Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Decision-Level versus Feature-Level Novelty Detection for Open-Set Intrusion Detection: A Cascaded EfficientNet--GBDT and Isolation Forest Framework Evaluated on TON_IoT

Idris Mohammed Kolo

kolos_2006@yahoo.com

Federal University Lokoja

Sunday Eric Adewumi

Federal University Lokoja

Oladipo Francisca

Federal University Lokoja

Taiwo Kolajo

Federal University Lokoja

Research Article

Keywords: Intrusion detection, TON IoT, Open-set recognition, Isolation Forest, EfficientNet, Gradient boosting, Novelty detection, Industrial IoT, Petroleum pipeline monitoring

Posted Date: November 19th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11518650/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Decision-Level versus Feature-Level Novelty Detection for Open-Set Intrusion Detection: A Cascaded EfficientNet–GBDT and Isolation Forest Framework Evaluated on TON_IoT

Idris Mohammed Kolo[1*], Sunday Eric Adewumi[1], Oladipo Francisca[1], Taiwo Kolajo[1]

[1*]Department of Computer Science, Federal University Lokoja, Lokoja, Kogi State, Nigeria.

*Corresponding author(s). E-mail(s): kolos_2006@yahoo.com;
Contributing authors: adewumis@gmail.com;
Francisca.oladipo@fulokoja.edu.ng; Taiwo.kolajo@fulokoja.edu.ng;

Abstract

Closed-set classifiers on Internet of Things (IoT) and Industrial IoT (IIoT) intrusion benchmarks now reach accuracies at which aggregate accuracy no longer separates competing methods, while the capability that matters most for critical infrastructure, namely sensible behaviour on attacks never labelled during training, remains weakly addressed. This study proposes and evaluates a cascaded open-set intrusion detector on the network component of the Telemetry, Operating System and Network IoT (TON_IoT) dataset. Flow records are reduced by Principal Component Analysis (PCA) and Electromagnetic Field Optimisation (EFO), a compact EfficientNet extracts a learned embedding, a Gradient-Boosted Decision Tree (GBDT) ensemble classifies that embedding, and flows whose maximum GBDT class probability falls below 0.95 are forwarded to an Isolation Forest for novelty detection. Injection and Man-in-the-Middle (MITM) traffic are withheld from training to measure open-set behaviour. The study makes four contributions: (i) a controlled comparison of attaching the Isolation Forest to the reduced input features (decision-level) or to the learned embedding (feature-level) under identical preprocessing, data partitioning and withheld-attack protocol; (ii) evidence that replacing the softmax head with a GBDT head leaves closed-set predictions unchanged; (iii) operating points reported as alarm precision at stated attack base rates rather than detection rate alone; and (iv) measured training, storage, latency, memory and energy costs. On the eight known classes,

the GBDT head attains 98.26% accuracy (macro-F1 0.9789) on 25,421 test flows, identical flow for flow to the EfficientNet softmax baseline. On withheld attacks, the classifier alone flags none as unknown, decision-level attachment identifies 81.4% and feature-level attachment 76.1%, indicating that supervised training compresses away the variation that makes unseen traffic unusual. Withheld-attack capture rises from 76.1% at a 0.65% benign false-alarm rate (54.2% alarm precision at a 1% attack base rate) to 99.8% at 13.9%, where alarm precision falls to 6.8%. The cascade costs 4.7 times the inference time of the feature extractor alone and occupies 91.4 MB. These findings show that the placement of the novelty detector, not the classification head, determines open-set capability, with direct implications for intrusion detection at petroleum-pipeline monitoring gateways.

Keywords: Intrusion detection, TON_IoT, Open-set recognition, Isolation Forest, EfficientNet, Gradient boosting, Novelty detection, Industrial IoT, Petroleum pipeline monitoring

1 Introduction

A recent study on the Telemetry, Operating System and Network Internet of Things (TON_IoT) dataset on the network component demonstrates that lightweight decision-tree and boosting models already can reach performance in the high nineties. Compact classical models have also been quantised and successfully used on actual microcontrollers with minimal impact on the prediction accuracy (Ismail et al 2025; Dharini et al 2026; De Bernardis et al 2026). Given this context, it is difficult to justify a new architecture for the benchmark based on a small increase in F1-score. It should be the case that its value is due to the fact that it overcomes some deficiency of the conventional closed-set classifiers.

In this study, the problem under study is related to decision behaviour under attacks that are not previously observed. Conventional multiclass classifier makes an assumption that all input belong to one of the classes that were presented at the training phase. It then has to classify each observation to a learned class, even if the observation is not in the learned label space (Scheirer et al 2013). In critical-infrastructure monitoring, such an assumption is hard to justify as there may be attacks that are not seen and/or new attacks that emerge that are more important than attacks represented in the training data. This problem is not completely solved by using the maximum softmax probability as a rejection criterion. On one hand, deep neural networks can make high confidence predictions on inputs sampled from unseen distributions (Hendrycks and Gimpel 2017), and on the other hand are prone to overconfidence on in-distribution inputs (Guo et al 2017). Therefore, a high predictive confidence does not mean that the input is familiar to the model.

A feasible solution is to cascade a closed-set classifier with a novelty detector. The classifier classifies traffic in the learned decision space and uncertain observations are passed on to a second stage for further evaluation. Because of its relatively low computational cost and the fact that it is able to identify unusual observations without

performing any distance computation or density estimation, this is an ideal role for which isolation forest is suited in resource-limited environment (Liu et al 2008, 2012).

What the novelty detector should look at is the only issue remaining. One approach is to use the same set of input features as the reduced input to the classifier, which is termed decision-level attachment because both the detector and the classifier share the same reduced input feature set and interact only at the level of the routing decision. The second alternative is to use the novelty detector on the learned feature representation generated in the classifier. This is what we call feature level attachment. The idea is that a detector in the original feature space is unaware of which dimensions contain the most relevant information to the classification problem, while a learned embedding might focus on information that is more relevant to distinguishing security-related patterns. This argument aligns with representation-learning based anomaly detection (Ruff et al 2018). This study does not assume that the more complicated design is better, but instead tests both designs under the same experimental conditions and compares their withheld attack class detection capabilities.

Operational usefulness provides a second motivation for the evaluation. In real monitoring scenarios, the amount of benign traffic is typically much higher than malicious traffic. Thus, even a detector with a low false-positive rate will result in an alarm stream with a high number of false alarms if the prevalence of the attack is low (Axelsson 2000). These threshold-free measurements do not show this issue and therefore do not provide a complete picture of the practical detector performance. This concern is particularly relevant in network anomaly detection, where the evaluation results might not directly correspond to actual deployment actions (Sommer and Paxson 2010; Arp et al 2022). Therefore, the operating points of this study are presented in terms of the precision of the alarm at explicitly specified attack base rates.

The contributions of this study are four-fold. First, it offers a comparison of decision-level and feature-level attachment in which the same novelty detector, preprocessing procedure, data partitioning, and withheld-attack protocol are used. The results indicate that feature-level attachment achieves lower detection rate for withheld attacks when compared to simple decision-level configuration and reasons of this are discussed. Secondly, the proposed architecture does not lead to better closed-set accuracy (it is identical on all flows tested) when the neural network's softmax classification head is replaced with a gradient-boosted classifier. Third, the study considers detector operating points based on alarm precision at realistic attack base rates, and not on detection rates. Finally, it provides measured computational cost – the training time, memory usage, inference time, storage, and energy consumption. These measurements reveal that the full cascade is about 4.7 times more expensive than using the feature extractor alone.

2 Related Work

2.1 Lightweight detection on TON_IoT

TON_IoT is a heterogeneous cybersecurity testbed with telemetry data, operating-system data and network data for IoT and Industrial IoT (IIoT) systems. It has a network part which is comprised of labelled flow records for normal traffic and nine

attack classes (Moustafa 2021; Alsaedi et al 2020). Comparing the results between IoT intrusion-detection datasets is, however, a challenge as the description of the features, attack categories and data-collection procedures differ significantly across benchmarks (Booij et al 2022).

Recent studies also reveal that TON_IoT has a high efficiency baseline. Ismail et al (2025) demonstrate that lightweight decision-tree and boosting models can compete with more computationally expensive models on TON_IoT, WUSTL-IIoT-2021, and Edge-IIoTset. Dharini et al (2026) also show that an Internet Protocol (IP) address and port identifiers can be stripped off, and that the number of variables in the feature space can be reduced down to five variables that are based on behaviour without significant loss of predictive power. De Bernardis et al (2026) continues this research by quantising trained models to 8-bit integer (INT8) precision and deploying them on real microcontrollers. Such results do not provide much opportunity to justify a deeper architecture based on closed set accuracy alone. The more pertinent question is whether such an architecture can offer capabilities that lightweight models don't easily offer. Hence, this paper concentrates on open-set behaviours, specifically identifying observations in attack classes that were not present at training time.

The EfficientNet was initially introduced as a compound-scaling method, which scales the network depth, width, and resolution of the input images simultaneously, with relatively small parameter counts (Tan and Le 2019). The architecture was designed for computer vision, and in order to apply it to tabular network-flow data, the feature vector needs to be mapped into a representation that is given an imposed spatial structure. This brings in an important modeling assumption as there are no natural spatial relationships between tabular network features like in an image (Sharma et al 2019). It therefore cannot be assumed that this transfer is suitable. Heterogeneous tabular benchmarks demonstrate that tree-based ensembles still achieve good performance and can even surpass deep neural architectures (Grinsztajn et al 2022). This use of EfficientNet therefore needs to be justified on the basis of what it can represent and what it can open-set, not just because of its architectural complexity.

2.2 Isolation Forest, representation and open-set recognition

Isolation Forest works by recursively partitioning the space of features, and measuring the speed with which individual points can be isolated, to detect anomalous observations. Observations with higher anomaly scores are observations separated by relatively few partitions, and are assumed to be less consistent with the dominant data distribution (Liu et al 2008, 2012). The effectiveness of its application on TON_IoT is not consistent. For instance, Zahoor et al (2025) note that One-Class Support Vector Machine (SVM) outperforms Isolation Forest in terms of precision, recall, and accuracy. A possible explanation is the representation that the detector uses. If Isolation Forest is used directly on a heterogeneous feature space, variations in the normal traffic can look as if they are malicious traffic as well. Representation-learning methods tackle this issue by mapping the original inputs into a feature space that is more task-relevant. Deep Support Vector Data Description (Deep SVDD) does just that – it learns a representation where normal observations are confined to a compact region (Ruff et al 2018). This encourages us to ask if there is any advantage to using a learned

representation for Isolation Forest as opposed to using the reduced input features directly. An evaluation of the hypothesis is provided in Section 5.6, and the results do not corroborate the expected benefit of the learned representation.

The general issue is one of open set recognition. Scheirer et al (2013) provide a formalisation of the risk associated with the case where a classifier is called to classify classes that were not part of the training label space. OpenMax overcomes this drawback by attempting to re-calibrate the activations in the penultimate layer and by explicitly assigning an observation to a unknown category (Bendale and Boult 2016). A more straightforward method is to reject on the top softmax probability, an important baseline for open-set detection methods (Hendrycks and Gimpel 2017). Vaze et al (2022) also demonstrate that a well-trained closed-set classifier with an appropriate confidence score can achieve better open-set performance than is sometimes reported in the specialised literature. But this confidence based rejection is contingent on the quality of the probability estimates. Neural classifiers are also known to be poorly calibrated and they can be known to give a confidence level that is not close to the actual probability of being correct (Guo et al 2017). This correspondence can be enhanced using post-hoc calibration techniques, like isotonic regression, without retraining the underlying classifier (Zadrozny and Elkan 2002). The impact of this calibration procedure is investigated in Section 5.4.

