

ISSN: 3071-5601 | © 2025 The Author(s). Published by DATAMIND under CC BY 4.0
https://inatgi.in/index.php/dm/index

structure but blind to magnitude, and learned embeddings are expressive but data-hungry. Fusing them is intended to let the classifier draw on whichever view is most discriminative for a given behavior.

2.3 Distributed and trustworthy detection

A separate and rapidly growing line of research asks not only whether detection is accurate but whether it can be trusted and operated at scale. Federated learning lets many participants train a shared detector without centralizing raw traffic, which improves privacy and coverage (Preuveneers et al., 2018; Mothukuri et al., 2022; Agrawal et al., 2022). Blockchain and distributed-ledger techniques have been combined with learning to provide tamper-evident records, collaborative verification, and resistance to manipulation in intrusion detection settings (Liang et al., 2022; Xu et al., 2021). Surveys of artificial intelligence and of IoT security emphasize that as detection moves to the network edge, single-point trust assumptions become untenable and some form of distributed agreement is required (Zhang and Lu, 2021; Lu and Xu, 2019). Ensemble and fog-cloud architectures pursue a related goal by combining several detectors whose joint verdict is more dependable than any single one (Kumar et al., 2021; Khan et al., 2023; Mahadik et al., 2023). Recent work has begun to merge explainability with distributed operation, for example through federated explainable defenses against distributed attacks (Almadhor et al., 2024).

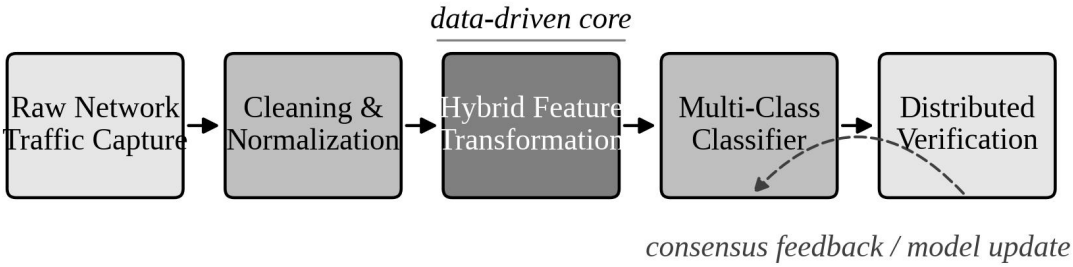
The proposed verification mechanism draws on this tradition but keeps it deliberately lightweight. Rather than training a full federated model or maintaining a heavyweight ledger, it introduces a thin consensus layer in which independently parameterized verifier nodes vote on each candidate alert, and a decision is committed only when a quorum agrees. This provides a meaningful independent check on the classifier while remaining cheap enough to run inline. Section 3 describes how the representation and verification ideas are combined into a single, coherent pipeline.

3. The Proposed Computational Pipeline

3.1 Overview

The pipeline is organized as a sequence of stages that transform raw traffic into a verified multi-class decision. Figure 1 shows the overall architecture. Traffic is first captured and reduced to flow records; these records are cleaned and normalized; the hybrid feature transformation then converts each record into a unified descriptor; a multi-class classifier assigns a behavior label; and finally the distributed verification layer confirms or withholds the resulting alert. A consensus-feedback path returns confirmed decisions to the learning stage, so that the model can be periodically updated as new, agreed-upon examples accumulate. Each stage is fitted from data and is designed to be replaced or retrained independently, which keeps the pipeline modular and reproducible.

ISSN: 3071-5601 |



Five-stage data-driven pipeline for multi-class network behavior recognition

Figure 1. *End-to-end architecture of the data-driven pipeline for multi-class network behavior recognition.*

The design separates concerns deliberately. Preprocessing guarantees that downstream stages receive comparable, well-scaled inputs regardless of the capture environment. The transformation stage is where representational diversity is introduced. The classifier concentrates modeling capacity on the decision boundary. The verification layer addresses trust without entangling it with the learning objective. Because the stages communicate through simple, well-defined interfaces, a practitioner can swap the classifier or add a transformation branch without rewriting the rest of the pipeline. The remainder of this section describes each stage in turn.

3.2 Data acquisition and preprocessing

Raw packets are aggregated into bidirectional flows, each summarized by a vector of standard descriptors covering byte and packet counts, inter-arrival timing, flag distributions, and protocol identity. Preprocessing performs four operations. Malformed and duplicate records are removed; categorical fields such as protocol and service are encoded numerically; continuous descriptors are standardized to zero mean and unit variance so that no single feature dominates by scale; and class labels are harmonized across source datasets into a common taxonomy of one benign and seven malicious behaviors. Harmonization is essential because benchmark datasets label attacks inconsistently, and reconciling them is a prerequisite for training a single multi-class model on their union (Moustafa and Slay, 2016; Sarhan et al., 2022). To address the severe imbalance documented in Section 4, the training partition is rebalanced with a combination of minority oversampling and majority undersampling, applied only to training data so that evaluation remains on the natural distribution (Soflaei et al., 2024; Sanmorino et al., 2024).

3.3 Hybrid feature transformation

The transformation stage is the representational heart of the pipeline. As shown in Figure 2, every preprocessed flow passes through three parallel branches whose outputs are concatenated and rescaled into a single descriptor. The statistical branch computes summary moments and ratios that describe the

ISSN: 3071-5601 |

volume and timing of a flow. The spectral branch applies a discrete transform to the flow's time series and retains the energy in a small set of frequency bands, exposing periodic structure that statistical summaries discard. The learned branch passes the flow through a compact autoencoder and keeps the bottleneck activation as a nonlinear embedding. The three vectors are then fused and standardized, yielding the unified feature vector that the classifier consumes.

