Proceedings of the 20th Conference on Computer Science and Intelligence Systems (FedCSIS) pp. 345–350
DOI: 10.15439/2025F5708
ISSN 2300-5963 ACSIS, Vol. 43

Out-Of-Distribution Is Not Magic: The Clash Between Rejection Rate and Model Success

Itay Meiri*, Ran Dubin†, Amit Dvir‡, Chen Hajaj§
*Dept. of Computer Science, Ariel Cyber Innovation Center, Ariel University, Israel
Email: itay.meiri2@msmail.ariel.ac.il
†Dept. of Computer and Software Engineering, Ariel Cyber Innovation Center, Ariel University, Israel
ORCID: 0000-0002-2055-2211
Email: rand@ariel.ac.il
‡Dept. of Computer and Software Engineering, Ariel Cyber Innovation Center, Ariel University, Israel
ORCID: 0000-0002-3670-0784
Email: amitdv@ariel.ac.il
§Dept. of Industrial Engineering and Management, Ariel Cyber Innovation Center, Ariel University, Israel
ORCID: 0000-0001-9940-5654
Email: chenha@ariel.ac.il

***Abstract*—Recent advancements in Internet protocols, including DNS over HTTPS (DoH) and Encrypted Service Name Indicators (ESNI), are making traditional Deep Packet Inspection (DPI) engines obsolete. Consequently, there is a growing need for next-generation traffic classification using artificial intelligence (AI). While DPI automatically categorizes unknown traffic as 'other,' AI-based models cannot automatically handle unknown or Out-of-Distribution (OOD) traffic. AI models must effectively detect and classify OOD traffic to ensure robustness, reliability, and accuracy in real-world applications; however, current research often fails to address the challenges of OOD detection.**

In this paper, we evaluate various state-of-the-art OOD detection techniques for internet traffic classification and explore the drawbacks and advantages of using different threshold levels for the model's tolerance for OOD. Our findings reveal that varying rejection rates have distinct effects on OOD techniques, leading to a change in the optimal strategy for achieving dependable and precise detection across diverse OOD scenarios. We demonstrate that adjusting rejection rates from 10% to 30% can significantly improve the True Detection Rate (TDR) by up to 50%, while the False Detection Rate (FDR) may increase by less than 10%. Moreover, we emphasize that rejection-rate-based evaluation is pivotal for next-generation flow classification, promising a substantial reduction in FDR through rigorous methodological assessment.

***Index Terms*—Out of Distribution, Traffic Classification, Malware Detection**

I. Introduction

TRAFFIC Classification (TC) is a critical process that automatically categorizes Internet network traffic into distinct classes, such as traffic attribution, application type, or benign/malicious traffic. Regardless of the task, traffic classification plays a crucial role in cybersecurity, Quality of Experience (QoE), and Quality of Service (QoS), as it enables the implementation of predetermined policies to treat traffic classes differently, optimizing network performance and reliability. Traditional traffic classification techniques have relied on classifying applications or services based on fixed port numbers [1]. While these techniques offer advantages such as user privacy preservation, speed, and wide device coverage, they are limited by their reliance on fixed ports and susceptibility to cheating via packet editing. These limitations have prompted the development of Deep Packet Inspection (DPI) classification techniques [2], which involve inspecting the actual payload of packets and are less vulnerable to cheating. However, DPI techniques are resource-intensive, slow, and lack the same privacy guarantees as port-based techniques.

The widespread adoption of encryption protocols like TLS and DoH has further complicated traffic classification, rendering classical DPI techniques obsolete [3]. Consequently, researchers have turned to Machine Learning (ML) and later Deep Learning (DL) techniques for traffic classification. While ML-based approaches initially required manual feature extraction by experts, DL techniques have emerged as promising alternatives for traffic classification [4]. Despite the success of ML and DL [5], one significant weakness persists: their inability to classify instances outside of the closed set of classes in the training data. Network traffic is inherently dynamic, with new applications continually being introduced. This dynamic nature makes it challenging for models to accurately classify unseen classes without retraining. Additionally, acquiring samples for new classes is time-consuming and often results in limited datasets, leading models to favor older and more represented classes during training [6]. To address these challenges and ensure accurate and robust classification in dynamic settings, models must be able to "reject" the classification of a sample that does not belong to any of the learned classes. This capability, known as Out-of-Distribution (OOD) detection [7], [8], is crucial for effectively handling samples outside the training set. Previous works evaluate OOD detection techniques against a well-fit, state-of-the-art model without considering the implications of using less accurate models, which are common in real-world settings [9].