2.3 The evaluation validity and research gap

Network anomaly detection is especially hard to assess, as the impact of false positive and false negative rates is very asymmetric, and benchmark datasets do not necessarily resemble the traffic occurring in real operational networks (Sommer and Paxson 2010). In a review of 30 leading security studies, Arp et al (2022) report sees pervasive issues with sampling bias, data snooping, and spurious correlations. In these instances, a model can learn incidental characteristics of the data set such as ranges of addresses, instead of characteristics that actually represent malicious behavior. This issue is shown in TON_IoT by Dharini et al (2026) and a bias-aware feature refinement approach is proposed, which is used in the present study. Other drawbacks of widely used network intrusion detection system (NIDS) benchmarks are duplicate records and incorrectly labeled records, both of which can artificially boost the reported performance (Liu et al 2022). In general, ID models have poor generalisation capabilities across data sets and operating environments (Apruzzese et al 2022).

While there is literature supporting the use of each of the components in our framework, it is not clear where a novelty detector should be placed in the classification pipeline, or whether a learned feature representation would be beneficial or a simpler decision-level attachment would be sufficient. This study aims to answer this question by comparing the open-set detection, calibration, computational cost, and operational alarm precision in a controlled setup and under the same experimental protocol.

3 Research Questions

RQ1. Do the hybrid configurations alter closed-set multiclass classification performance compared with the standalone EfficientNet model on TON_IoT?

RQ2. Is the Isolation Forest more effective at detecting withheld attack traffic when it operates on the reduced input features (decision-level attachment) or on the learned EfficientNet embedding (feature-level attachment)?

RQ3. How does withheld-attack detection vary with the benign false-alarm rate, and what alarm precision do these operating points produce under a realistic attack base rate?

RQ4. What additional inference latency, storage, and energy costs are introduced by the hybrid configurations compared with the standalone network?

4 Methodology

This section describes how the proposed open-set intrusion-detection framework was designed and evaluated. Section 4.1 formalises the detection problem. Section 4.2 explains the framework shown in Fig. 1 and identifies the improvements this study makes to the conventional closed-set design. Section 4.3 describes the dataset and its preprocessing. Sections 4.4–4.7 then present the four processing stages in the order in which a flow record passes through them, showing how the equation that governs each stage is obtained and defining every variable it contains. Section 4.8 sets out the evaluation protocol.

4.1 Problem formulation

Let $D = \{(x_i, y_i)\}_{i=1}^{N}$ denote the set of N labelled flow records, where $x_i \in \mathbb{R}^d$ is the d-dimensional feature vector of the i-th flow and $y_i \in Y_{known} = \{1, \ldots, C\}$ is its class label, with $C = 8$ known classes. At deployment, the system may also receive flows whose labels belong to $Y_{unknown}$, the set of attack behaviours that were not observed during training. The task is therefore to learn a decision function g that assigns every observation either to one of the known classes or to a rejection outcome, denoted $\perp$ (Scheirer et al 2013):

$$g : \mathbb{R}^d \to Y_{known} \cup \{\perp\} \tag{1}$$

A conventional closed-set classifier can only output a label in Y_{known}, so every flow from $Y_{unknown}$ is necessarily assigned to a known class. The framework described next adds the rejection outcome $\perp$ through a confidence router and a novelty detector.

4.2 Overview of the proposed framework

Fig. 1 shows the complete pipeline. A labelled flow record (left of the figure) passes through four numbered stages. Stage 1 reduces the record to a compact representation z by principal component analysis (PCA) followed by electromagnetic field optimisation (EFO) subset selection (Section 4.4). Stage 2 passes z through a compact EfficientNet whose penultimate layer produces the learned embedding $e = f_\theta(z)$ (Section 4.5). Stage 3 classifies e with a gradient-boosted decision tree (GBDT) ensemble, which outputs a predicted label $\widehat{y}$ and a class-probability vector p (Section 4.6). Stage 4, the confidence router, compares the largest class probability $p_{\max}$ with the threshold $\tau_{conf} = 0.95$. Flows with $p_{\max} \geq 0.95$ keep the GBDT label (upper-right

box), whereas flows with $p_{\max} < 0.95$ are passed to the Isolation Forest novelty stage (lower-right box), which either confirms the GBDT label or returns the unknown outcome $\perp$ (Section 4.7). At the bottom of the figure, the two dashed lines represent the two different inputs to the Isolation Forest compared in this study: the reduced representation z (decision-level attachment, green) and the learned embedding e (feature-level attachment, purple).

The design starts by creating a conventional closed set network where EfficientNet (Tan and Le 2019, 2021) is trained end to end and the final class decision is done using the softmax layer. EfficientNet has been successfully applied to intrusion detection by mapping tabular network traffic features to image-like grids (Tekleselassie 2022). Closed-set intrusion detection systems, however, are tested only against attack classes that they are trained against, thus making them susceptible to previously unseen (zero-day) attacks (Farrukh et al 2023; Souza et al 2025). Thus, unknown-attack detection can be built on a well-trained closed-set classifier (Vaze et al 2022). This study builds on that base in three ways, all of which are evident in Fig. 1. First, the softmax head is only used during training and as a reference baseline (dashed box inside Stage 2), and the class decision deployed is made by a GBDT head fitted on out-of-fold embeddings (Stage 3). Second, a confidence router (Stage 4) is placed downstream from the classifier to only look at flows that the classifier is not very confident about. Third, an Isolation Forest novelty stage is added to provide the reject result ($\perp$) that the closed set baseline does not have, and it is inserted at two different attachment points to evaluate the impact of the representation used by the detector while keeping everything else constant. All fitted components are trained on data that are different from the locked test partition, while the embeddings for the GBDT head are trained on out-of-fold embeddings to ensure that the embeddings seen during training are similar to those available at inference, which reduces the risk of evaluation bias (Arp et al 2022).

4.3 Dataset and preprocessing

We use the network component of the TON_IoT dataset (Moustafa 2021), collected from the testbed described by Alsaedi et al (2020). Eight traffic classes are included in the closed-set training and evaluation procedure, with performance assessed on a locked test partition containing 25,421 flows. Injection and man-in-the-middle (MITM) traffic, comprising 21,005 flows in total, are excluded from the training data, the classification head, and the Isolation Forest. These withheld classes are introduced only during the open-set evaluations reported in Sections 5.6 and 5.7.

Although the network dataset was published in 2021, it was chosen deliberately, for four reasons. First, it remains one of the most widely used public benchmarks for IoT and IIoT intrusion detection, and the recent studies against which this work is positioned, including Ismail et al (2025), Zahoor et al (2025), Dharini et al (2026) and De Bernardis et al (2026), all evaluate on it; using the same benchmark makes the present results directly comparable with that current literature. Second, it combines normal traffic with nine labelled attack classes collected from a heterogeneous IoT/IIoT testbed, which provides enough distinct attack families to withhold two of them entirely while still training on eight known classes, as the open-set protocol

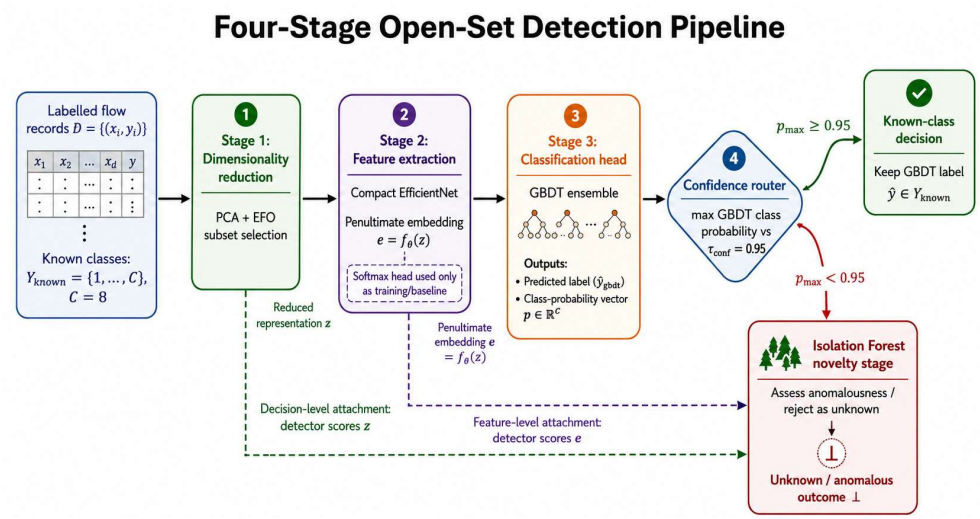


Fig. 1 The proposed four-stage open-set detection pipeline

Table 1 Test-partition composition

Class	Flows	Class	Flows
backdoor	2,807	password	2,979
ddos	2,999	ransomware	2,210
dos	2,849	scanning	3,000
normal	6,306	xss	2,271
Known-class total	25,421	Withheld (injection, mitm)[1]	21,005

[1]Withheld classes are excluded from training, the GBDT head and the Isolation Forest, and are used only in the open-set evaluation.

requires. Third, its known weaknesses, in particular address and port identifiers that permit shortcut learning, are documented (Dharini et al 2026; Booij et al 2022) and can therefore be controlled explicitly, as described below. Fourth, the dataset is openly available, which allows the experiments to be reproduced. The limits that this choice places on generalisation to supervisory control and data acquisition (SCADA) traffic in petroleum pipelines are discussed in Sections 6 and 7.

To reduce the risk that the models learn network-specific identifiers rather than behavioural characteristics, we follow Dharini et al (2026) and remove source and destination addresses, port numbers, and timestamps. Categorical variables are encoded using mappings derived solely from the training partition, with previously unseen categories assigned to a reserved value. Continuous variables are standardised using statistics computed from the training data only. All decision thresholds are selected on the validation partition and fixed before evaluation on the locked test set.

4.4 Stage 1: Dimensionality reduction by PCA and EFO

The raw flow features are strongly correlated; Section 5.1 shows that the most label-relevant of them form tight, protocol-specific blocks. Stage 1 removes this redundancy before feature learning in two steps: PCA constructs a compact orthogonal basis, and EFO then selects the subset of that basis that is most useful for classification.

Principal component analysis.

Let $X \in \mathbb{R}^{n \times d}$ be the matrix of the n preprocessed training flows, each with d features, centred so that every column has zero mean. PCA seeks the orthogonal directions along which the training data vary most (Jolliffe and Cadima 2016). The spread of the data is summarised by the sample covariance matrix

$$\Sigma = \frac{1}{n-1} X^T X \tag{2}$$

where $\Sigma \in \mathbb{R}^{d \times d}$ and X^T is the transpose of X. The variance of the data projected onto a unit-length direction v is $v^T \Sigma v$. Maximising this variance subject to $\|v\| = 1$ with a Lagrange multiplier λ, and setting the derivative of $v^T \Sigma v - \lambda \left(v^T v - 1\right)$ with respect to v to zero, gives the eigenvalue equation

$$\Sigma v_j = \lambda_j v_j, \quad j = 1, \ldots, d \tag{3}$$

where v_j is the j-th eigenvector (principal direction) and $\lambda_j \geq 0$ is the corresponding eigenvalue, equal to the variance of the data along v_j. Sorting the eigenvalues in decreasing order and stacking the first k eigenvectors as columns gives the projection matrix $V_k = [v_1, \ldots, v_k] \in \mathbb{R}^{d \times k}$. Each preprocessed flow $\widetilde{x}$ is then projected onto these k directions:

$$u = V_k^T \widetilde{x} \tag{4}$$

where $u \in \mathbb{R}^k$ is the vector of principal-component scores. In this study $k = 30$ components were computed; the first 17 explain 95% and all 30 explain 99.8% of the training variance (Table 2).