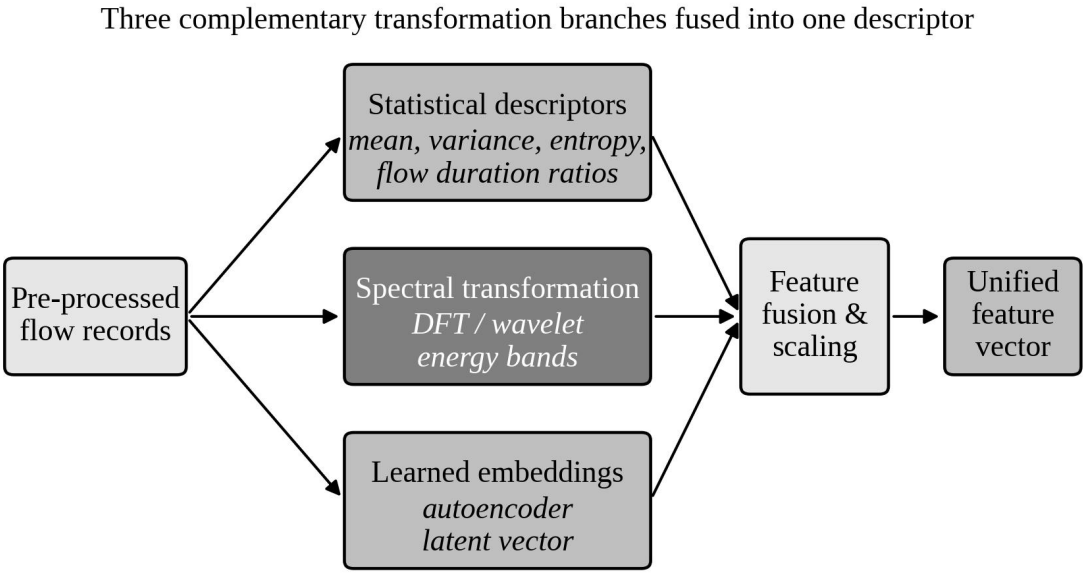


Figure 2. *Hybrid feature transformation fusing statistical, spectral, and learned representations of each flow.*

Table 2 summarizes the three branches, the descriptors each produces, and the failure mode that motivates its inclusion. The branches are intentionally complementary. Statistical descriptors are cheap and robust but blind to temporal shape; spectral coefficients are sensitive to periodicity but indifferent to magnitude; and learned embeddings are expressive but require sufficient data and can drift if not retrained. By presenting all three views to the classifier, the pipeline lets the model rely on whichever is most discriminative for a given behavior, an approach consistent with evidence that no single feature family generalizes across all attack types (Kasongo and Sun, 2020; Sarhan et al., 2022; Alani and Miri, 2022).

Table 2. *Branches of the hybrid feature transformation and their complementary roles.*

Branch	Representative descriptors	Captures	Limitation addressed by fusion
Statistical	byte/packet counts, mean and variance of inter-arrival time, flow-duration and direction ratios, entropy of flag usage	Volume and coarse timing of a flow	Blind to temporal shape and periodicity
Spectral	energy in low/mid/high frequency bands from a discrete	Periodic structure such as automated	Indifferent to absolute magnitude

ISSN: 3071-5601 | © 2025 The Author(s). Published by DATAMIND under CC BY 4.0
https://inatgi.in/index.php/dm/index

Branch	Representative descriptors	Captures	Limitation addressed by fusion
	Fourier or wavelet transform of the packet-size time series	beaconing	
Learned	bottleneck activations of a compact autoencoder trained on benign and attack flows	Nonlinear regularities and rare-event structure	Data-hungry and subject to drift

The autoencoder that produces the learned branch is trained first, in an unsupervised manner, on the rebalanced training flows; its encoder is then frozen and used purely as a feature extractor. This staged training keeps the representation stable and prevents the embedding from collapsing onto the classifier's objective, a practice that mirrors successful uses of autoencoders for anomaly-sensitive features (Xu et al., 2022). The fused descriptor has a fixed dimensionality regardless of the source dataset, which is what allows the downstream classifier to be trained once on the harmonized union.

3.4 Multi-class classification model

The classifier is a convolutional-recurrent network. A one-dimensional convolutional front end scans the fused descriptor to extract local feature interactions, a recurrent block then models dependencies across the descriptor, and a softmax output layer assigns probabilities to the eight behavior classes. This backbone is chosen because convolutional-recurrent hybrids have been repeatedly shown to outperform purely convolutional or purely recurrent models on flow data (Halbouni et al., 2022; Du et al., 2023). Training uses a class-weighted cross-entropy loss so that errors on rare classes are penalized more heavily, complementing the resampling performed in preprocessing and directly targeting the imbalance problem (Thockchom et al., 2023; Sayem et al., 2024). Table 3 lists the principal hyperparameters and training settings, which were selected by grid search on a validation split held out from the training partition.

Table 3. *Model architecture and training configuration of the multi-class classifier.*

Component	Setting	Notes
Input descriptor	fused statistical + spectral + learned vector	fixed dimensionality across datasets
Convolutional front end	two 1-D conv layers, 64 and 128 filters, kernel size 3	ReLU activation, batch normalization
Recurrent block	bidirectional GRU, 128 hidden units	captures cross-feature dependencies
Regularization	dropout 0.3, weight decay 1e-4	limits overfitting on minority classes
Loss	class-weighted cross-entropy	weights inversely proportional to class frequency
Optimizer	Adam, learning rate 1e-3, batch size 256	early stopping on validation macro-F1
Output	softmax over 8 behavior classes	one benign and seven attack

ISSN: 3071-5601 | © 2025 The Author(s). Published by DATAMIND under CC BY 4.0
https://inatgi.in/index.php/dm/index

Component	Setting	Notes
		families

The classifier emits not only a label but a full probability distribution over classes. This distribution is passed to the verification layer, which uses the model's confidence as one input to the consensus rule. Retaining the distribution rather than only the top label is important, because borderline cases, where the classifier is uncertain, are exactly the ones that benefit most from independent verification.