In this paper, we assess the impact of OOD methods on classifier performance in both binary and multiclass traffic

classification tasks. Our contribution extends existing research by highlighting the impact of employing OOD techniques on varying accuracy-level classifiers. Furthermore, we elucidate the trade-offs associated with employing threshold-based OOD techniques by introducing a new metric, **Rejection Rate (RR)**.

These trade-offs are frequently overlooked in discussions surrounding OOD techniques, as they tend not to cater to specific model requirements. By shedding light on these trade-offs, we provide valuable insights for practitioners and researchers, enabling them to make more informed decisions when selecting OOD detection techniques for their models. For example, models connected through a pipeline might not suffer from high rejection rates or data loss if they ensure that the remaining data is of high quality and reliability. This leads to a preference for performance and a high rejection rate of OOD detection techniques.

Lastly, we examine existing OOD techniques on various tasks and datasets, showing that the performance of "state-of-the-art" OOD detection methods is contingent on external factors such as acceptable rejection rates and model performance.

II. Related Work

OOD detection has been studied under many different names, such as zero-day detection, open-set recognition, and rejection option classifiers [4], [7], [8], [10], [11], [12], [13], [14]. Shared by all is the classification model's ability to reject an input if it detects that it is from a class outside its training set. Thulasidasan et al. [10] add a rejection class, K+1, trained on a mix of OOD samples. This method is limited in practice because it demands extra OOD data—either a Generative Adversarial Network or by merging some classes—plus an architectural change and full retraining, with no guarantee that real-world OOD data will match what was seen in training. Devries and Taylor [11] instead train the model to output calibrated confidence scores, using misclassified In-Distribution (ID) samples as stand-ins for OOD. These are built in parallel to the original classification model, which does not affect the model's accuracy. Nonetheless, this requires knowledge of the internal workings of the model and retraining.

Another approach uses Siamese Neural Networks (SNN) as an OOD detection method for network traffic [12]. The authors use the ability of SNNs to find similarities in input to detect OOD, but these require clustering, data balancing, and training. He et al. [13] construct skew data and then train multiple one-class Random Forest classifiers based on the skew data. This is an expensive operation and gets increasingly more complex as the number of classes increases.

Out-of-distribution detector for neural networks (ODIN) [15] augments any pre-trained model without architectural changes by pairing temperature scaling with slight input perturbations. It computes a classifier's confidence, perturbs the input proportionally, and recomputes confidence. Because ID samples are more affected, they generally yield higher post-perturbation confidences than OOD samples, enabling threshold-based rejection. Our work is the first to test ODIN on encrypted network traffic for OOD detection.

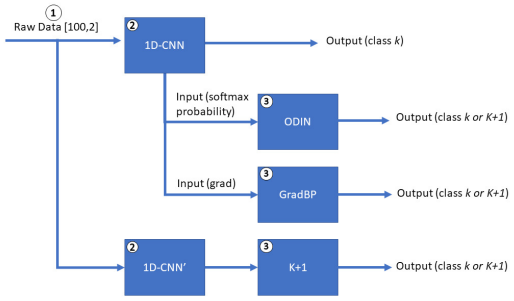

Fig. 1. Our Evaluation System

GradBP [4] treats the magnitude of the first back-propagation gradient, calculated as if training continued, as an OOD proxy. Novel inputs typically induce larger gradient magnitudes, allowing a single threshold to suffice for OOD detection. GradBP is space and time-efficient, works on pre-trained models, and has already been validated on encrypted traffic.

To conclude, several solutions to traffic classification exist, each characterized by its degrees of success and practicality. However, none of these works explores the influence of thresholds and their effect on OOD classification. As we will demonstrate, the influence of the threshold is critical for adequately evaluating OOD techniques.