Electromagnetic field optimisation.

Retaining components by explained variance alone is not necessarily optimal for classification, because the variance of a component and its relevance to the class label are different quantities. EFO (Abedinpourshotorban et al 2016) is therefore used to choose which of the k components to keep. EFO is a physics-inspired population metaheuristic in which candidate solutions are treated as electromagnetic particles: the population is partitioned into positive, negative and neutral fields, and new candidates are generated by attraction toward high-fitness solutions and repulsion from low-fitness solutions, with the golden ratio governing the balance of these forces. Each candidate subset $S \subseteq \{1, \ldots, k\}$ is encoded as a binary mask of length k and scored by the fitness function

$$F(S) = \alpha \cdot \mathrm{MacroF1}_{val}(S) - \beta \cdot \frac{|S|}{k} \tag{5}$$

where $MacroF1_{val}(S)$ is the macro-averaged F1-score obtained on the validation partition by a fast proxy classifier (a Random Forest; Table 2) trained on the training partition using only the components in S; $|S|$ is the number of selected components; k is the total number of components; and α and β are non-negative weights. The first term rewards subsets that preserve class-discriminative information, and the second term penalises retained dimensionality, discouraging unnecessarily large subsets. EFO searches for the subset S^* that maximises $F(S)$ over 50 iterations, and the locked test set is not used at any point in the search. The search retained $|S^*| = 26$ of the $k = 30$ components. The reduced representation passed to Stage 2 is therefore

$$z = u_{S^*} \in \mathbb{R}^{26} \tag{6}$$

that is, the principal-component scores of Eq. (4) restricted to the 26 components in S^*.

4.5 Stage 2: Feature extraction with EfficientNet

The reduced vector z is the input to a compact EfficientNet, a convolutional architecture that scales network depth, width and input resolution jointly through compound scaling (Tan and Le 2019). Because EfficientNet expects spatially structured input, z is arranged into an array with an imposed spatial structure before the first convolution (Section 2.1) (Sharma et al 2019). Let f_θ denote the part of the network from the input up to and including the penultimate layer, where θ is the set of trainable weights and biases of all its convolutional, squeeze-and-excitation (SE) and dense layers. The learned feature vector, or embedding, of a flow is the output of this penultimate layer:

$$e = f_\theta(z) \tag{7}$$

where z is the reduced representation of Eq. (6) and $e \in \mathbb{R}^{d_e}$ is the d_e-dimensional embedding. During training, the embedding is followed by a softmax output layer with weight vectors w_c and biases b_c for each class c, which converts e into class probabilities:

$$q_c(z) = \frac{\exp\left(w_c^T e + b_c\right)}{\sum_{j=1}^{C} \exp\left(w_j^T e + b_j\right)} \tag{8}$$

where $q_c(z)$ is the softmax probability that the flow belongs to class c. The parameters θ, w_c and b_c are learned by minimising the cross-entropy loss over the N_{tr} training flows:

$$L = -\frac{1}{N_{tr}} \sum_{i=1}^{N_{tr}} \sum_{c=1}^{C} y_{i,c} \log q_c\left(z_i\right) \tag{9}$$

where $y_{i,c}$ equals 1 if training flow i belongs to class c and 0 otherwise. Training ran for nine epochs, with a learning rate of 10^{-3} for the first six epochs and 5×10^{-4} thereafter (Table 2). After training, the softmax layer of Eq. (8) is not used for the deployed decision; it is retained only as the closed-set reference baseline of Hendrycks and Gimpel (2017), and the embedding e of Eq. (7) is passed to Stage 3.

4.6 Stage 3: GBDT classification head

Tree ensembles are strong classifiers for tabular data (Grinsztajn et al 2022), so the deployed class decision is made by a GBDT ensemble fitted on the embedding e. Gradient boosting builds an additive model one tree at a time: at each round, a new regression tree is fitted to the negative gradient of the loss with respect to the current model output, and its prediction is added to the model after shrinkage (Friedman 2001; Chen and Guestrin 2016). For C classes, one additive score function is grown per class, so that after M rounds the score of class c is

$$F_c(e) = \sum_{m=1}^{M} \eta h_{m,c}(e) \tag{10}$$

where $h_{m,c}$ is the regression tree added for class c in round m, M is the number of boosting rounds (trees per class), and $\eta \in (0, 1]$ is the learning rate that shrinks the contribution of each tree. The class scores are converted into a probability vector $p = (p_1, \ldots, p_C)$ by the softmax transformation, and the predicted class is the one with the highest probability:

$$p_c = \frac{\exp\left(F_c(e)\right)}{\sum_{j=1}^{C} \exp\left(F_j(e)\right)}, \qquad \widehat{y} = \underset{c \in Y_{known}}{\arg\max}\, p_c \tag{11}$$

where p_c is the estimated probability that the flow belongs to class c and $\widehat{y}$ is the GBDT label. The GBDT head is trained on out-of-fold embeddings, that is, embeddings computed for each training flow by a network that was not fitted on that flow, so that its training inputs resemble the embeddings of unseen test flows.

4.7 Stage 4: Confidence routing and Isolation Forest novelty detection

The largest entry of p measures how confident the GBDT head is in its own decision:

$$p_{\max} = \max_{c \in Y_{known}} p_c \tag{12}$$

The confidence router compares $p_{\max}$ with the threshold τ_{conf} and decides whether the flow x needs further examination:

$$\text{route}(x) = \begin{cases} 1, & \text{if } p_{\max} < \tau_{conf} \\ 0, & \text{otherwise} \end{cases} \tag{13}$$

where $\tau_{conf} = 0.95$ was selected on the validation partition. Flows with $route(x) = 0$ keep the GBDT label $\widehat{y}$; flows with $route(x) = 1$ are forwarded to the Isolation Forest. Routing therefore conditions the novelty stage on low classifier confidence: the Isolation Forest never sees traffic the classifier is confident about, and at $\tau_{conf} = 0.95$ it receives roughly 9% of the stream.

The Isolation Forest (Liu et al 2008, 2012) is fitted on benign training flows only, so it learns the profile of normal traffic. It is an ensemble of isolation trees, each grown on a random subsample of ψ training points by repeatedly choosing a random feature

Table 2 Experimental configuration

Stage	Parameter	Value
Feature reduction	PCA components computed; retained by EFO	30; 26
	Cumulative variance at 17 / 30 components	95% / 99.8%
	EFO iterations; proxy classifier	50; Random Forest
Feature extractor	Training epochs; learning-rate schedule	9; 10^{-3} for 6 epochs, then 5×10^{-4}
	Training time; serialised size	4.40 min; 70.8 MB (.keras)
GBDT head	Fitted on; input	out-of-fold embeddings; e
Router	Signal; confidence threshold τ_{conf}	max GBDT class probability; 0.95
	Share of stream routed at τ_{conf}	$\approx 9\%$
Isolation Forest	Deviation threshold τ; fitting data	0.006; benign training profile
	Training time; serialised size	0.17 min; 20.6 MB (.pkl)
Evaluation	Test partition; withheld families	25,421 flows; injection, mitm (21,005)

and a random split value until every point is isolated. Points that differ from the benign profile are separated after few splits and therefore end on short paths from the root. For an observation v, let $h(v)$ be its path length, the number of edges from the root to the terminal node in one tree, and $E\left[h(v)\right]$ its average over all trees. Because the expected path length grows with the subsample size, it is normalised by $c(\psi)$, the average path length of an unsuccessful search in a binary search tree built on ψ points:

$$c(\psi) = 2H(\psi - 1) - \frac{2(\psi - 1)}{\psi} \tag{14}$$

where $H(i) \approx \ln i + 0.5772$ is the i-th harmonic number and 0.5772 is Euler's constant. The anomaly score of v is then

$$s(v, \psi) = 2^{-\frac{E[h(v)]}{c(\psi)}} \tag{15}$$

Scores close to 1 indicate short isolation paths and therefore strong atypicality, whereas scores well below 0.5 indicate observations consistent with the benign profile. The two attachment variants differ only in the vector v supplied to Eq. (15): $v = z$ (Eq. (6)) under decision-level attachment and $v = e$ (Eq. (7)) under feature-level attachment. Let $\delta(v)$ denote the deviation of $s(v, \psi)$ from the scores of the benign training profile. The final output of the pipeline for a forwarded flow is

$$\hat{y}_{final} = \begin{cases} \perp, & \text{if } \delta(v) > \tau \\ \hat{y}, & \text{otherwise} \end{cases} \tag{16}$$

where $\tau = 0.006$ is the deviation threshold, selected on validation data and frozen before evaluation on the test set. Algorithm 1 summarises the complete inference procedure for both variants, and Table 2 lists the experimental configuration.

Algorithm 1 End-to-end inference, both attachment variants

Require: flow record x; scaler; projection V_k; EFO mask S^*; feature extractor f_θ; GBDT head G; Isolation Forest IF; thresholds $\tau_{conf} = 0.95$, $\tau = 0.006$; variant $\in$ {decision-level, feature-level}

Ensure: label $\hat{y} \in Y_{known}$ or the unknown outcome $\perp$

$\tilde{x} \leftarrow$ preprocess(x) ▷ training-partition encodings and statistics

$u \leftarrow V_k^T \tilde{x}$; $z \leftarrow u_{S^*}$ ▷ PCA projection, Eq. (4), EFO subset, Eq. (6)

$e \leftarrow f_\theta(z)$ ▷ EfficientNet embedding, Eq. (7); softmax head not evaluated

$p \leftarrow G(e)$; $\hat{y} \leftarrow \arg\max p$; $p_{\max} \leftarrow \max p$ ▷ GBDT head, Eqs. (11)–(12)

if $p_{\max} \geq \tau_{conf}$ **then**

  return $\hat{y}$ ▷ confident: novelty stage not invoked, Eq. (13)

end if

$v \leftarrow z$ **if** variant = decision-level **else** e

$s \leftarrow s(v, \psi)$ by Eqs. (14)–(15); $\delta(v) \leftarrow$ deviation of s from the benign profile

if $\delta(v) > \tau$ **then**

  return $\perp$ ▷ reject as unknown, Eq. (16)

else

  return $\hat{y}$ ▷ keep the GBDT label

end if

4.8 Evaluation protocol

Closed-set performance is evaluated using overall accuracy together with per-class precision, recall, F1-score, and support. Threshold-independent performance is assessed using the area under the receiver operating characteristic (ROC) curve (AUC), precision-recall curves, and average precision (AP). Precision-recall analysis is included because ROC curves can give an overly favourable impression when the class distribution is highly imbalanced, whereas precision-recall curves are more sensitive to changes in false-positive behaviour (Davis and Goadrich 2006; Saito and Rehmsmeier 2015).