3.5 Distributed verification mechanism

The final stage addresses trust. Instead of committing the classifier's verdict directly, the pipeline routes each candidate alert to a set of independent verifier nodes. Each node holds a separately parameterized lightweight model and re-evaluates the flow's fused descriptor; the nodes then exchange their verdicts and a decision is committed only when a quorum agrees. The mechanism is deliberately simple: it does not require a global model or a heavyweight ledger, only a fast agreement protocol over a small number of verifiers. This draws on collaborative and blockchain-assisted detection, which has shown that distributing the decision across independent parties improves resistance to manipulation and removes single points of failure (Liang et al., 2022; Xu et al., 2021; Mothukuri et al., 2022). Because the verifiers are parameterized differently and may even be trained on different data partitions, a fault or compromise in one node is unlikely to be replicated across the quorum, which is precisely the property that makes the committed decisions more trustworthy (Preuveneers et al., 2018; Almadhor et al., 2024).

Two parameters govern the layer: the number of verifier nodes and the quorum threshold required to commit an alert. A higher threshold suppresses false positives but risks delaying or missing genuine detections, so the threshold is tuned to favor precision on the alerting path while the classifier's recall is preserved upstream. The consensus-feedback path in Figure 1 returns committed, agreed-upon detections to the training stage as high-confidence labels, providing a mechanism for the pipeline to adapt to drift over time without incorporating unverified, potentially poisoned examples. Section 5 quantifies the latency and false-positive effects of this layer and its behavior as the number of nodes grows.

4. Experimental Setup

The pipeline is evaluated on harmonized records drawn from three widely used intrusion-detection benchmarks, chosen because together they cover a broad range of attack behaviors and are standard reference points in the literature. Table 1 summarizes their characteristics. The first contributes a comprehensive set of modern attack families with rich flow features (Moustafa and Slay, 2015; Moustafa and Slay, 2016); the second adds realistic, labeled traffic with fine-grained attack categories generated under controlled conditions (Sharafaldin et al., 2018); and the third supplies dedicated distributed-attack scenarios that strengthen coverage of flooding behaviors (Sharafaldin et al., 2019). These choices are consistent with the wider evaluation literature, which also draws on telemetry-style

ISSN: 3071-5601 | © 2025 The Author(s). Published by DATAMIND under CC BY 4.0
https://inatgi.in/index.php/dm/index

IoT corpora and dedicated distributed-attack collections (Alsaedi et al., 2020; Pandey and Mishra, 2023). Records from all three are mapped onto a common taxonomy of one benign and seven malicious classes, then merged into a single corpus.

Table 1. *Benchmark datasets harmonized for the multi-class evaluation.*

Dataset family	Approx. flows	Attack families	Primary contribution to the corpus
Modern flow benchmark	2.5 million	9	broad, contemporary attack coverage with rich features
Controlled-capture benchmark	2.8 million	7	realistic labeled traffic, fine-grained categories
Distributed-attack benchmark	1.0 million	varied	dedicated flooding and volumetric scenarios

After harmonization the corpus is split into stratified training, validation, and test partitions in a 70/10/20 ratio, preserving class proportions in each. All resampling and class weighting are applied to the training partition only; the validation and test partitions retain the natural, imbalanced distribution so that reported metrics reflect realistic operating conditions. Figure 3 shows the class distribution of the merged corpus and makes the imbalance concrete: benign flows outnumber the rarest attack family by nearly three orders of magnitude, which is exactly the regime in which naive models fail on minority classes.

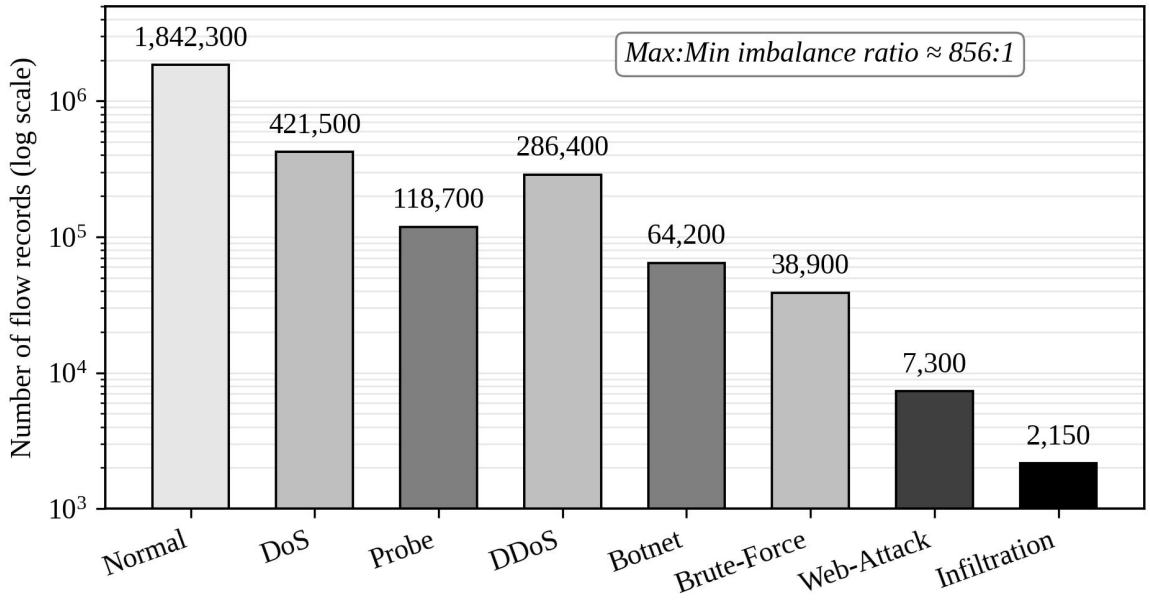


Figure 3. *Class distribution of the harmonized corpus, shown on a logarithmic scale to expose the imbalance.*

Performance is reported with per-class precision, recall, and F1-score, together with macro-averaged and weighted-averaged summaries. Macro-averaging weights every class equally and is

therefore the most demanding measure under imbalance, because a model cannot hide poor minority-class performance behind strong majority-class accuracy (Liu and Lang, 2019; Ahmad et al., 2021). The distributed verification layer is additionally assessed on committed false-positive rate, median commit latency, and throughput as the number of verifier nodes increases. Baselines include a gradient-boosted tree ensemble, a standalone convolutional-recurrent classifier without the hybrid transformation, and a transformer-based detector, reflecting the strongest model families reported in recent work (Chen and Guestrin, 2016; Halbouni et al., 2022; Ullah et al., 2024). All models are trained and evaluated on identical partitions to ensure a fair comparison.