Our main goal is to evaluate OOD techniques and compare their results for traffic classification, then experiment with the effects of the threshold. We chose a typical 1D-CNN as our base architecture, and three OOD techniques - ODIN [15], GradBP [4], and K+1 [16] as a baseline. ODIN and GradBP are model-agnostic and do not require model modification, retraining, or generating any additional data. Both ODIN and GradBP operate on the model output under the same space and time complexity class and require a threshold. All methods will be concatenated to our base architecture. Our evaluation system can be seen in Fig. 1.

III. System Architecture and Evaluation Framework

A. Basic System Components

Input. Each sample is a 100-step sequence of packet *direction* (0 = client $\rightarrow$ server, 1 = server $\rightarrow$ client) and min–max scaled *size* ($[0, 1]$), a standard low-cost traffic representation [4], [17] (Fig. 1,(1)).

Output. The classifier predicts a label $k \in [1, K]$; OOD detections are mapped to $k = K + 1$.

B. Out-of-Distribution Detectors

ODIN [15]. For a trained network with logits $f(x)$, we (i) rescale them by a temperature T, (ii) add a small perturbation $\epsilon \cdot \text{sign}\big(-\nabla_x \log S(x;T)\big)$, and (iii) reject any sample whose

TABLE I
TRAFFIC DATASETS USED IN THIS STUDY. B = BENIGN, M = MALWARE, TLS = DATASET IS PREDOMINATELY TLS-ENCRYPTED.

Dataset	Role	Classes	Samples	TLS	Notes
MTAB [21]	ID	2 (B/M)	29,000	Mix	MTA + benign
TEMU [20]	ID	8 (B)	20,000	Yes	Browser / misc.
USTCB [21], [22]	OOD	5 (M)	58,000	Mix	Diverse malware
VNAT [23]	OOD	6 (B)	5,500	Mix	VPN successor

largest softmax probability $S(x;T)$ falls below a threshold δ. The triplet (T, ϵ, δ) is chosen on validation data; rejected inputs are mapped to label $K+1$.

GradBP [4]. Keeping weights frozen, we back-propagate each test point once and measure the ℓ_2-norm of the resulting gradient at the penultimate model layer, $\|\delta^{L-1}\|_2$. Points with $\|\delta^{L-1}\|_2 > \epsilon_{\text{GR}}$ are considered OOD and likewise assigned $K+1$.

K + 1 [16]. The classifier head is expanded to $K+1$ outputs and fine-tuned on a small, heterogeneous set of OOD traces labelled $K+1$. At inference, any prediction of the extra node signals an outlier.

All OOD models are under label (3) in Fig. 1.

C. *Datasets*

To obtain a balanced and realistic evaluation spectrum, we use *MTAB* and *TEMU* as our two ID references and *USTCB* and *VNAT* as the OOD challengers. Together, these four datasets cover benign and malicious traffic, multiple application protocols, and both encrypted and plain-text flows, providing a rigorous test bed for evaluating ID performance and OOD generalisation. Specifically, **MTAB** (29 k flows, 2 classes) merges malicious mail-transfer (MTA) traffic with benign sessions from ISCX2016 [18], StratosphereIPS [19], and TEMU [20], yielding a mixed TLS/plain corpus labelled *benign* or *malware*. **TEMU** (20 k flows, 8 benign classes) contains browser and miscellaneous application traffic, most of it TLS-encrypted. **USTCB** (58 k flows, 5 malware classes) augments the USTC malware set—covering the *Cridex*, *Neris*, *Mi-uref*, and *Htbot* families—with the same benign sources as MTAB, giving a malware-heavy OOD benchmark. Finally, **VNAT** (5.5 k flows, 6 benign classes) offers an updated VPN/no-VPN corpus spanning streaming, chat, file-transfer, and other activities with a mix of encrypted and clear-text flows. Table I summarises the core statistics of each dataset, while the exact pairing of datasets in each experiment is shown later in Table II.

D. *Evaluation Metrics*

We report common performance indicators used for OOD evaluation [12], [13]:

Accuracy – proportion of correct decisions among the accepted samples.

True Detection Rate (TDR) – sensitivity to OOD inputs (OOD true positives divided by all OOD cases).