Probability calibration is examined using reliability diagrams and per-class expected calibration error (ECE). To assess how the learned representation differs from the original input space, both spaces are visualised through two-dimensional t-distributed stochastic neighbour embedding (t-SNE) projections (van der Maaten and Hinton 2008). The detection thresholds are treated as operating points rather than as parameters selected to maximise a single performance measure. The choice of threshold is dependent upon the attack rate in the traffic stream being monitored because a low false-positive rate may generate a high amount of false alarms for a traffic stream with a low attack rate (Axelsson 2000). Therefore, each operating point is given as well as the alarm precision associated with a specified attack base rate, π:

$$\text{Alarm precision} = \frac{\pi \cdot DR}{\pi \cdot DR + (1 - \pi) \cdot FPR} \tag{17}$$

where π is the attack base rate (ABR), i.e., the proportion of attack flows in the monitored stream; DR is the detection rate (DR), i.e., the proportion of attack flows that raise an alarm; and FPR is the false-positive rate (FPR), i.e., the proportion of benign flows that raise an alarm. The numerator is the expected fraction of all flows

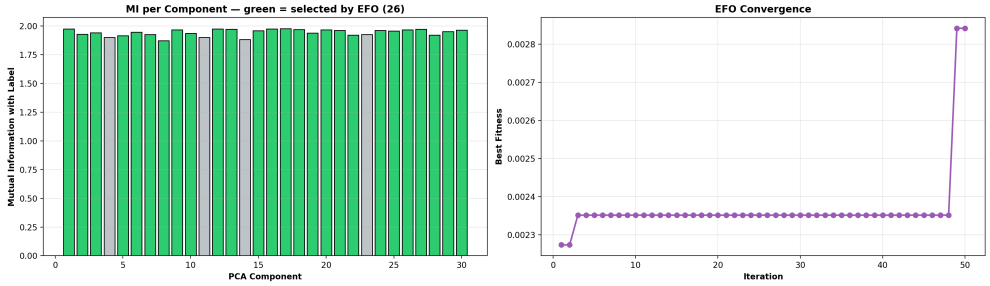


Fig. 2 Estimated mutual information between each PCA component and the class label, with EFO-selected components in green (left), and EFO best-fitness convergence over 50 iterations (right).

which are correctly alarmed attacks, the denominator is the expected fraction of all flows that raise any alarm, hence the ratio is the fraction of all alarms which are true attacks.

Computational costs are assessed through training time, serialised model size, per-sample inference latency, throughput, resident memory consumption and energy use.

5 Results

5.1 Feature reduction

The EFO procedure saved 26 out of 30 PCs. Fig. 2 displays the estimate of mutual information of each component with class label and the convergence behaviour of the search. Mutual information is fairly evenly distributed across the components, with a value of about 1.9 nats, which means that class related information is not contained in just a few dominant components. The same is true for the selected subset, as the four components removed are spread throughout the component set and not limited to the last several directions.

Convergence trace exhibits significant improvement at iteration 3 and at iteration 48, and is not much different between these two iterations. This indicates that there are many subsets of the search space with similar fitnesses, and the occasional configuration that is much better. This type of behaviour is similar to binary subset selection with several components that provide overlapping or redundant information.

Fig. 3 shows why dimensionality reduction is available at all. The twenty-five most label-relevant raw features organise into three tight blocks corresponding to the Domain Name System (DNS), Secure Sockets Layer (SSL) and Hypertext Transfer Protocol (HTTP) protocol families, with high within-block and low between-block correlation. A compact orthogonal basis removes this redundancy without discarding the protocol information itself.

5.2 Representation

Fig. 4 compares the input feature space with the EfficientNet embedding space using t-SNE (van der Maaten and Hinton 2008), with the withheld injection and MITM

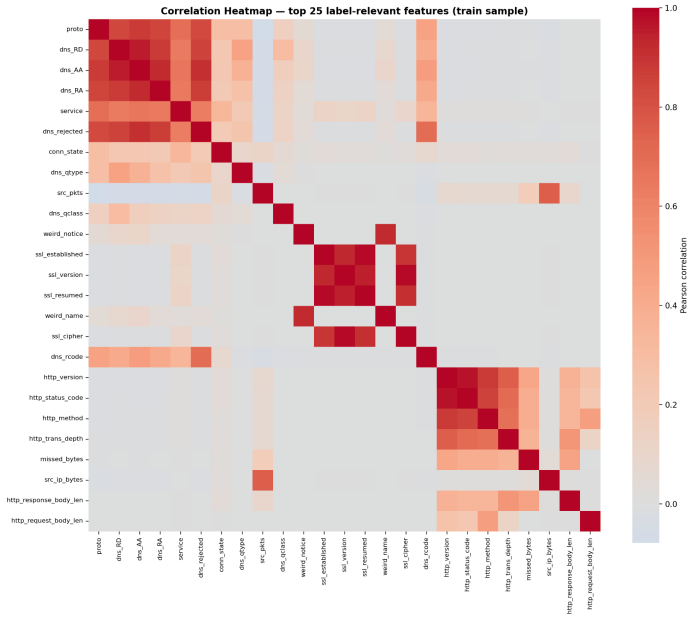


Fig. 3 Pearson correlation among the twenty-five most label-relevant raw features, showing DNS, SSL and HTTP redundancy blocks.

samples shown as crosses. In the input representation, the known classes already exhibit some local structure, including narrow bands and small clusters. The withheld attacks, however, are distributed across the projection and often appear close to regions occupied by known classes.

The EfficientNet embedding produces a more compact organisation of the known classes, while the withheld attacks form several visually distinct clusters. This pattern provides the initial motivation for evaluating the feature-level attachment strategy. The projection also shows an important limitation. Although the withheld samples form dense groups, several of these groups remain embedded among the known-class regions rather than appearing clearly separated from them. Their apparent clustering in two dimensions therefore does not imply that they are isolated in the full embedding space. Since Isolation Forest operates on the full representation rather than on the t-SNE projection, visual separation alone cannot be taken as evidence of stronger novelty detection. The results in Section 5.6 confirm that these two forms of separation do not align in this case.

5.3 Closed-set classification

The GBDT head correctly classifies 24,979 of the 25,421 test flows, corresponding to an accuracy of 98.26% and a macro-F1 score of 0.9789. Table 3 compares this result with the EfficientNet softmax head evaluated on the same embedding. Their closed-set performance is effectively identical: both achieve a macro-F1 of 0.9789 and correctly classify the same number of test flows. The minor per-class differences therefore

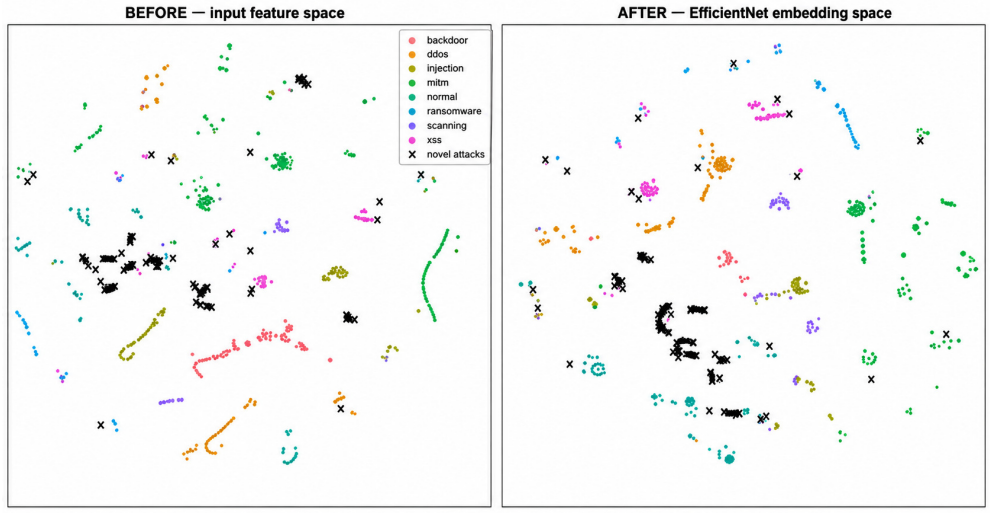


Fig. 4 t-SNE projection of the input feature space (left) and the EfficientNet embedding space (right), with withheld injection and MITM observations marked as crosses.

reflect a redistribution of errors rather than an overall improvement in classification. Replacing the softmax head with a boosted-tree classifier does not improve closed-set performance in this setting. The novelty-detection stage also has limited scope to change this result because it is applied only to observations routed as low-confidence cases. This establishes an important baseline for the remainder of the study. With the primary classifier already achieving approximately 98% accuracy on known traffic, the potential value of the cascade lies mainly in how it handles uncertain and previously unseen observations rather than in further gains in closed-set classification.

Fig. 5 gives the per-class view and Fig. 6 the confusion matrix in counts and row-normalised form. Ransomware precision (0.9246) and cross-site scripting (XSS) recall (0.9384) are the weakest entries, and these are the two smallest classes at 2,210 and 2,271 flows.

Fig. 7 shows that the remaining classification errors are concentrated in a small number of class pairs. The most prominent error occurs when XSS traffic is classified as ransomware, accounting for 121 misclassifications. The reverse confusion, ransomware classified as XSS, contributes a further 51 cases. Together, these two directions represent about 39% of the total closed-set errors.

This concentration suggests that the main limitation of the classifier lies in separating these two attack categories rather than in the hybrid decision mechanism. Because both classes are associated with web or payload-related behaviour and are relatively underrepresented in the test partition, further gains in closed-set performance are more likely to come from improved feature design, targeted class rebalancing, or other measures aimed specifically at this confusion pair. Changes to the hybrid decision layer would be unlikely to address this source of error directly.

Table 3 Per-class performance of the EfficientNet softmax head (baseline) and the GBDT head on the same embedding

		P / R / F1	
Class	Support	Softmax head	GBDT head
backdoor	2,807	1.0000 / 0.9989 / 0.9995	1.0000 / 0.9989 / 0.9995
ddos	2,999	0.9922 / 0.9703 / 0.9811	0.9925 / 0.9710 / 0.9816
dos	2,849	0.9877 / 0.9842 / 0.9859	0.9904 / 0.9817 / 0.9861
normal	6,306	0.9968 / 0.9929 / 0.9948	0.9962 / 0.9933 / 0.9948
password	2,979	0.9902 / 0.9849 / 0.9875	0.9925 / 0.9836 / 0.9880
ransomware	2,210	0.9246 / 0.9765 / 0.9498	0.9246 / 0.9765 / 0.9498
scanning	3,000	0.9848 / 0.9923 / 0.9885	0.9806 / 0.9943 / 0.9874
xss	2,271	0.9496 / 0.9384 / 0.9440	0.9501 / 0.9384 / 0.9442
Accuracy	25,421	0.9826	0.9826
Macro average	25,421	0.9782 / 0.9798 / 0.9789	0.9784 / 0.9797 / 0.9789

P precision; R recall; F1 F1-score.