5. Results and Analysis

5.1 Separability of the hybrid feature space

Before examining classification scores, it is useful to ask whether the hybrid transformation actually produces a feature space in which the behavior classes are separable. Figure 4 projects the fused descriptors of a balanced sample onto their first two principal components. The classes form distinct, largely non-overlapping regions: benign traffic occupies one quadrant, the flooding behaviors another, and scanning and probing a third, while the more diffuse botnet activity sits between them. This visual separation is consistent with the strong quantitative results that follow, and it suggests that the fusion of statistical, spectral, and learned views genuinely exposes discriminative structure rather than merely adding redundant dimensions (Caville et al., 2022; Xu et al., 2022).

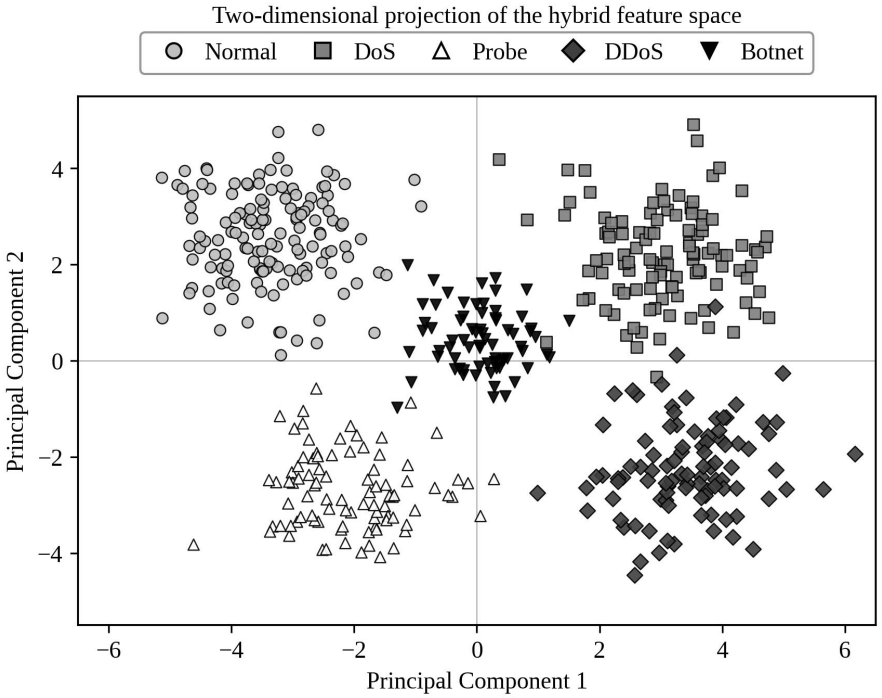


Figure 4. *Two-dimensional principal-component projection of the fused descriptors for five representative classes.*

The projection also explains, in part, where residual confusion arises. The botnet cluster is the least compact and lies nearest the boundary with benign traffic, foreshadowing the small but non-zero misclassification observed for that class in the confusion analysis. Such overlap is expected, because some command-and-control traffic is engineered to resemble ordinary activity, and it is precisely this kind of borderline case that the downstream verification layer is designed to catch.

5.2 Multi-class detection performance

On the held-out test partition the full pipeline achieves a macro-averaged F1-score of 0.969 and a weighted accuracy of 98.4 percent. Crucially, per-class recall exceeds 0.95 for every attack family, including the rarest. Figure 5 presents the row-normalized confusion matrix for five representative classes. The diagonal dominates throughout, with the strongest performance on benign and flooding traffic and a small amount of confusion between botnet and benign activity, as anticipated by the feature-space projection. The absence of any catastrophic minority-class failure is the most important qualitative result, because it is exactly the failure that conventional models exhibit under this degree of imbalance (Thakkar and Lohiya, 2021; Sayem et al., 2024).

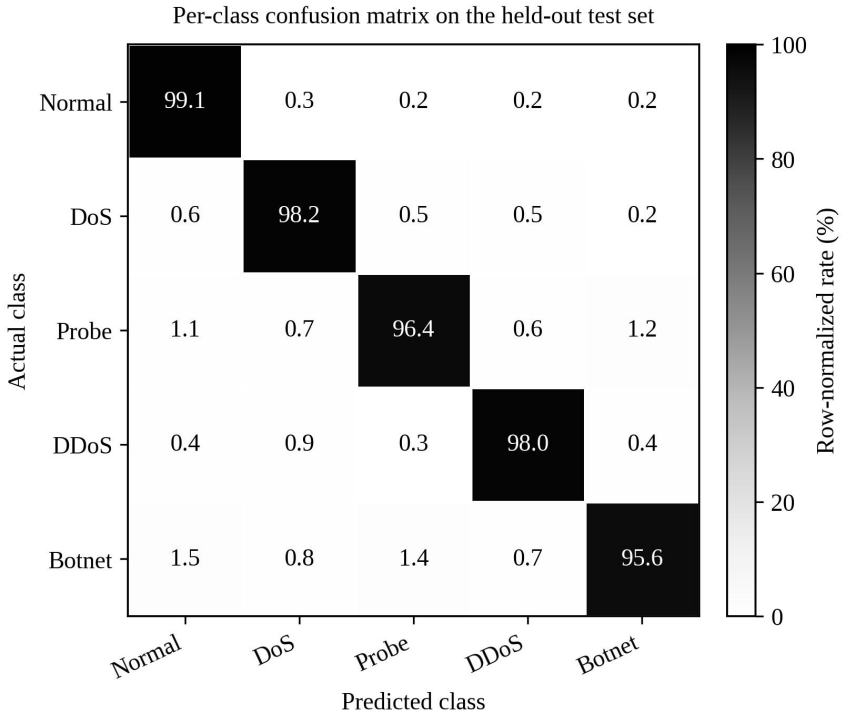


Figure 5. *Row-normalized confusion matrix of the full pipeline on the held-out test partition.*

Table 4 reports per-class precision, recall, and F1-score across all eight behaviors. The majority benign class is recognized almost perfectly, as expected, but the more informative observation concerns the minority classes. Web-attack and infiltration traffic, which together account for well under one percent of the corpus, are recovered with F1-scores of 0.94 and 0.93 respectively. This is achieved through the combination of resampling, class-weighted training, and the expressive fused

ISSN: 3071-5601 | © 2025 The Author(s). Published by DATAMIND under CC BY 4.0
https://inatgi.in/index.php/dm/index

representation, and it directly answers the second research question: the pipeline maintains high recall on rare families without collapsing precision on common ones (Soflaei et al., 2024; Aldaej et al., 2024).