False Detection Rate (FDR) – type-I error on ID data (ID false positives divided by all ID cases).

Rejection Rate (RR) – The percent of samples rejected.

Recall and **Precision** – class-label sensitivity and positive predictive value, both computed on the accepted subset.

IV. EXPERIMENTAL DESIGN

We structured our evaluation around two distinct classification tasks: **Binary Classification** (i.e., distinguishing between benign and malicious traffic) and **Multiclass Classification** (i.e., identifying various well-known applications). Each classification task was evaluated using three separate experiments:

Experiment 1: This experiment assessed the base model's performance without OOD detection and unknown classes. It simulates a standard scenario where a model is developed and tested against a predefined dataset of known classes.

Experiment 2: In this experiment, we evaluated the base model's performance without OOD detection but with the introduction of unknown classes. This mimics deploying the Experiment 1 model in a real-world setting, where data may include previously unseen classes.

Experiment 3: The final experiment aimed to address the challenge of unknown classes by evaluating the base model with OOD detection capabilities.

The structure of our experiments can be seen in Table II. All OOD detection methods were tested on the same test set for robust evaluation. For comprehensive insights, including confusion matrices, tables, and threshold graphs, we have provided detailed results in our GitHub repository [24].

Lastly, when considering the preferability of one method or another, we limited the rejection rate to 50%. The limitation has two reasons: First, we could naively reject nearly all inputs without limit until few ID samples remain and achieve perfect accuracy. We do not consider such a result helpful in evaluating OOD detection performance. Second, there is no standard rejection rate, and the authors of ODIN and GradBP do not explain the effects of different rejection rates. It is standard practice to use FDR=95%TDR, such as in [13], [12], [4], [25], [26], [15], but this result is not useful for all use cases. ODIN [15] suggests hyperparameter tuning that will result in the correct identification of 95% ID samples, but does not explain the effect on rejection rates. GradBP [4] does not give any guidelines at all. As we will demonstrate, the optimal strategy is task-dependent and model-dependent.

V. EXPERIMENTAL RESULTS

A. ***Experiment 1*** *- without unknown classes, without OOD detection*

Our first experiment assesses our models' performance in both binary and multiclass classification tasks, focusing on scenarios where unknown classes and OOD data are not considered. The base models demonstrate remarkable performance despite utilizing only packet size and direction as time-series features. As shown in the Base column of Table III, the accuracy for both Binary and Multiclass tasks reaches impressive levels of 98% and 94%, respectively. This underscores the effectiveness of our models in classifying traffic even without the need for OOD detection mechanisms.

TABLE II
COMPARISON OF ARCHITECTURES AND EVALUATION METRICS ACROSS BINARY AND MULTICLASS CLASSIFICATION TASKS. BASE IS WITHOUT INCLUDING UNKNOWN CLASSES, AND BASE* IS WITH INCLUSION OF UNKNOWN CLASSES.

Exp.	Architecture	Task	Train/Test	Unknown	Evaluation Metrics
1	**Base**	Binary Multiclass	MTAB TEMU	– –	Accuracy, Recall, Precision –
2	**Base***	Binary Multiclass	MTAB TEMU	USTCB VNAT	Accuracy, Recall, Precision –
3	**Base + ODIN**	Binary Multiclass	MTAB TEMU	USTCB VNAT	Accuracy, Recall, Precision, True/False Detection Rate, Rejection Rate –
3	**Base + GradBP**	Binary Multiclass	MTAB TEMU	USTCB VNAT	Accuracy, Recall, Precision, True/False Detection Rate, Rejection Rate –
3	**Base' + (K+1)**	Binary Multiclass	MTAB+USTCB TEMU+VNAT	USTCB VNAT	Accuracy, Recall, Precision, True/False Detection Rate, Rejection Rate –

TABLE III
PERFORMANCE COMPARISON OF MODELS WITH (BASE*) AND WITHOUT (BASE) THE INCLUSION OF UNKNOWN CLASSES IN BINARY AND MULTICLASS CLASSIFICATION TASKS.