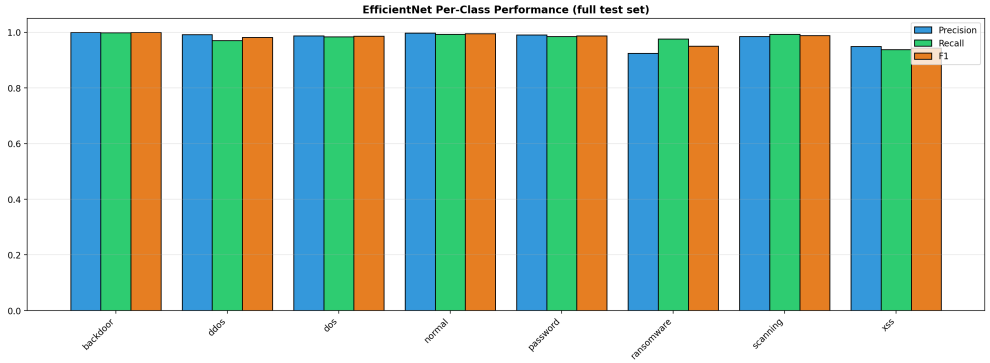


Fig. 5 Per-class precision, recall and F1 on the full known-class test set.

Fig. 8 reports the mean activation magnitude of the final squeeze-and-excitation block by class across the compressed feature axis. The normal class dominates the map, with activations three to eight times those of any attack class, while the attack classes present a broadly similar low-magnitude profile. The network encodes benign traffic with a distinctive high-magnitude response and distinguishes attack families through smaller differences. We report this as an observation about the learned representation; it describes activation magnitude rather than class-discriminative attention, and we do not draw interpretability conclusions from it.

5.4 Confidence and calibration

Fig. 9 compares the classifier's maximum class probability for correct and incorrect known-class predictions, and for known against withheld attack observations. The

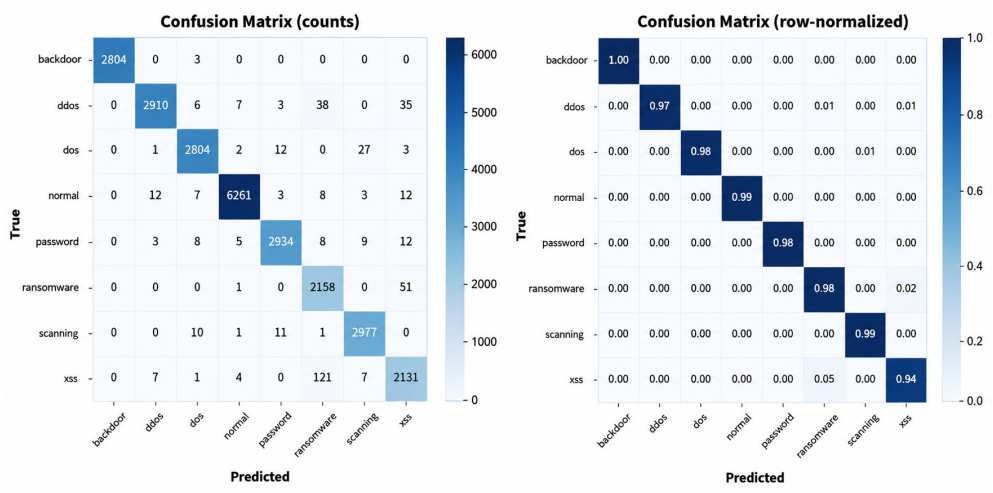


Fig. 6 Closed-set confusion matrix in counts (left) and row-normalised form (right).

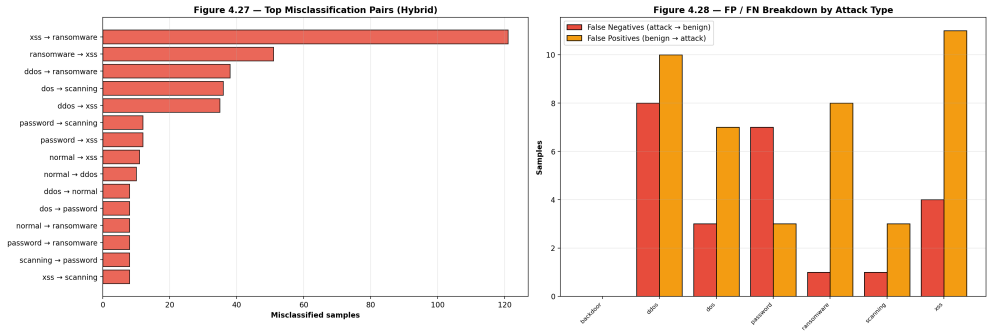


Fig. 7 Top misclassification pairs (left) and false-positive and false-negative counts by attack type (right).

right-hand panel is the central motivation for adding a novelty component: withheld injection and MITM traffic receives high softmax confidence across much of the range and overlaps substantially with known-class traffic, so no confidence threshold separates the two cleanly.

Fig. 10 and Table 4 summarise the calibration results before and after applying per-class isotonic regression. The model is already well calibrated before correction, with a mean per-class expected calibration error of 0.00098. Isotonic regression produces only a negligible change, reducing the mean ECE to 0.00096. The effect is also inconsistent across classes. Calibration improves for ddos, dos, and normal traffic, but deteriorates for backdoor, password, ransomware, scanning, and XSS. XSS shows the largest post-calibration error, with an ECE of 0.00170, and also experiences the greatest deterioration after correction. The deviation from the ideal calibration line visible

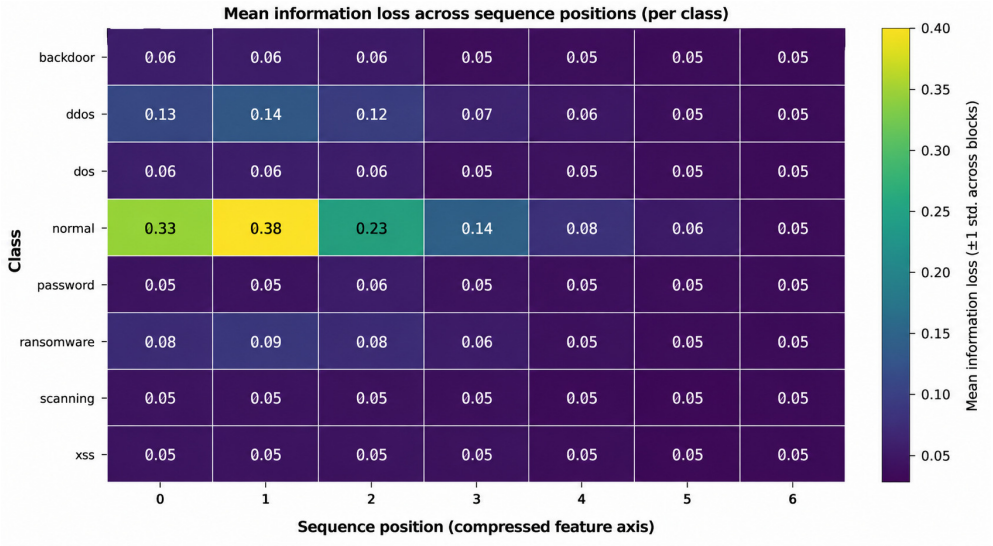


Fig. 8 Mean absolute activation of the final squeeze-and-excitation block by class across the compressed feature axis.

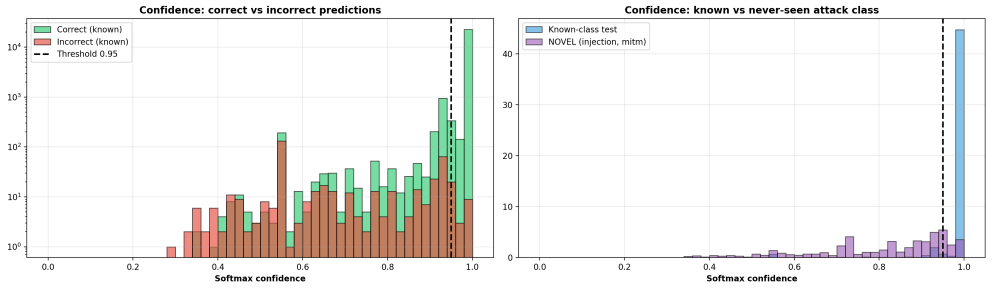


Fig. 9 Classifier confidence for correct versus incorrect known-class predictions (left) and for known-class versus withheld attack observations (right).

in the left panel of Fig. 10 occurs mainly in low-confidence bins containing relatively few observations. Consequently, these local deviations have little effect on the overall calibration error. Given the already low pre-calibration ECE and the absence of a consistent improvement after isotonic regression, the calibration stage is not retained in the deployed pipeline.

5.5 Binary detection performance

Fig. 11 presents the precision-recall results, while Fig. 12 shows the corresponding ROC performance for the binary attack-versus-benign task evaluated over the complete traffic stream, including the withheld attack classes. EfficientNet achieves an average

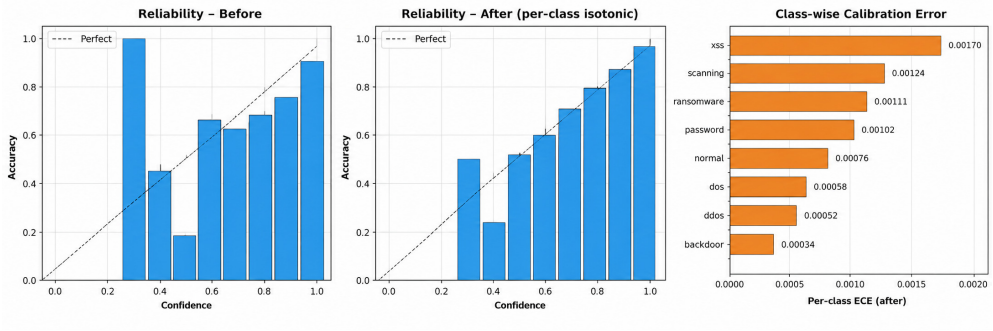


Fig. 10 Reliability before (left) and after (centre) per-class isotonic calibration, and per-class calibration error after correction (right).

Table 4 Per-class expected calibration error before and after per-class isotonic regression

Class	ECE before	ECE after	Change
backdoor	0.000118	0.000148	degraded
ddos	0.001444	0.001021	improved
dos	0.000853	0.000672	improved
normal	0.001599	0.001106	improved
password	0.000766	0.000946	degraded
ransomware	0.001122	0.001245	degraded
scanning	0.000798	0.000861	degraded
xss	0.001148	0.001705	degraded
Mean	0.000981	0.000963	no material change

precision of 0.9996 and an AUC of 0.9976. By comparison, Isolation Forest alone records an average precision of 0.8903 and an AUC of 0.7158, while the hybrid model achieves 0.9992 and 0.9962, respectively. On both threshold-independent measures, the hybrid configuration performs slightly below the supervised EfficientNet model. These results therefore provide no evidence that hybridisation improves the overall ranking of attack and benign traffic. Its contribution is instead observed under the open-set conditions examined in Sections 5.6 and 5.7, where the closed-set supervised classifier has no mechanism for identifying previously unseen attacks as unknown. The class-specific precision-recall results in Fig. 11 also identify XSS as the weakest-performing class, with an average precision of 0.982, consistent with the confusion patterns reported in Fig. 7.