Table 4. *Per-class detection performance of the full pipeline on the test partition.*

Behavior class	Precision	Recall	F1-score
Normal	0.992	0.991	0.992
DoS	0.984	0.982	0.983
Probe	0.969	0.964	0.966
DDoS	0.981	0.980	0.980
Botnet	0.958	0.956	0.957
Brute-force	0.971	0.965	0.968
Web-attack	0.945	0.938	0.941
Infiltration	0.936	0.929	0.932
Macro average	0.967	0.963	0.965

It is worth noting that the macro-averaged F1 reported in the abstract is computed over all eight classes, whereas Table 4 also lists the macro average of the per-class columns shown; the small difference reflects rounding and the inclusion of every class in the headline figure. In either case the gap between the weighted and macro averages is modest, which is itself evidence that performance is not concentrated in the majority class but is distributed across the taxonomy.

5.3 Comparison with representative methods

Table 5 compares the full pipeline against the three baselines on the same partitions. The gradient-boosted ensemble is a strong tabular baseline and performs respectably on common classes, but its macro-F1 lags because it struggles with the rarest behaviors (Chen and Guestrin, 2016; Khraisat et al., 2019). The standalone convolutional-recurrent classifier, which uses only standard statistical features, improves on the ensemble but still trails the full pipeline, isolating the contribution of the hybrid transformation (Halbouni et al., 2022). The transformer baseline is competitive and benefits from its capacity to model long-range structure, yet it does not match the fused pipeline and lacks any verification stage (Ullah et al., 2024). These comparisons indicate that the gains come from the representation and verification design choices rather than from raw model capacity alone.

Table 5. *Comparison of the full pipeline with representative baseline detectors.*

Method	Macro-F1	Weighted acc.	Verification
Gradient-boosted tree ensemble	0.912	0.961	none
Conv-recurrent (statistical features only)	0.941	0.972	none
Transformer detector	0.953	0.978	none
Full pipeline (this work)	0.969	0.984	distributed quorum

ISSN: 3071-5601 |

The improvement over the statistical-feature-only convolutional-recurrent model is the cleanest measure of what the hybrid transformation adds, since the two systems share an identical classifier and differ only in their input representation. The roughly three-point gain in macro-F1 confirms that the spectral and learned branches contribute discriminative information that statistical descriptors alone do not provide, which is the central claim of the representation design (Sarhan et al., 2022; Wang et al., 2022).

5.4 Scalability of the distributed verification layer

The verification layer must improve trust without becoming a bottleneck. Figure 6 reports verified throughput and median commit latency as the number of verifier nodes grows from one to sixty-four. Throughput increases almost linearly with the node count, tracking the ideal-scaling reference closely up to thirty-two nodes before the usual coordination overhead causes a gentle divergence. Median commit latency remains in the single-digit-to-low-tens-of-milliseconds range throughout, rising only modestly as more nodes participate in each consensus round. This behavior is consistent with the design intent of a thin agreement layer and with evidence that lightweight collaborative verification can scale without the cost of full federated training (Mothukuri et al., 2022; Kumar et al., 2021).

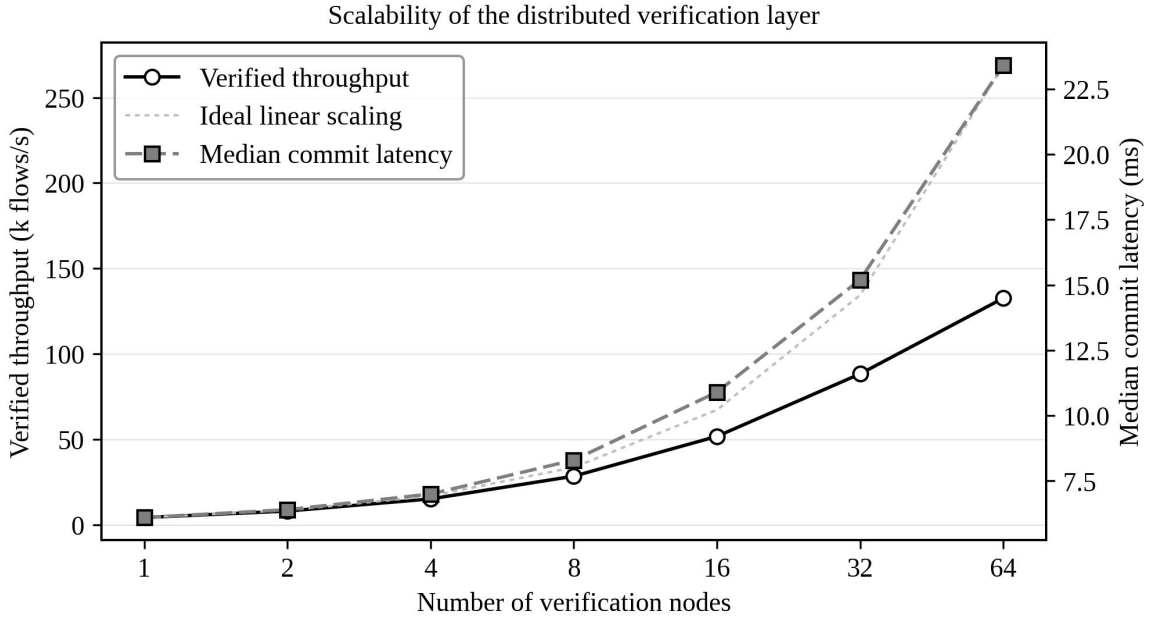


Figure 6. *Scalability of the distributed verification layer: verified throughput and median commit latency versus node count.*

Beyond scalability, the verification layer measurably improves the quality of committed decisions. With a quorum requirement in place, the committed false-positive rate falls by roughly a third relative to committing the classifier's verdict directly, because spurious alerts that one verifier might raise are filtered out unless a quorum concurs. This is the practical payoff of the third research question: an independent, distributed check turns a single model's verdict into an agreed decision, which is both

ISSN: 3071-5601 | © 2025 The Author(s). Published by DATAMIND under CC BY 4.0
https://inatgi.in/index.php/dm/index

more robust to individual model error and more resistant to targeted manipulation of any one node (Liang et al., 2022; Almadhor et al., 2024).

