Lightweight Traffic Classification: A Simple Baseline Matching Deep Learning Performance

Jaroslav Pesek
CESNET & CTU in Prague
Prague, Czech Republic
jaroslav.pesek@fit.cvut.cz

Jan Luxemburk
CESNET & CTU in Prague
Prague, Czech Republic
luxemjan@fit.cvut.cz

Karel Hynek
CESNET & CTU in Prague
Prague, Czech Republic
hynekkar@cesnet.cz

***Abstract*—Network traffic classification solutions have increasingly adopted complex models to improve performance. Yet, we show that a simple 1-nearest neighbor classifier—using only a compact feature vector from the first ten packets of each flow—matches or even outperforms recent deep learning models on eight popular benchmark datasets, even when trained on just 10% of the data. The evidence points to sample redundancy, not architectural invention, as the main source of reported gains. The redundant flows in benchmark datasets, combined with an inadequate evaluation, obscure the field's actual progress rate. Therefore, we call for redundancy-aware evaluation protocols and new, heterogeneous datasets that test real-world generalization in an operational environment.**

***Index Terms*—network traffic classification, datasets, benchmarks, machine learning, dataset quality**

I. Introduction

Network traffic classification (TC) has seen significant advancements in recent years, largely driven by advances in machine learning. These developments are frequently based on, and evaluated using, well-established datasets such as ISCXVPN2016 [1], CIC-IDS-2017 [2], or CTU-13 [3], which are relatively outdated and were created in controlled lab environments. A recent study by Luxemburk et al. [4] reported unexpectedly high accuracy achieved by a simple baseline k-nearest neighbors classifier across several networking datasets. The authors attributed this performance to a high degree of sample redundancy—many flows within these datasets are nearly identical. When such datasets are randomly split into training, validation, and testing subsets, identical or highly similar samples often appear in both training and test sets. This overlap can significantly inflate reported performance metrics, giving a misleading impression of a model's generalization.

We selected eight widely used networking datasets recognized by the TC research community. We tested them using a basic 1-nearest neighbor classifier (1-NN) using only packet sequences for classification and achieved results comparable to or exceeding state-of-the-art performance. Going further than Luxemburk et al. [4], we investigated redundancy by progressively subsampling the training set and observing the resulting classification accuracy and recall changes.

Our findings confirm that many popular networking datasets suffer from severe redundancy, which may have led to overestimating actual progress in the field. We call for more research into this issue to better understand its causes and to ensure that future benchmarking reflects realistic model performance.

978-3-903176-74-4

II. Related Work

The limitations of traffic classification datasets, which are synthetically generated in controlled lab environments, have been widely recognized in numerous surveys and research papers [5]. Several studies, such as Lanvin et al. [6] or Aceto et al. [7], have critically examined individual datasets, highlighting methodological flaws and dataset artifacts. Consequently, Lanvin et al. have released corrected versions or explicitly detailed their data refinement processes. Aceto et al. [7] thoroughly documented their cleaning procedures and explained the sources of errors in the ISXCVPN2016 [1].

Despite this attention, only a few studies have explored the issue of information redundancy within these datasets. Notably, Soukup et al. [8] identified redundancy as a significant dataset characteristic. They defined redundancy at a threshold level of α, e.g., $\alpha = 0.9$, meaning the fraction of the training set required to achieve at least 90% of the maximum F1 score achievable with the entire dataset. Compared to their study, we analyze a broader set of datasets and avoid relying on a fixed performance threshold. Instead, we evaluate a baseline 1-NN model across multiple training set subsamples (including extremely small ones) to observe performance degradation. Our study, therefore, offers a more comprehensive analysis of dataset characteristics and their impact on classifier performance.

III. Approach

We adopted the approach proposed by Luxemburk et al. [4], which they describe as a simple baseline. The method forms a feature vector using the first n packets and their corresponding inter-packet times. In our study, we focus specifically on the first **10 packets**, encoding each using three types of features: packet sizes s (clipped to the range $[0, 1500]$), packet directions d (with values ± 1.0), and inter-packet arrival times i (clipped to $[0, 1000]$ ms and scaled by a factor of 0.1). These values are concatenated to form a feature vector. Formally, the feature extraction function Φ, that maps each flow $\mathcal{F}$ to a real-valued vector of length 30 is defined as $\Phi(\mathcal{F}) = (s_1, \ldots, s_{10}, d_1, \ldots, d_{10}, i_1, \ldots, i_{10})$, where all elements have been preprocessed as described. This feature representation is then used for classification via the 1-NN algorithm, which

TABLE I: Comparison of classification performance across datasets and tasks. We report the state-of-the-art accuracy for each dataset, our baseline accuracy (mean ± standard deviation), the difference in percentage points, and macro-averaged recall.