Fig. 13 makes the consequence of the cascade concrete on binary decisions. At $\tau_{conf} = 0.95$ the novelty stage moves a single observation in each direction, leaving specificity at 0.993 and attack recall at 0.874 unchanged. Roughly 9% of the stream is routed at this threshold, so the stage is active but almost always confirms the label it receives. This is the binary counterpart of the closed-set result in Table 3, and it

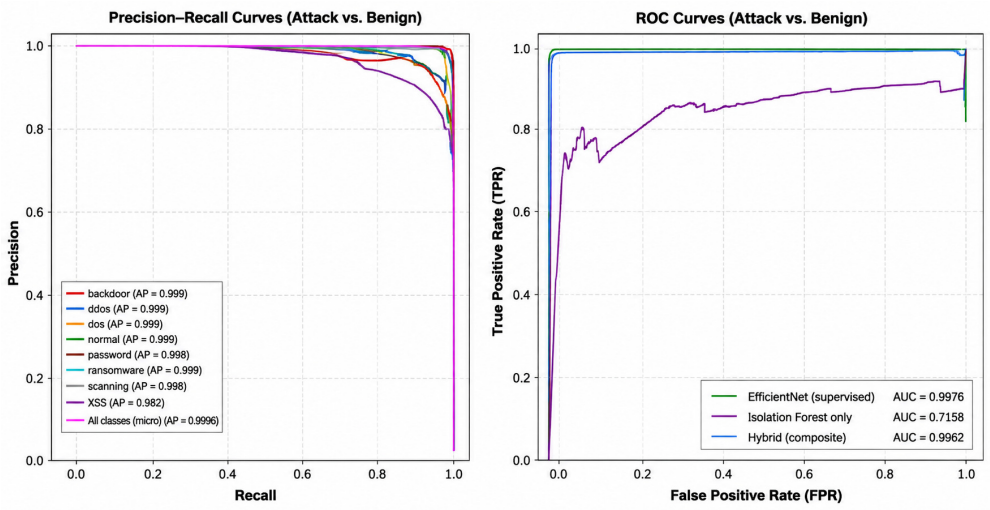


Fig. 11 Precision-recall curves per attack class (left) and binary average-precision comparison including withheld attacks (right).

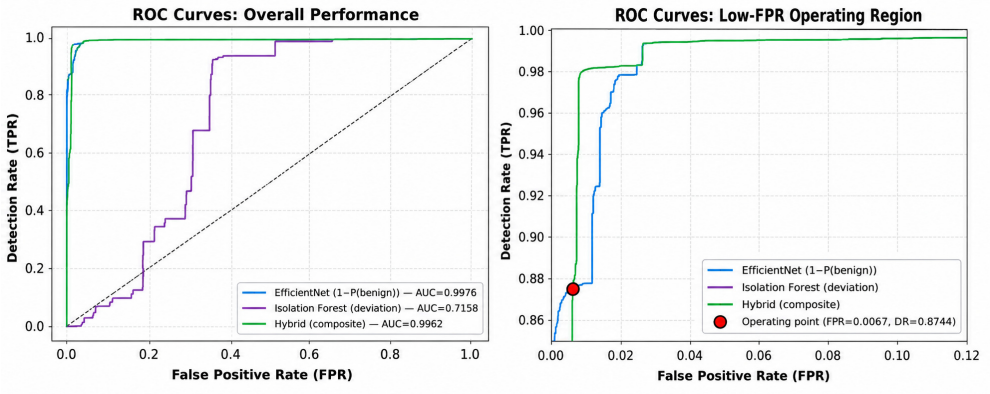


Fig. 12 Binary ROC comparison including withheld attacks (left) and the low-false-positive region in detail (right).

reinforces that the value of the architecture lies entirely in its treatment of traffic outside the training label space.

5.6 Routing behaviour and withheld-attack capture

The novelty detector can only assess observations that are passed on by the confidence router, so its effective reach depends directly on the choice of τ_{conf}. Fig. 14 illustrates this relationship across a range of threshold values. As τ_{conf} increases from 0.5 to 1.0, the proportion of observations retained by the classification head decreases from

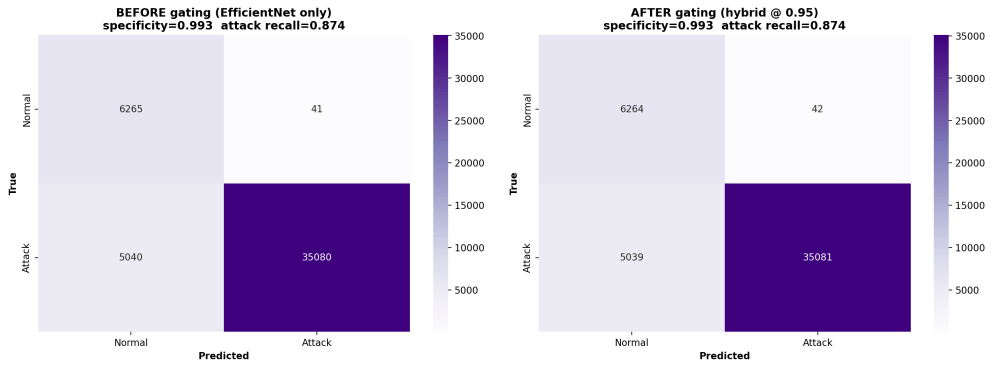


Fig. 13 Binary confusion matrices before and after the gating stage at $\tau_{conf} = 0.95$.

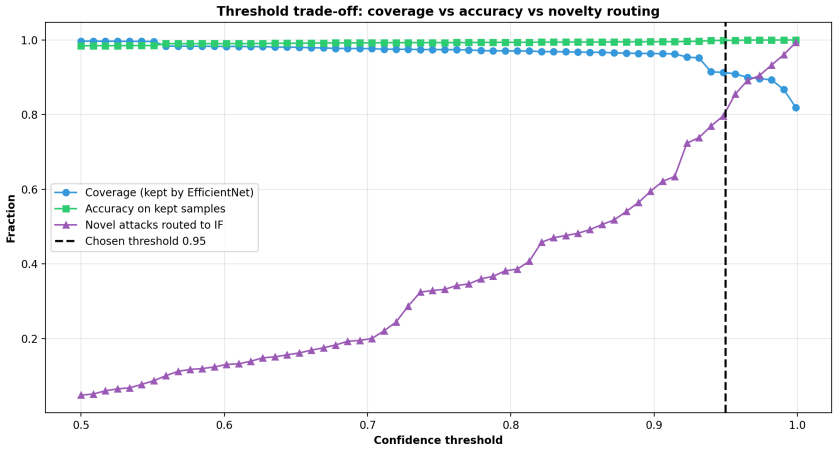


Fig. 14 Routing-threshold sweep: coverage retained by the classification head, accuracy on retained flows, and the fraction of withheld traffic forwarded to the Isolation Forest.

1.00 to 0.82, while the proportion of withheld attack traffic forwarded to the novelty stage rises from 0.05 to 0.99. These trends reflect the same routing decision from two perspectives. Importantly, the classification accuracy of retained observations remains close to 1.00 across the sweep. Increasing the threshold therefore reduces coverage rather than the correctness of predictions that are retained. At the operating point used in this study, $\tau_{conf} = 0.95$, the classification head retains approximately 91% of the traffic stream and forwards 81.5% of the withheld attack traffic. This forwarding rate sets the upper bound on the proportion of withheld attacks that the downstream novelty detector can identify.

Fig. 15 summarises the resulting detection performance. The left panel confirms the closed-set result. Both the EfficientNet softmax baseline and the GBDT-headed pipeline achieve a multiclass accuracy of 98.26%, while the standalone Isolation Forest records 86.58% on the binary attack-versus-benign task. The centre panel presents

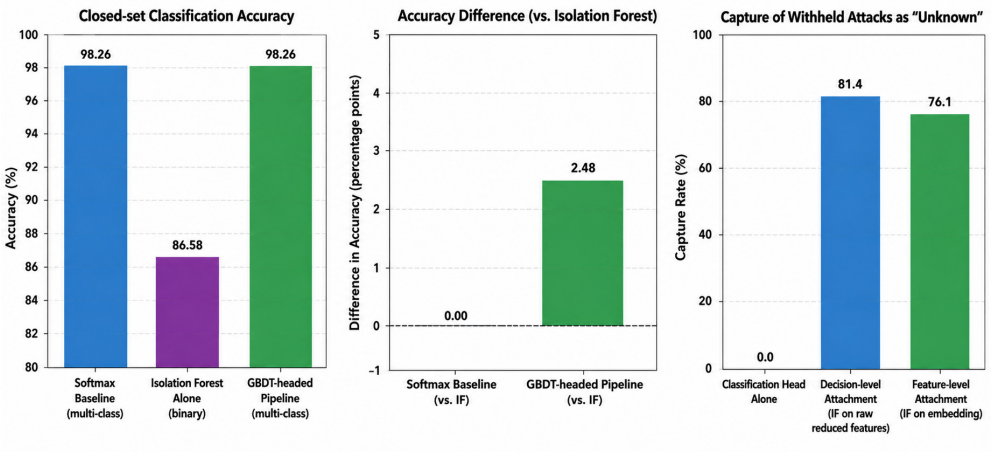


Fig. 15 Component versus pipeline accuracy (left), improvement in percentage points (centre), and withheld-attack capture for the three arrangements (right).

these differences directly, showing a 2.48 percentage-point advantage of the pipeline over the Isolation Forest and no improvement over the baseline classification head. The right panel presents the main open-set result. The classification head alone identifies none of the withheld attacks as unknown, which follows from its closed-set design because no unknown class is available at prediction time. With decision-level attachment, where the Isolation Forest operates on the reduced input features, 81.4% of withheld attacks are identified. Feature-level attachment, in which the detector operates on the learned embedding, achieves 76.1%. The feature-level configuration therefore performs 5.3 percentage points below the decision-level configuration. Neither result exceeds the 81.5% of withheld traffic forwarded by the confidence router, which sets the maximum proportion available for detection at the novelty stage.

This result runs against the expectation set by the representation-learning literature (Ruff et al 2018) and against the qualitative impression of Fig. 4. We offer the following account. The embedding is optimised for discrimination among the eight known classes, so it has no incentive to preserve variation that is irrelevant to that objective, including the variation that makes a ninth behaviour unusual.

5.7 Operating points and alarm precision

Fig. 16 shows the effect of varying the override threshold across 42 evaluated operating points. Withheld-attack capture remains at 76.1% while the benign false-alarm rate is below approximately 1.4%. Capture then increases sharply as the false-alarm rate moves through the range of about 7% to 11%, reaching 99.8% at a 13.9% false-alarm rate and 100% at 19.9%. The steep section of the curve is not used for operating-point selection because relatively small changes in the threshold produce large changes in system behaviour. Operating points outside this region provide a more stable balance between withheld-attack detection and benign false alarms.

Table 5 Summary of the compared arrangements

System	Closed-set accuracy	Binary AUC / AP	Withheld attacks identified
EfficientNet softmax head (baseline)	98.26%	0.9976 / 0.9996	0.0%[1]
GBDT head on embedding, no novelty stage	98.26%	–	0.0%[1]
Isolation Forest alone (raw features)	86.58% (binary)	0.7158 / 0.8903	–
Cascade, decision-level attachment	98.26%	0.9962 / 0.9992	81.4%
Cascade, feature-level attachment	98.26%	–	76.1%

AUC area under the ROC curve; AP average precision; – not measured / not applicable.

[1]By construction: a closed-set classifier has no rejection outcome.