Metric	Binary Classification		Multiclass Classification	
	Base	Base*	Base	Base*
Accuracy	0.98	0.73	0.94	0.74
Recall	0.98	0.65	0.88	0.77
Precision	0.98	0.49	0.93	0.64

*B. **Experiment 2** - with unknown classes, without OOD Detection*

Next, we introduce previously unseen classes into the classification tasks, simulating real world deployment in which the model encounters new classes and no OOD detection is applied. This setting yields a pronounced drop relative to Experiment 1 across all metrics, as reported in the $Base^*$ columns of Table III. The decline exposes the model's vulnerability to novel inputs and underscores the need for robust OOD detection to mitigate this risk.

*C. **Experiment 3** - with unknown classes, with OOD detection*

Our third experiment evaluates OOD detection performance, especially the trade-offs of using a threshold-based technique on the accuracy and overall sample rejection rates. For this task, we implemented three detection techniques that will be evaluated: ODIN[15], GradBP[4], K+1[16].

In the following subsections, we demonstrate the model's performance across several rejection rates and justify choosing one threshold over another.

*1) **Classification Task - Binary, OOD - ODIN**:* Tables IV and VI show that as the rejection rate initially increases, nearly all rejected samples are OOD, as both accuracy and TDR rise. As the threshold tightens, we observe an exponential increase in FDR with minimal accuracy gains. This is essential because there might be a significant increase in the model's accuracy by allowing the model to reject slightly more ID samples. Increasing the rejection rate from 10% to 30% produced several effects: the TDR increased from 31% to 66%, while the FDR only increased from 3% to 18%, and accuracy increased from 79% to 87%. In contrast, going from a rejection rate of 30% to 50% had exponentially diminishing returns. The TDR increased from 66% to 75%, while FDR increased from about 18% to 41%, leaving the accuracy unchanged at 87%. At that point, accuracy rates began decreasing as the threshold tightened.

Despite allowing a rejection rate of up to 50%, the best accuracy figures came from rejecting only 31% of the overall samples. The accuracy score increased from 72% to 87%, with an FDR of 18% and a TDR of 67%, as seen in Table VI. These results demonstrate that despite ODIN being originally developed for images, it is generic enough to work on network traffic. However, there are apparent weaknesses as well. ODIN misses examples that are likely to be misclassified, and nearly all OOD samples missed were classified as benign. This weakness is particularly problematic because the purpose of the model is to detect malicious traffic.

Note that the rejection rate does not influence the K+1 technique; thus, it was not included in this analysis and does not appear in Table IV.

*2) **Classification Task - Binary, OOD - GradBP**:* Similarly to ODIN, as shown in Tables IV and VI, the FDRs are low as the threshold decreases, proving that nearly all samples rejected are OOD. Unlike ODIN, the GradBP results are much better at the maximum threshold rate of 50% (98% TDR). Nearly all OOD samples have been rejected, increasing the model's accuracy to 99% - higher than the base model without OOD data. The FDR is 32%, which means most ID samples are not rejected. The higher accuracy signals, again, that this method also rejects nearly all misclassified labels. Out of the missed OOD samples, only 29/31 were classified as benign by the model, compared to 659/661 in ODIN. We also compared these values at a 30% rejection rate; 193/362 OOD samples were detected as benign, a significantly better result than ODIN. As the threshold tightens, GradBP shows an exponential increase in practical terms. This means we can significantly increase the likelihood of all passed samples being ID if data loss is not a significant concern.

TABLE IV
EXPERIMENT 3: THE INFLUENCE OF REJECTION RATE ON THE PERFORMANCE OF ODIN AND GRADBP METHODS FOR THE BINARY CLASSIFICATION TASK.

Metric	10%		30%		50%		70%		90%	
	ODIN	**GradBP**	**ODIN**	**GradBP**	**ODIN**	**GradBP**	**ODIN**	**GradBP**	**ODIN**	**GradBP**
Accuracy	0.79	0.79	0.87	**0.93**	0.87	**0.99**	0.83	**1.00**	–	**1.00**
Recall	0.66	0.66	0.66	0.66	0.67	0.67	0.67	0.67	–	**1.00**
Precision	**0.54**	0.53	0.59	**0.62**	0.60	**0.66**	0.59	**0.67**	–	**1.00**
FDR	0.03	0.03	0.18	**0.12**	0.41	**0.33**	0.65	**0.60**	–	**0.87**
TDR	**0.31**	0.30	0.66	**0.82**	0.75	**0.98**	0.80	**1.00**	–	**1.00**

TABLE V
EXPERIMENT 3: INFLUENCE OF REJECTION RATE ON MODEL PERFORMANCE FOR MULTICLASS CLASSIFICATION USING ODIN AND GRADBP.