Dataset	Task	SOTA Accuracy [%]	Baseline Accuracy [%]	Δ Accuracy [pp]	Macro Recall [%]
ISCXTor2016 [10]	tor, non-tor	100.00 ± 0.01 [11]	99.63 ± 0.02	-0.37 ± 0.02	95.54 ± 0.48
USTC-TFC2016 [12]	10 mal., 1 benign	98.30 [13]	96.14 ± 0.64	-2.16 ± 0.64	90.40 ± 0.76
VNAT [14]	vpn, non-vpn	98.03 [15]	99.91 ± 0.01	1.88 ± 0.01	98.97 ± 0.34
CTU-13 [3]	mal., benign	99.30 [16]	99.46 ± 0.08	0.16 ± 0.08	98.37 ± 0.50
MQTTSet [17]	4 mal., 1 benign	99.90 [18]	100.00 ± 0.01	0.10 ± 0.01	99.95 ± 0.08
ISCXVPN2016 App [1]	17 benign svc.	79.92 ± 1.31 [19]	72.54 ± 1.05	-7.38 ± 1.68	62.65 ± 2.84
ISCXVPN2016 Traffic [1]	6 benign types	81.71 ± 1.26 [19]	74.39 ± 1.35	-7.32 ± 1.85	73.87 ± 1.90
CIC-DoHBr [20]	mal., benign	99.99 [21]	99.96 ± 0.00	-0.03 ± 0.00	99.91 ± 0.02
CIC-IDS-2017 [2]	7 mal., 1 benign	95.79 [22]	99.81 ± 0.02	4.02 ± 0.02	88.01 ± 0.65

assigns each flow the class label of its closest neighbor in the feature space.

During evaluation, two sampling strategies to create a training set were applied. **Random sampling** selects samples uniformly at random. **Semi-balanced sampling** [4] adjusts the selection probability for a given class C, defined as $N_C^{-\lambda}$, where N_C represents the frequency of class C. Here, parameter $\lambda > 0$ controls the degree of bias towards less frequent labels.

IV. Evaluation

We evaluated our approach on multiple popular network traffic datasets. We measured the baseline performance using five different train/test splits. Sampling experiments were executed using both described sampling strategies. In all experiments, only the first 10 packets of each flow were used to construct 30-dimensional flow feature vectors as described in the previous section.

A. Experimental Setup

To evaluate the impact of training set size on classification performance, we varied the fraction of the full training data used for sampling. We measured two metrics: overall accuracy and macro-averaged recall. We first trained and evaluated a 1-NN classifier on the complete training set. All 1-NN models were implemented with Faiss [9] using the L^1 (Manhattan) distance.

For each measurement, we performed five independent train/test splits to account for variability in data partitioning. Furthermore, we repeated each sampling strategy five times to account for the randomness introduced by the sampling procedures, which yielded 25 runs per sampling fraction, ensuring robust accuracy and macro-average recall estimates.

B. State-of-the-art comparison

The used datasets are in Table I. All of the datasets are highly cited and used as a benchmark for evaluating the results of proposed models. As SOTA results, we chose recent papers with reported accuracy to compare our baselines. In most of the research papers in SOTA, we found several problems, as mentioned in [23], i.e., an inappropriate baseline. Except for [11] and [19], no picked paper made any effort to reduce the variance and reported only one number as a final result.

Compared papers predominantly employ deep learning or related techniques. Examples include conventional dense and CNN architectures [11], [16], [19], representation learning approaches [13], time-series models using LSTMs [15], and autoencoders [22]. A smaller subset of studies has explored random forests [18], [21]. Our baseline matches SOTA accuracy on four datasets, beats it on VNAT (+1.88) and CIC-IDS-2017 (+4.02), and loses on USTC-TFC2016 (−2.16) and the two ISCXVPN2016 variants (approx.−7.00).[1]

C. Sampling Analysis

Fig. 1 illustrates the trade-off between classification performance and size of the datasets. The left subplot in each pair shows the classification accuracy, and the right subplot shows macro-averaged recall. Fig. 2 aggregates the accuracy degradation (Δ Accuracy) relative to the full-data baseline for the semi-balanced sampler. The median loss is below 3 pp at 1% of the data and falls under 1 pp with a 10% fraction. Outliers appear only for the two ISCXVPN2016 variants.