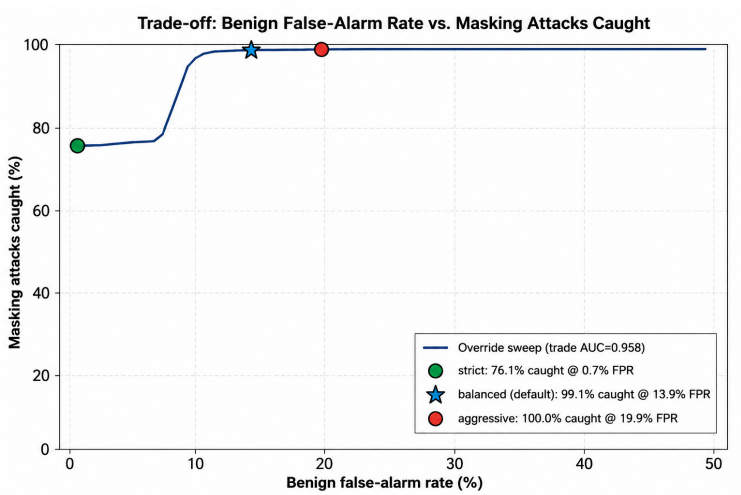


Fig. 16 Withheld-attack capture against benign false-alarm rate across the override sweep, with the strict, balanced and aggressive points marked.

Table 6 translates the operating points into alarm precision, which more directly reflects the burden faced by an operations team. At an attack base rate of 1%, the strict operating point yields an alarm precision of 54.2%, meaning that slightly more than half of the alerts correspond to genuine attacks. Under the aggressive setting, alarm precision falls to 4.8%. When the attack base rate is reduced to 0.1%, the difference becomes more pronounced, with alarm precision decreasing to 10.5% at the strict point and to 0.5% at the aggressive point. These results show that detection rate alone is insufficient for selecting an operational threshold. A setting that captures more attacks may also generate an alert stream dominated by false positives. For continuous monitoring, we therefore adopt the strict operating point, which captures 76.1% of withheld attacks at a benign false-alarm rate of 0.65%. Higher-capture

Table 6 Operating points with implied alarm precision, computed as $\pi DR/(\pi DR + (1-\pi)FPR)$ (Eq. 17)

Operating point	Withheld capture	Benign FPR	Alarm precision	
			$\pi = 1\%$	$\pi = 0.1\%$
Strict (adopted)	76.1%	0.65%	54.2%	10.5%
Intermediate	99.8%	13.9%	6.8%	0.7%
Aggressive	100.0%	19.9%	4.8%	0.5%

π attack base rate; DR detection rate; FPR false-positive rate.

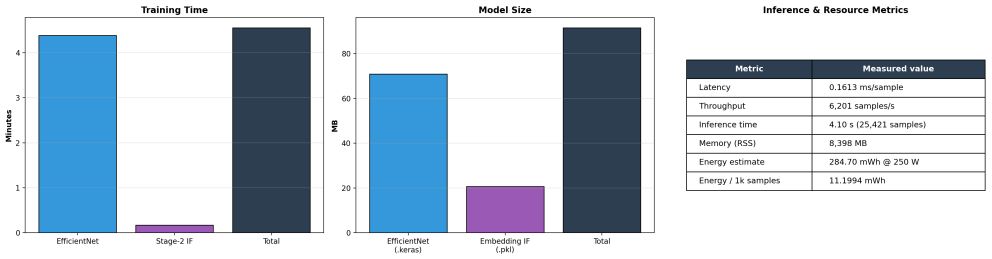


Fig. 17 Training time by stage (left), serialised model size (centre), and measured inference and resource metrics (right).

operating points are better suited to incident-response conditions, where a temporary increase in analyst workload may be acceptable.

5.8 Computational cost

Fig. 17 summarises the computational and resource costs of the proposed pipeline. Training requires 4.40 minutes for the EfficientNet model and 0.17 minutes for the Isolation Forest, giving a total training time of 4.57 minutes. The serialised model size is 70.8 MB for the network and 20.6 MB for the detector, resulting in a combined storage requirement of 91.4 MB. Inference over the 25,421-flow test partition takes 4.10 s, corresponding to 0.1613 ms per sample and a throughput of approximately 6,201 samples/s. The measured resident memory usage is 8,398 MB. Assuming a 250 W power draw, the estimated energy consumption is 284.70 mWh for the full test partition, or 11.20 mWh per 1,000 samples. By comparison, the network alone processes the same test partition in 0.87 s. Adding the hybrid novelty-detection stage therefore increases inference time by a factor of approximately 4.7.

The throughput is high and the training cost trivial, so speed at the gateway is not the constraint. The constraint is footprint. A 91.4 MB model and an 8.4 GB resident set are workstation figures, and we do not claim edge deployability on this evidence; establishing it would require the quantisation and physical deployment measurement that De Bernardis et al (2026) demonstrate for comparable models on microcontrollers, which we identify as the next engineering step. The relevant exchange for a monitoring

Table 7 Measured resource profile

Measure	EfficientNet	Isolation Forest	Complete pipeline
Training time	4.40 min	0.17 min	4.57 min
Serialised size	70.8 MB	20.6 MB	91.4 MB
Inference, 25,421 flows	0.87 s	–	4.10 s
Latency / throughput	–	–	0.1613 ms; 6,201 samples/s
Resident memory	–	–	8,398 MB
Energy at 250 W	–	–	284.70 mWh (11.20 mWh/1k)

gateway is that a 4.7-fold increase in inference time and 20.6 MB of additional model buys detection of 81.4% of traffic the supervised classifier cannot flag at all.

6 Discussion

Three main findings emerge from the results, two of which show that added architectural complexity does not improve closed-set performance.

First, neither the gradient-boosted classification head nor the novelty-detection stage provides a measurable advantage under the closed-set condition. The softmax baseline and the GBDT head correctly classify the same 24,979 of 25,421 flows and achieve an identical macro-F1 score of 0.9789. In the binary evaluation, the cascade changes the outcome of only one observation among 46,426, while its AUC and average precision are slightly lower than those of the baseline. This behaviour follows directly from the structure of the cascade. The novelty detector receives only the approximately 9% of observations for which the GBDT confidence falls below the routing threshold and therefore cannot affect predictions for the remaining 91%. When the first-stage classifier is already correct on approximately 98% of known traffic, there is little scope for a secondary detector to improve closed-set accuracy. This point is particularly important on TON_IoT, where lightweight models have already reported comparable classification performance at substantially lower computational cost (Ismail et al 2025; De Bernardis et al 2026). The present results therefore provide no basis for claiming an advantage for the hybrid architecture in terms of closed-set classification.

The benefit of the architecture becomes apparent under the open-set condition. A conventional closed-set classifier cannot identify withheld attack traffic as unknown because the output space contains no rejection class. By contrast, the cascade with decision-level attachment identifies 81.4% of the withheld attack observations. Much of this capability is introduced by the confidence router. At $\tau_{conf} = 0.95$, the router forwards 81.5% of withheld traffic to the novelty detector, effectively setting the upper bound on the proportion that the second stage can identify. The resulting open-set performance should therefore be attributed to the combined effect of confidence-based routing and novelty detection rather than to the Isolation Forest alone. For gateway-based monitoring, this ability to recognise traffic outside the learned class space is more consequential than marginal changes in closed-set accuracy.

The third finding concerns the placement of the novelty detector. Contrary to the expectation suggested by representation-learning approaches (Ruff et al 2018), feature-level attachment performs worse than decision-level attachment. When Isolation Forest operates on the learned embedding, it detects 76.1% of withheld attacks, compared with 81.4% when the same detector is applied to the reduced input features. A plausible explanation is that supervised feature learning retains information that is useful for distinguishing the known classes while suppressing variation that does not contribute to the training objective. Some of this discarded variation may nevertheless be useful for distinguishing previously unseen attack traffic. The result therefore suggests that a learned representation optimised for closed-set discrimination is not necessarily the most informative space for novelty detection. This observation may extend beyond the present architecture to other systems that apply unsupervised anomaly detection to representations learned under supervised objectives.

The operating-point analysis further shows why detection rate alone is insufficient for assessing practical usefulness. An operating point that detects 99.8% of withheld attacks while producing a 13.9% benign false-alarm rate results in an alarm precision of only 6.8% when the attack base rate is 1%. At a 0.1% attack base rate, alarm precision falls to 0.7%. Such an operating point would generate a large volume of false alerts relative to genuine attacks and would therefore impose a substantial burden on operational monitoring. In practice, the acceptable operating point is constrained not only by detection performance but also by the number of false alarms that analysts can reasonably investigate (Axelsson 2000; Sommer and Paxson 2010). For this reason, the stricter operating point is adopted in the present study, and alarm precision is reported alongside detection and false-positive rates.

These findings should, however, be interpreted within the limits of the benchmark. TON_IoT was collected in an IoT and IIoT testbed environment (Moustafa 2021; Alsaedi et al 2020) and does not reproduce the communication protocols, temporal regularity, or process dependencies found in SCADA systems used for petroleum-pipeline monitoring. Industrial control datasets such as the Mississippi State gas-pipeline captures (Morris and Gao 2014) and the Secure Water Treatment (SWaT) testbed dataset (Goh et al 2016) exhibit different traffic and process characteristics, while poor cross-dataset generalisation remains a recognised problem in intrusion detection research (Apruzzese et al 2022). The present results therefore support the architectural findings observed under the TON_IoT experimental setting.

7 Limitations and Future Work

Both of the two attack families are created in the same testbed environment with the same data-collection process as the training traffic, though they are not included in the open-set evaluation. Purely novel attacks can vary in origin, traffic characteristics, tooling, and class label. The reported capture rates should thus be considered as possibly optimistic estimates of performance against truly novel attacks (Arp et al 2022). It also only covers injection and MITM traffic, and not whether the decision-level attachment has an advantage for other families of attacks. A leave-one-family-out evaluation would be a better test of this result and is an important direction for further work.

The current results are estimates based on one data partition and a single random seed and so should not be treated as a confidence interval. The main comparison result is the 5.3 percentage difference between the two hybrid configurations, which will need to be repeated with several different seeds before it could be considered robust. We also did not conduct a specific audit for exact or close copies of records in the training and test partitions. This overlap may lead to overreporting of performance in network intrusion-detection benchmarks (Liu et al 2022) and a duplicate analysis should be added to future evaluations.

The study does not attempt to make generalisation beyond TON_IoT, nor does it consider the temporal drift of the decision thresholds. A temporally ordered evaluation would be more suitable to consider the performance at varying traffic conditions (Pendlebury et al 2019). The present analysis also assumes that an adversary is not adaptive. An attacker who knows the detection architecture, can attempt to craft traffic to look as though it is in a dense area of a known class, increasing the chances of it not being rejected. This is especially true because of the results in Section 5.6, where sometimes the unfamiliar traffic is mapped into dense areas of the supervised embedding.

Future work will thus be directed in three directions. In the first place, we will investigate if the representations learned to maintain the density structure perform better at novelty detection than purely discriminative embeddings. Second, we will evaluate edge deployment via model quantisation and direct measurement on real hardware, similar to the evaluation done by De Bernardis et al (2026). Third, we will evaluate the architecture on other industrial control datasets, including gas-pipeline and SWaT, to see if the behaviour seen with TON_IoT applies to other data sets.