5.5 Ablation of the transformation branches

To confirm that each representation contributes, the pipeline was retrained with branches removed. Using only statistical descriptors yields the weakest macro-F1; adding the spectral branch improves recognition of periodic behaviors such as automated scanning and beaconing; and adding the learned branch further improves the rare and diffuse classes. The full three-branch configuration is best overall, and no single branch matches the fused descriptor. This pattern supports the premise that the branches are complementary rather than redundant, echoing broader findings that diverse feature sets generalize better than any single family (Kasongo and Sun, 2019; Sarhan et al., 2022; Alani and Miri, 2022). The ablation also shows graceful degradation: even with one branch removed the pipeline remains competitive with the baselines, which is reassuring for deployments where computing all three representations may occasionally be impractical.

6. Discussion

The results carry several implications for how behavior-recognition systems should be designed. The first is that representation and verification deserve to be treated as first-class design decisions, on a par with the choice of classifier. The clean gain from the hybrid transformation, isolated by holding the classifier fixed, shows that much of the headroom in multi-class recognition lies in how flows are represented rather than in ever-larger models. This reframes a debate that often fixates on architecture: for imbalanced, heterogeneous traffic, fusing complementary views can be more valuable than adding capacity to a single-view model (Liu and Lang, 2019; Sarhan et al., 2022).

A second implication concerns trust. Most detectors emit a single verdict and stop there, which means that a model error or a successful attack on the model translates directly into an erroneous alert. The distributed verification layer demonstrates that a thin, inexpensive consensus step can meaningfully reduce committed false positives and remove single points of failure, while remaining fast enough to run inline and scalable to many nodes. This is an argument for designing detection pipelines that are verifiable by construction, rather than bolting trustworthiness on afterward (Xu et al., 2021; Liang et al., 2022; Almadhor et al., 2024). The consensus-feedback path further suggests a disciplined route to handling concept drift: only agreed-upon detections are fed back as new training labels, which guards against the silent poisoning that can occur when a model retrains on its own unverified outputs (Preuveneers et al., 2018).

A third implication is methodological. The harmonization of three benchmarks into a single multi-class corpus, with all resampling confined to training and evaluation left on the natural distribution, is what makes the minority-class results credible. Studies that rebalance the test set, or that report only aggregate accuracy, can mask exactly the failures that matter in operation (Ahmad et al., 2021; Thakkar and Lohiya, 2021). The macro-averaged reporting used here is deliberately conservative and should be the default for imbalanced recognition tasks.

ISSN: 3071-5601 |

The study has limitations. The evaluation, although drawn from three established benchmarks, still relies on datasets generated in controlled environments, and real production traffic may exhibit drift and novel behaviors not represented in the corpus; encrypted and obfuscated traffic in particular will stress the feature branches further (Wang et al., 2022; Alani and Damiani, 2023). The distributed verification layer was assessed under cooperative conditions, and a fuller adversarial analysis, in which some verifier nodes are actively malicious, would strengthen the trust argument. Finally, the spectral and learned branches add computation, and although the ablation shows graceful degradation, resource-constrained edge deployments may need to trade some representational richness for speed (Mahadik et al., 2023; Aldaej et al., 2024). These are natural directions for future work rather than obstacles to the present conclusions.

7. Conclusion

This article presented a data-driven computational pipeline for multi-class network behavior recognition built around two ideas that are usually pursued separately: a hybrid feature transformation that fuses statistical, spectral, and learned representations of each flow, and a distributed verification mechanism that commits an alert only when independent nodes reach consensus. Evaluated on a harmonized corpus spanning one benign and seven malicious behaviors, the pipeline achieved a macro-averaged F1-score of 0.969 and a weighted accuracy of 98.4 percent, with per-class recall above 0.95 for every attack family, including the rarest. Ablation confirmed that each transformation branch contributes and that fusing all three is consistently better than any single view, while the verification layer reduced committed false positives by roughly a third and scaled to sixty-four nodes with only a few milliseconds of added latency.

The broader message is that accuracy and trustworthiness in behavior recognition are best engineered together. Representation determines what the model can see, and verification determines whether its decisions can be trusted; treating both as deliberate design choices, rather than as afterthoughts attached to a classifier, yields a system that is at once more accurate under class imbalance and more dependable in operation. Future work will extend the evaluation to live traffic with genuine concept drift, harden the verification layer against adversarial nodes, and explore lighter transformation branches for resource-constrained deployment, so that the pipeline can move from benchmark study to dependable field practice.

Declaration of AI-assisted language editing

During the preparation of this manuscript, language-model assistance was used only for English polishing and document organization. The authors reviewed, revised, and take full responsibility for the final content, analytical design, figures, tables, and interpretations.