Metric	10%		30%		50%		70%		90%	
	ODIN	**GradBP**	**ODIN**	**GradBP**	**ODIN**	**GradBP**	**ODIN**	**GradBP**	**ODIN**	**GradBP**
Accuracy	**0.79**	0.78	**0.90**	0.87	**0.96**	0.93	**0.98**	0.95	–	**0.98**
Recall	**0.82**	0.77	**0.85**	0.76	**0.86**	0.85	**0.83**	0.82	–	**0.80**
Precision	**0.70**	0.66	**0.75**	0.64	**0.79**	0.74	**0.75**	0.73	–	0.69
FDR	0.07	0.07	**0.20**	0.22	**0.39**	0.41	**0.63**	0.64	–	0.88
TDR	**0.21**	0.20	**0.70**	0.59	**0.92**	0.84	**0.97**	0.93	–	0.99

TABLE VI
EXPERIMENT 3: PERFORMANCE SUMMARY FOR BINARY CLASSIFICATION USING ODIN, GRADBP, AND K+1 METHODS.

Metric	ODIN	GradBP	K+1
Accuracy	0.88	**0.99**	0.94
Recall	**0.67**	**0.67**	0.66
Precision	0.59	**0.66**	0.63
FDR	0.19	0.32	**0.02**
TDR	0.68	**0.98**	0.86
Rejection Rate	0.31	0.49	**0.23**

*3) **Classification Task - Binary, OOD - K+1**:* As seen in Table VI, the results from adding a rejection class were inconclusive. The accuracy achieved was 93.8%, which is lower than GradBP and ODIN. In the K+1 technique, there is no threshold to work with when adding a rejection class, so the rejection rate is fixed at 23%.

*4) **Classification Task - Multiclass, OOD - ODIN**:* As can be seen from Tables V and VII, there is full utilization of the rejection rate at about 50%, achieving an accuracy, TDR, and FDR of 96%, 92%, and 39%, higher than the base model when tested without OOD data. This ability means that ODIN rejects OOD samples and misclassifies known applications. For comparison, at a 30% rejection rate, the model had good accuracy, TDR, and FDR of 90%, 70%, and 20%. Table V (30% rejection rate, ODIN). Using the threshold to increase its effectiveness allows for a greater range of risk management; if losing most data is not a significant concern, we can increase the confidence that the samples that do pass are both ID and correctly classified.

*5) **Classification Task - Multiclass, OOD - GradBP**:* The results seen in Table VII for GradBP were interesting; at a 50% rejection rate, we observed an accuracy rate of 93%, lower than ODIN. TDR was 84%, also lower than ODIN, whereas the FDR was 41% compared to 39% in ODIN. The closest result was in recall, which has 85% compared to ODIN with 86%. The increase in rejection rate did not contribute much beyond 30%.

*6) **Classification Task - Multiclass, OOD - K+1**:* As shown in Table VII, the accuracy rate was 84%, with a recall of 78% and a precision of 66%. The rejection rate accounted for only 12% of the total samples, and the TDR was just 54%. However, the FDR remained remarkably low at only 1.5%, similar to the binary model.

TABLE VII
EXPERIMENT 3: PERFORMANCE SUMMARY FOR MULTICLASS CLASSIFICATION USING ODIN, GRADBP, AND K+1 METHODS.