8 Conclusion

We investigated the ability of a cascaded intrusion-detection architecture on TON_IoT that consisted of a small EfficientNet feature extractor, a gradient-boosted decision tree classification head, and an Isolation Forest applied to low-confidence observations. Two novelty-detection setups were tested: (1) the Isolation Forest was run on the reduced input features, and (2) the Isolation Forest was run on the learned embedding.

There is no benefit from the extra components in a closed-set evaluation. In the eight known classes, the EfficientNet softmax baseline and the GBDT based configurations score 98.26% and the novelty stage flips only one prediction in the binary evaluation. The difference comes in under open set conditions. The closed-set classifier has no rejection class, and thus it does not identify any of the withheld attacks as unknown. Decision-level attachment is able to detect 81.4% withheld attacks, while feature-level attachment is able to detect 76.1% withheld attacks. The poor performance of the learned embedding indicates that supervised representation learning could mask variation that is irrelevant to discriminating between known classes but important for detecting previously unseen behaviour.

Better capture rates can be achieved, but at the cost of significant false alarm rates. If withheld-attack capture is increased to 99.8%, it will be necessary to accept 13.9% benign false-alarm rate, which means an alarm precision of only 6.8% at the

stated base rate. Thus we choose the more conservative operating point which achieves 76.1% of withheld attacks with an alarm precision of 54.2%. This open set ability also comes with a computational price. The cascade adds 4.7 times to the inference time and 20.6 MB to the model size.

The results show that the primary benefit of the cascade is that it is able to classify previously unseen attack traffic and not necessarily improve the classification of known classes. They demonstrate that the position of the novelty detector is significant – the simpler decision-level configuration is superior to attachment to the supervised embedding. These results are only valid for this current TON_IoT assessment and should be repeated on several different random seeds and industrial control datasets before deployment can be recommended.

Acknowledgements. [To be completed by the authors, or delete this heading.]

Declarations

- **Funding:** [The authors did not receive support from any organisation for the submitted work. / Details of funding.]
- **Competing interests:** The authors declare no competing interests.
- **Ethics approval and consent to participate:** Not applicable. The study uses the publicly available TON_IoT dataset and involves no human participants or animals.
- **Consent for publication:** Not applicable.
- **Data availability:** The TON_IoT network dataset analysed in this study is publicly available from the UNSW Canberra Cyber TON_IoT repository (Moustafa 2021; Alsaedi et al 2020).
- **Materials availability:** Not applicable.
- **Code availability:** [Available from the corresponding author on reasonable request. / Repository link.]
- **Author contribution:** [I.M.K.: conceptualisation, methodology, software, experiments, writing – original draft; S.E.A., O.F. and T.K.: supervision, validation, writing – review and editing. All authors read and approved the final manuscript.]

References

Abedinpourshotorban H, Shamsuddin SM, Beheshti Z, Jawawi DNA (2016) Electromagnetic field optimization: a physics-inspired metaheuristic optimization algorithm. Swarm Evol Comput 26:8–22. https://doi.org/10.1016/j.swevo.2015.07.002

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

Apruzzese G, Pajola L, Conti M (2022) The cross-evaluation of machine learning-based network intrusion detection systems. IEEE Trans Netw Serv Manag 19(4):5152–5169. https://doi.org/10.1109/TNSM.2022.3157344

Arp D, Quiring E, Pendlebury F, Warnecke A, Pierazzi F, Wressnegger C, Cavallaro L, Rieck K (2022) Dos and don'ts of machine learning in computer security. In: Proceedings of the 31st USENIX Security Symposium, Boston, MA, USA, pp 3971–3988

Axelsson S (2000) The base-rate fallacy and the difficulty of intrusion detection. ACM Trans Inf Syst Secur 3(3):186–205. https://doi.org/10.1145/357830.357849

Bendale A, Boult TE (2016) Towards open set deep networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, pp 1563–1572. https://doi.org/10.1109/CVPR.2016.173

Booij TM, Chiscop I, Meeuwissen E, Moustafa N, den Hartog FTH (2022) ToN_IoT: the role of heterogeneity and the need for standardization of features and attack types in IoT network intrusion data sets. IEEE Internet Things J 9(1):485–496. https://doi.org/10.1109/JIOT.2021.3085194

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, San Francisco, CA, USA, pp 785–794. https://doi.org/10.1145/2939672.2939785

Davis J, Goadrich M (2006) The relationship between precision-recall and ROC curves. In: Proceedings of the 23rd International Conference on Machine Learning (ICML), Pittsburgh, PA, USA, pp 233–240. https://doi.org/10.1145/1143844.1143874

De Bernardis EP, Kuznetsov O, Arnesano M, Zhansaya P, Sydykova M (2026) Lightweight machine learning intrusion detection for IoT/IIoT networks: quantisation strategies and physical deployment on resource-constrained microcontrollers. Electronics 15(13):2869. https://doi.org/10.3390/electronics15132869

Dharini N, Janani VS, Katiravan J (2026) Efficient detection of intrusions in TON-IoT dataset using hybrid feature selection approach. Sci Rep 16:7763. https://doi.org/10.1038/s41598-026-37834-y

Farrukh YA, Wali S, Khan I, Bastian ND (2023) Detecting unknown attacks in IoT environments: an open set classifier for enhanced network intrusion detection. arXiv:2309.07461. https://doi.org/10.48550/arXiv.2309.07461

Friedman JH (2001) Greedy function approximation: a gradient boosting machine. Ann Stat 29(5):1189–1232. https://doi.org/10.1214/aos/1013203451

Goh J, Adepu S, Junejo KN, Mathur A (2016) A dataset to support research in the design of secure water treatment systems. In: Proceedings of the 11th International Conference on Critical Information Infrastructures Security (CRITIS), Paris, France, pp 88–99. https://doi.org/10.1007/978-3-319-71368-7_8

Grinsztajn L, Oyallon E, Varoquaux G (2022) Why do tree-based models still outperform deep learning on typical tabular data? In: Advances in Neural Information Processing Systems 35 (NeurIPS), Datasets and Benchmarks Track, New Orleans, LA, USA

Guo C, Pleiss G, Sun Y, Weinberger KQ (2017) On calibration of modern neural networks. In: Proceedings of the 34th International Conference on Machine Learning (ICML), Sydney, Australia, pp 1321–1330

Hendrycks D, Gimpel K (2017) A baseline for detecting misclassified and out-of-distribution examples in neural networks. In: Proceedings of the International Conference on Learning Representations (ICLR), Toulon, France

Ismail S, Dandan S, Qushou A (2025) Intrusion detection in IoT and IIoT: comparing lightweight machine learning techniques using TON_IoT, WUSTL-IIoT-2021, and EdgeIIoTset datasets. IEEE Access 13:73468–73485. https://doi.org/10.1109/ACCESS.2025.3554083

Jolliffe IT, Cadima J (2016) Principal component analysis: a review and recent developments. Philos Trans R Soc A 374(2065):20150202. https://doi.org/10.1098/rsta.2015.0202

Liu L, Engelen G, Lynar T, Essam D, Joosen W (2022) Error prevalence in NIDS datasets: a case study on CIC-IDS-2017 and CSE-CIC-IDS-2018. In: Proceedings of the IEEE Conference on Communications and Network Security (CNS), Austin, TX, USA, pp 254–262. https://doi.org/10.1109/CNS56114.2022.9947235

Liu FT, Ting KM, Zhou ZH (2008) Isolation forest. In: Proceedings of the 8th IEEE International Conference on Data Mining (ICDM), Pisa, Italy, pp 413–422. https://doi.org/10.1109/ICDM.2008.17

Liu FT, Ting KM, Zhou ZH (2012) Isolation-based anomaly detection. ACM Trans Knowl Discov Data 6(1):3. https://doi.org/10.1145/2133360.2133363

Morris T, Gao W (2014) Industrial control system traffic data sets for intrusion detection research. In: Butts J, Shenoi S (eds) Critical Infrastructure Protection VIII. IFIP Advances in Information and Communication Technology, vol 441. Springer, Berlin, Heidelberg, pp 65–78. https://doi.org/10.1007/978-3-662-45355-1_5

Moustafa N (2021) A new distributed architecture for evaluating AI-based security systems at the edge: network TON_IoT datasets. Sustain Cities Soc 72:102994. https://doi.org/10.1016/j.scs.2021.102994

Pendlebury F, Pierazzi F, Jordaney R, Kinder J, Cavallaro L (2019) TESSERACT: eliminating experimental bias in malware classification across space and time. In: Proceedings of the 28th USENIX Security Symposium, Santa Clara, CA, USA, pp 729–746

Ruff L, Vandermeulen RA, Görnitz N, Deecke L, Siddiqui SA, Binder A, Müller E, Kloft M (2018) Deep one-class classification. In: Proceedings of the 35th International Conference on Machine Learning (ICML), Stockholm, Sweden, pp 4393–4402

Saito T, Rehmsmeier M (2015) The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets. PLoS ONE 10(3):e0118432. https://doi.org/10.1371/journal.pone.0118432

Scheirer WJ, de Rezende Rocha A, Sapkota A, Boult TE (2013) Toward open set recognition. IEEE Trans Pattern Anal Mach Intell 35(7):1757–1772. https://doi.org/10.1109/TPAMI.2012.256

Sharma A, Vans E, Shigemizu D, Boroevich KA, Tsunoda T (2019) DeepInsight: a methodology to transform a non-image data to an image for convolution neural network architecture. Sci Rep 9:11399. https://doi.org/10.1038/s41598-019-47765-6

Sommer R, Paxson V (2010) Outside the closed world: on using machine learning for network intrusion detection. In: Proceedings of the IEEE Symposium on Security and Privacy, Oakland, CA, USA, pp 305–316. https://doi.org/10.1109/SP.2010.25

Souza MMC, Pontes C, Gondim J, Garcia LPF, DaSilva L, Cavalcante EFM, Marotta MA (2025) A novel open set energy-based flow classifier for network intrusion detection. Comput Secur 104569. https://doi.org/10.1016/j.cose.2025.104569

Tan M, Le Q (2019) EfficientNet: rethinking model scaling for convolutional neural networks. In: Proceedings of the 36th International Conference on Machine Learning (ICML), Proceedings of Machine Learning Research, vol 97, Long Beach, CA, USA, pp 6105–6114

Tan M, Le Q (2021) EfficientNetV2: smaller models and faster training. In: Proceedings of the 38th International Conference on Machine Learning (ICML), Proceedings of Machine Learning Research, vol 139, pp 10096–10106

Tekleselassie H (2022) Two-dimensional projection based wireless intrusion classification using lightweight EfficientNet. J Cyber Secur Mobility 11(4):601–620. https://doi.org/10.13052/jcsm2245-1439.1145

van der Maaten L, Hinton G (2008) Visualizing data using t-SNE. J Mach Learn Res 9:2579–2605

Vaze S, Han K, Vedaldi A, Zisserman A (2022) Open-set recognition: a good closed-set classifier is all you need. In: Proceedings of the International Conference on Learning Representations (ICLR). https://openreview.net/forum?id=5hLP5JY9S2d

Zadrozny B, Elkan C (2002) Transforming classifier scores into accurate multiclass probability estimates. In: Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Edmonton, AB, Canada, pp 694–699. https://doi.org/10.1145/775047.775151

Zahoor A, Abbasi W, Babar MZ, Aljohani A (2025) Robust IoT security using isolation forest and one class SVM algorithms. Sci Rep 15:36586. https://doi.org/10.1038/s41598-025-20445-4