Our experiments show that:

- Even with a severely reduced fraction of the training data, overall accuracy and macro-averaged recall drops are minimal (see aggregated accuracy results using semi-balanced sampling in Fig. 2). Most of the datasets' drops are not significant, except ISCXVPN2016.
- The semi-balanced sampling strategy consistently yields higher macro-averaged recall across all datasets.
- We additionally confirmed the redundancy-driven performance effects reported by Luxemburk et al. [4].
- Across all sampling fractions, the standard deviation of both accuracy and macro-recall stays low, indicating highly stable results not driven by "lucky" splits.
- Semi-balanced sampling mitigates class-imbalance effects.
- These findings underscore the need for future benchmark datasets to control for sample redundancy and include more heterogeneous, real-world traffic to reflect true model generalization.

V. Conclusion

In this work, we have demonstrated that a simple 1-NN classifier, operating on the 30-dimensional feature vector consisting of 10 packet lengths, directions, and inter-packet

[1] The higher performance drop on ISCXVPN2016 can be attributed to the SOTA's reliance on packet payload features. Other SOTA approaches we compared to in Table I do not use payload.

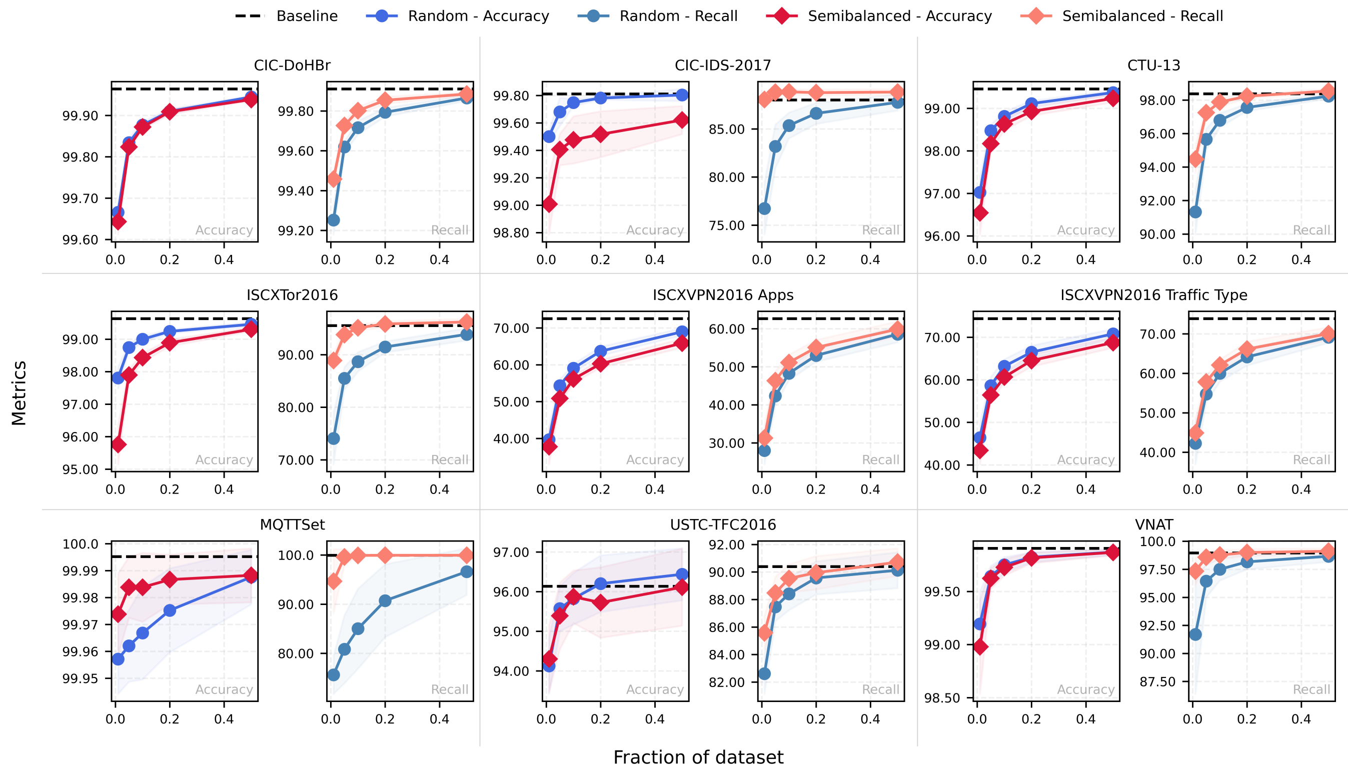

Fig. 1: Comparison of classification accuracy (left subplot of each group), weighted accuracy (macro-averaged recall; right subplot of each group), versus the fraction of training data used across multiple network traffic datasets. The black dashed lines indicate the baseline using the full training set.