Metric	ODIN	GradBP	K+1
Accuracy	**0.96**	0.93	0.84
Recall	**0.86**	0.85	0.78
Precision	**0.79**	0.75	0.66
FDR	0.39	0.40	**0.02**
TDR	**0.92**	0.84	0.54
Rejection Rate	0.50	0.49	**0.12**

VI. DISCUSSION AND CONCLUSIONS

Our empirical study of ODIN, GradBP, and K+1 across binary and multiclass traffic classification tasks paints a nuanced picture; there is no universally superior OOD method. Each technique excels only under specific operational conditions, a finding with immediate consequences for how OOD detection is benchmarked and deployed. In the multiclass setting, ODIN achieved the highest overall accuracy, reaching 96% in our experiments and preserving a favorable balance between true and false detection rates once the rejection rate exceeded roughly 30%. ODIN's temperature-scaled perturbation interacts advantageously with the richer softmax that accompanies

multiclass models, yielding robustness as the class cardinality increases. By contrast, in the binary task, GradBP emerged as the preferred method whenever practitioners could tolerate moderate to high rejection rates. Above the 30% rejection threshold, GradBP drove accuracy toward 99% and pushed the TDR past 98%, maintaining lower FDR rates than ODIN.

Although often dismissed as a baseline, K+1 demonstrated excellent precision at very low FDR. However, its practical value is undercut by the need for explicit OOD samples, a fixed rejection threshold, the computation costs of model retraining, and the possible effects on model performance. Rejection rates surfaced as the governing hyperparameter in the third experiment. With RR capped at 10%, ODIN and GradBP were statistically indistinguishable in the binary task, offering practitioners no basis for preferring one over the other. Once rejection rates were relaxed into the 30%-50% regime, the behavior of both methods diverged sharply: GradBP continued to accrue accuracy and detection gains, whereas ODIN effectively plateaued. In the multiclass task, the pattern reversed - a relaxed rejection rate benefited ODIN and put it significantly ahead of GradBP across all metrics, while GradBP's improvements saturated. When rejection rates were not relaxed, the performances of GradBP and ODIN were virtually identical. The bidirectional sensitivities confirm that rejection rates reshape each detector's risk-utility frontier and must be considered when benchmarking OOD detection methods. The operational consequences could be applied immediately; in domains where practitioners could tolerate elevated rejection rates at the price of suppressing false acceptances, candidate detection methods should be tested under different rejection rate circumstances to determine the ideal fit. Our findings show that OOD detection is far from a "plug-and-play" commodity. Its efficacy is contingent upon task granularity, traffic type, and tolerance for rejected samples.

Future research endeavors should explore the impact of rejection rates across a broader array of OOD detection methods and classification domains. Specifically, how strongly rejection rates improve OOD detection and performance across OOD detection performance and model accuracy. Future investigations should explore the connection between different models, domains, and rejection rates, potentially uncovering novel approaches for improved performance.

ACKNOWLEDGMENT

This work was supported by the Ariel Cyber Innovation Center, the Israel Innovation Authority, and the Trust.AI consortium.

REFERENCES

[1] D. Moore, K. Keys, R. Koga, E. Lagache, and K. C. Claffy, "The {CoralReef} software suite as a tool for system and network administrators," in *LISA*, 2001.

[2] S. Xu, J. Han, Y. Liu, H. Liu, and Y. Bai, "Few-shot traffic classification based on autoencoder and deep graph convolutional networks," *Scientific Reports*, vol. 15, no. 1, p. 8995, 2025.

[3] P. Tang, Y. Dong, S. Mao, H.-L. Wei, and J. Jin, "Online classification of network traffic based on granular computing," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 53, no. 8, pp. 5199–5211, 2023.

[4] L. Yang, A. Finamore, F. Jun, and D. Rossi, "Deep Learning and Traffic Classification: Lessons learned from a commercial-grade dataset with hundreds of encrypted and zero-day applications," *arXiv preprint arXiv:2104.03182*, 2021.

[5] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in *Advances in Neural Information Processing Systems*, vol. 25, 2012.

[6] X. Wang, Y. Wang, Y. Lai, Z. Hao, and A. X. Liu, "Reliable open-set network traffic classification," *IEEE Transactions on Information Forensics and Security*, 2025.

[7] S. Vaze, K. Han, A. Vedaldi, and A. Zisserman, "Open-Set Recognition: A good closed-set classifier is all you need," *CoRR*, vol. abs/2110.06207, 2021. [Online]. Available: https://arxiv.org/abs/2110.06207

[8] V. Franc, D. Průša, and V. Voracek, "Optimal strategies for reject option classifiers," *CoRR*, vol. abs/2101.12523, 2021. [Online]. Available: https://arxiv.org/abs/2101.12523

[9] Z. Fang, J. Lu, and G. Zhang, "Out-of-distribution detection with non-semantic exploration," *Information Sciences*, vol. 705, p. 121989, 2025.