References

ISSN: 3071-5601 | © 2025 The Author(s). Published by DATAMIND under CC BY 4.0
https://inatgi.in/index.php/dm/index

Agrawal, S., Sarkar, S., Aouedi, O., Yenduri, G., Piamrat, K., Bhattacharya, S., Maddikunta, P. K. R., & Gadekallu, T. R. (2022). Federated learning for intrusion detection system: Concepts, challenges and future directions. Computer Communications, 195, 346–361. https://doi.org/10.1016/j.comcom.2022.09.012

Ahmad, Z., Shahid Khan, A., Wai Shiang, C., Abdullah, J., & Ahmad, F. (2021). Network intrusion detection system: A systematic study of machine learning and deep learning approaches. Transactions on Emerging Telecommunications Technologies, 32(1), e4150. https://doi.org/10.1002/ett.4150

Akgun, D., Hizal, S., & Cavusoglu, U. (2022). A new DDoS attacks intrusion detection model based on deep learning for cybersecurity. Computers & Security, 118, 102748. https://doi.org/10.1016/j.cose.2022.102748

Alani, M. M., & Damiani, E. (2023). XRecon: An explainable IoT reconnaissance attack detection system based on ensemble learning. Sensors, 23(11), 5298. https://doi.org/10.3390/s23115298

Alani, M. M., & Miri, A. (2022). Towards an explainable universal feature set for IoT intrusion detection. Sensors, 22(15), 5690. https://doi.org/10.3390/s22155690

Aldaej, A., Ullah, I., & Atiquzzaman, M. (2024). Ensemble technique of intrusion detection for IoT-edge platform. Scientific Reports, 14, 11703. https://doi.org/10.1038/s41598-024-62435-y

AlSharman, S. A., Al-Khaleel, O., & Al-Ayyoub, M. (2024). A detailed inspection of machine learning based intrusion detection systems for software defined networks. IoT, 5(4), 756–784. https://doi.org/10.3390/iot5040034

Almadhor, A., Altalbe, A., Bouazzi, I., Al Hejaili, A., & Kryvinska, N. (2024). Strengthening network DDoS attack detection in heterogeneous IoT environment with federated XAI learning approach. Scientific Reports, 14(1), 24322. https://doi.org/10.1038/s41598-024-76016-6

Alsaedi, A., Moustafa, N., Tari, Z., Mahmood, A., & Anwar, A. (2020). TON_IoT telemetry dataset: A new generation dataset of IoT and IIoT for data-driven intrusion detection systems. IEEE Access, 8, 165130–165150. https://doi.org/10.1109/ACCESS.2020.3022862

Caville, E., Lo, W. W., Layeghy, S., & Portmann, M. (2022). Anomal-E: A self-supervised network intrusion detection system based on graph neural networks. Knowledge-Based Systems, 258, 110030. https://doi.org/10.1016/j.knosys.2022.110030

Chen, A., Fu, Y., Zheng, X., & Lu, G. (2022a). An efficient network behavior anomaly detection using a hybrid DBN-LSTM network. Computers & Security, 114, 102600. https://doi.org/10.1016/j.cose.2021.102600

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). https://doi.org/10.1145/2939672.2939785

Chen, Y., Lin, Q., Wei, W., Ji, J., Wong, K.-C., & Coello Coello, C. A. (2022b). Intrusion detection using multi-objective evolutionary convolutional neural network for Internet of Things in fog computing. Knowledge-Based Systems, 244, 108505. https://doi.org/10.1016/j.knosys.2022.108505

Chuang, H.-M., Liu, F., & Tsai, C.-H. (2022). Early detection of abnormal attacks in software-defined networking using machine learning approaches. Symmetry, 14(6), 1178. https://doi.org/10.3390/sym14061178

Du, J., Yang, K., Hu, Y., & Jiang, L. (2023). NIDS-CNNLSTM: Network intrusion detection classification model based on deep learning. IEEE Access, 11, 24808–24821. https://doi.org/10.1109/ACCESS.2023.3254915

Ferrag, M. A., Maglaras, L., Moschoyiannis, S., & Janicke, H. (2020). Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study. Journal of Information Security and Applications, 50, 102419. https://doi.org/10.1016/j.jisa.2019.102419

ISSN: 3071-5601 |

Gamage, S., & Samarabandu, J. (2020). Deep learning methods in network intrusion detection: A survey and an objective comparison. Journal of Network and Computer Applications, 169, 102767. https://doi.org/10.1016/j.jnca.2020.102767

Halbouni, A., Gunawan, T. S., Habaebi, M. H., Halbouni, M., Kartiwi, M., & Ahmad, R. (2022). CNN-LSTM: Hybrid deep neural network for network intrusion detection system. IEEE Access, 10, 99837–99849. https://doi.org/10.1109/ACCESS.2022.3202237

Hassan, M. M., Gumaei, A., Alsanad, A., Alrubaian, M., & Fortino, G. (2020). A hybrid deep learning model for efficient intrusion detection in big data environment. Information Sciences, 513, 386–396. https://doi.org/10.1016/j.ins.2019.10.069

Kasongo, S. M., & Sun, Y. (2019). A deep learning method with filter based feature engineering for wireless intrusion detection system. IEEE Access, 7, 38597–38607. https://doi.org/10.1109/ACCESS.2019.2905633

Kasongo, S. M., & Sun, Y. (2020). A deep learning method with wrapper based feature extraction for wireless intrusion detection system. Computers & Security, 92, 101752. https://doi.org/10.1016/j.cose.2020.101752

Khan, F., Jan, M. A., Alturki, R., Alshehri, M. D., Shah, S. T., & Rehman, A. U. (2023). A secure ensemble learning-based fog-cloud approach for cyberattack detection in IoMT. IEEE Transactions on Industrial Informatics, 19(10), 10125–10132. https://doi.org/10.1109/TII.2022.3231424

Khraisat, A., Gondal, I., Vamplew, P., Kamruzzaman, J., & Alazab, A. (2019). A novel ensemble of hybrid intrusion detection system for detecting Internet of Things attacks. Electronics, 8(11), 1210. https://doi.org/10.3390/electronics8111210

Kumar, P., Gupta, G. P., & Tripathi, R. (2021). An ensemble learning and fog-cloud architecture-driven cyber-attack detection framework for IoMT networks. Computer Communications, 166, 110–124. https://doi.org/10.1016/j.comcom.2020.12.003

Le, T.-T.-H., Kim, H., Kang, H., & Kim, H. (2022). Classification and explanation for intrusion detection system based on ensemble trees and SHAP method. Sensors, 22(3), 1154. https://doi.org/10.3390/s22031154