[10] S. Thulasidasan, S. Thapa, S. Dhaubhadel, G. Chennupati, T. Bhattacharya, and J. Bilmes, "An effective baseline for robustness to distributional shift," in *ICMLA*. IEEE, 2021, pp. 278–285.

[11] T. DeVries and G. W. Taylor, "Learning confidence for out-of-distribution detection in neural networks," *arXiv preprint arXiv:1802.04865*, 2018.

[12] Y. Chen, Z. Li, J. Shi, G. Gou, C. Liu, and G. Xiong, "Not Afraid of the Unseen: a Siamese Network based Scheme for Unknown Traffic Discovery," in *ISCC*, 2020, pp. 1–7.

[13] H. He, Y. Lai, Y. Wang, S. Le, and Z. Zhao, "A data skew-based unknown traffic classification approach for TLS applications," *Future Generation Computer Systems*, 2022.

[14] Z. Yang and W. Lin, "Unknown traffic identification based on deep adaptation networks," in *2020 IEEE 45th LCN Symposium on Emerging Topics in Networking (LCN Symposium)*. IEEE, 2020, pp. 10–18.

[15] S. Liang, Y. Li, and R. Srikant, "Enhancing the reliability of out-of-distribution image detection in neural networks," *arXiv preprint arXiv:1706.02690*, 2017.

[16] L. Neal, M. Olson, X. Fern, W.-K. Wong, and F. Li, "Open Set Learning with Counterfactual Images," in *European Conference, Munich, Germany, September 8–14, 2018*, 2018, pp. 620–635.

[17] S. Rezaei and X. Liu, "Deep Learning for Encrypted Traffic Classification: An Overview," *IEEE Communications Magazine*, vol. 57, no. 5, pp. 76–81, may 2019.

[18] G. Draper-Gil, A. H. Lashkari, M. S. I. Mamun, and A. A. Ghorbani, "Characterization of Encrypted and VPN traffic using time-related features," in *ICISSP, Rome, Italy, February 19-21*, 2016, pp. 407–414.

[19] Stratosphere, "Stratosphere Laboratory Datasets," 2015, retrieved March 13, 2020, from https://www.stratosphereips.org/datasets-overview.

[20] R. Dubin, A. Dvir, O. Pele, J. Muehlstein, Y. Zion, M. Bahumi, and I. Kirshenboim, "Analyzing HTTPS Encrypted Traffic to Identify User's Operating System, Browser and Application," in *IEEE CCNC*, June 2017.

[21] A. Lichy, O. Bader, R. Dubin, A. Dvir, and C. Hajaj, "When a rf beats a cnn and gru, together—a comparison of deep learning and classical machine learning approaches for encrypted malware traffic classification," *Computers & Security*, vol. 124, p. 103000, 2023.

[22] W. Wang and D. Lu, "USTC-TFC2016," 2016. [Online]. Available: https://github.com/yungshenglu/USTC-TFC2016

[23] S. Jorgensen, J. Holodnak, J. Dempsey, K. de Souza, A. Raghunath, V. Rivet, N. DeMoes, A. Alejos, and A. Wollaber, "Extensible Machine Learning for Encrypted Network Traffic Application Labeling via Uncertainty Quantification," *arXiv preprint arXiv:2205.05628*, 2022.

[24] I. Meiri, "Ood not magic - github repository," 2025, available at: https://github.com/ArielCyber/OOD_Is_Not_Magic.

[25] D. Hendrycks and K. Gimpel, "A baseline for detecting misclassified and out-of-distribution examples in neural networks," *arXiv preprint arXiv:1610.02136*, 2016.

[26] J. Ren, P. J. Liu, E. Fertig, J. Snoek, R. Poplin, M. Depristo, J. Dillon, and B. Lakshminarayanan, "Likelihood ratios for out-of-distribution detection," in *Advances in Neural Information Processing Systems*, vol. 32. Curran Associates, Inc., 2019.