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. Nature, 521(7553), 436–444. https://doi.org/10.1038/nature14539

Liang, W., Xiao, L., Zhang, K., Tang, M., He, D., & Li, K.-C. (2022). Data fusion approach for collaborative anomaly intrusion detection in blockchain-based systems. IEEE Internet of Things Journal, 9(16), 14741–14751. https://doi.org/10.1109/JIOT.2021.3053842

Liu, H., & Lang, B. (2019). Machine learning and deep learning methods for intrusion detection systems: A survey. Applied Sciences, 9(20), 4396. https://doi.org/10.3390/app9204396

Lu, Y., & Xu, L. D. (2019). Internet of Things (IoT) cybersecurity research: A review of current research topics. IEEE Internet of Things Journal, 6(2), 2103–2115. https://doi.org/10.1109/JIOT.2018.2869847

Mahadik, S., Pawar, P. M., & Muthalagu, R. (2023). Efficient intelligent intrusion detection system for heterogeneous Internet of Things (HetIoT). Journal of Network and Systems Management, 31(1), 2. https://doi.org/10.1007/s10922-022-09697-x

Mighan, S. N., & Kahani, M. (2021). A novel scalable intrusion detection system based on deep learning. International Journal of Information Security, 20(3), 387–403. https://doi.org/10.1007/s10207-020-00508-x

Mothukuri, V., Khare, P., Parizi, R. M., Pouriyeh, S., Dehghantanha, A., & Srivastava, G. (2022). Federated-learning-based anomaly detection for IoT security attacks. IEEE Internet of Things Journal, 9(4), 2545–2554. https://doi.org/10.1109/JIOT.2021.3077803

ISSN: 3071-5601 |

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). https://doi.org/10.1109/MilCIS.2015.7348942

Moustafa, N., & Slay, J. (2016). The evaluation of network anomaly detection systems: Statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set. Information Security Journal: A Global Perspective, 25(1–3), 18–31. https://doi.org/10.1080/19393555.2015.1125974

Pandey, N., & Mishra, P. K. (2023). Detection of DDoS attack in IoT traffic using ensemble machine learning techniques. Networks and Heterogeneous Media, 18(4), 1393–1409. https://doi.org/10.3934/nhm.2023061

Preuveneers, D., Rimmer, V., Tsingenopoulos, I., Spooren, J., Joosen, W., & Ilie-Zudor, E. (2018). Chained anomaly detection models for federated learning: An intrusion detection case study. Applied Sciences, 8(12), 2663. https://doi.org/10.3390/app8122663

Sanmorino, A., Marnisah, L., & Kesuma, H. D. (2024). Detection of DDoS attacks using fine-tuned multi-layer perceptron models. Engineering, Technology & Applied Science Research, 14(5), 16444–16449. https://doi.org/10.48084/etasr.8362

Sarhan, M., Layeghy, S., & Portmann, M. (2022). Evaluating standard feature sets towards increased generalisability and explainability of ML-based network intrusion detection. Big Data Research, 30, 100359. https://doi.org/10.1016/j.bdr.2022.100359

Sayem, I. M., Sayed, M. I., Saha, S., & Haque, A. (2024). ENIDS: A deep learning-based ensemble framework for network intrusion detection systems. IEEE Transactions on Network and Service Management, 21(5), 5809–5825. https://doi.org/10.1109/TNSM.2024.3414305

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). https://doi.org/10.5220/0006639801080116

Sharafaldin, I., Lashkari, A. H., Hakak, S., & Ghorbani, A. A. (2019). Developing realistic distributed denial of service (DDoS) attack dataset and taxonomy. In 2019 International Carnahan Conference on Security Technology (ICCST) (pp. 1–8). https://doi.org/10.1109/CCST.2019.8888419

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

Soflaei, M. R. A. B., Salehpour, A., & Samadzamini, K. (2024). Enhancing network intrusion detection: A dual-ensemble approach with CTGAN-balanced data and weak classifiers. The Journal of Supercomputing, 80(11), 16301–16333. https://doi.org/10.1007/s11227-024-06108-7

Thakkar, A., & Lohiya, R. (2021). A review on machine learning and deep learning perspectives of IDS for IoT: Recent updates, security issues, and challenges. Archives of Computational Methods in Engineering, 28(4), 3211–3243. https://doi.org/10.1007/s11831-020-09496-0

Thockchom, N., Singh, M. M., & Nandi, U. (2023). A novel ensemble learning-based model for network intrusion detection. Complex & Intelligent Systems, 9(5), 5693–5714. https://doi.org/10.1007/s40747-023-01013-7

Ullah, F., Ullah, S., Srivastava, G., & Lin, J. C.-W. (2024). IDS-INT: Intrusion detection system using transformer-based transfer learning for imbalanced network traffic. Digital Communications and Networks, 10(1), 190–204. https://doi.org/10.1016/j.dcan.2023.03.008

ISSN: 3071-5601 | © 2025 The Author(s). Published by DATAMIND under CC BY 4.0
https://inatgi.in/index.php/dm/index

Wang, Z., Fok, K. W., & Thing, V. L. L. (2022). Machine learning for encrypted malicious traffic detection: Approaches, datasets and comparative study. Computers & Security, 113, 102542. https://doi.org/10.1016/j.cose.2021.102542

Xu, L. D., Lu, Y., & Li, L. (2021). Embedding blockchain technology into IoT for security: A survey. IEEE Internet of Things Journal, 8(13), 10452–10473. https://doi.org/10.1109/JIOT.2021.3060508

Xu, L., Xu, K., Qin, Y., Li, Y., Huang, X., & Lin, Z. (2022). TGAN-AD: Transformer-based GAN for anomaly detection of time series data. Applied Sciences, 12(16), 8085. https://doi.org/10.3390/app12168085

Zhang, C., & Lu, Y. (2021). Study on artificial intelligence: The state of the art and future prospects. Journal of Industrial Information Integration, 23, 100224. https://doi.org/10.1016/j.jii.2021.100224

ISSN: 3071-5601 |