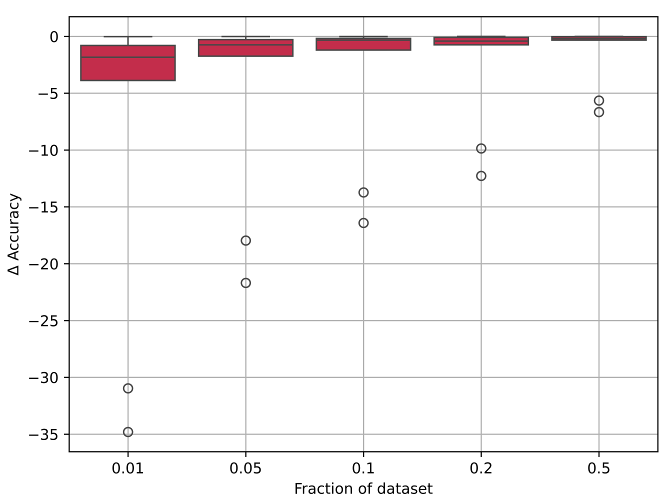

Fig. 2: Aggregated accuracy drop relative to the full-data baseline when training on fractions of the dataset (semibalanced sampling). Each box-and-whisker plot pools results across all datasets; open circles mark outliers, which correspond to the two ISCXVPN2016 variants in every case.

times, achieves performance on par with or even superior to state-of-the-art deep learning models across eight widely used network traffic benchmarks. By progressively subsampling the training data to as little as 1 %, we showed that overall accuracy and macro-averaged recall remain comparable to their full-set values, confirming that severe sample redundancy used together with random splitting drives much of the reported performance in existing literature. We further showed that a semi-balanced sampling strategy consistently improves recall for minority classes and matches random sampling in overall accuracy.

Our findings raise critical questions about current traffic classification datasets. Future work should focus on (i) constructing more heterogeneous benchmarks that better reflect real-world traffic; (ii) developing formal measures of dataset redundancy and flow similarity; and (iii) investigating lightweight classification methods that serve as baseline methods.

We hope this study will encourage the community to revisit dataset construction and benchmarking practices, ensuring progress in traffic classification translates into robust performance in operational environments.

Acknowledgment

This work was supported by the Ministry of the Interior of the Czech Republic, grant No. VJ02010024 and also by the Grant Agency of the Czech Technical University in Prague, grant No. SGS23/207/OHK3/3T/18.

References

[1] G. Draper-Gil *et al.*, "Characterization of encrypted and VPN traffic using time-related features:" in *Proceedings of the 2nd International Conference on Information Systems Security and Privacy*, Rome, Italy: SCITEPRESS - Science, 2016, pp. 407–414, ISBN: 978-989-758-167-0. DOI: 10.5220/0005740704070414.

[2] I. Sharafaldin *et al.*, "Toward generating a new intrusion detection dataset and intrusion traffic characterization:" in *Proceedings of the 4th International Conference on Information Systems Security and Privacy*, Funchal, Madeira, Portugal: SCITEPRESS - Science and Technology Publications, 2018, pp. 108–116, ISBN: 978-989-758-282-0. DOI: 10.5220/0006639801080116.

[3] S. García *et al.*, "An empirical comparison of botnet detection methods," *Computers & Security*, vol. 45, pp. 100–123, Sep. 1, 2014, ISSN: 0167-4048. DOI: 10.1016/j.cose.2014.05.011.

[4] J. Luxemburk *et al.*, *Universal embedding function for traffic classification via QUIC domain recognition pretraining: A transfer learning success*, Feb. 18, 2025. DOI: 10.48550/arXiv.2502.12930. [Online]. Available: http://arxiv.org/abs/2502.12930 (visited on 04/15/2025).

[5] D. Arp *et al.*, "Dos and don'ts of machine learning in computer security," in *31st USENIX Security Symposium (USENIX Security 22)*, 2022, pp. 3971–3988.

[6] M. Lanvin *et al.*, "Errors in the cicids2017 dataset and the significant differences in detection performances it makes," in *International Conference on Risks and Security of Internet and Systems*, Springer, 2022, pp. 18–33.

[7] G. Aceto *et al.*, "Distiller: Encrypted traffic classification via multimodal multitask deep learning," *Journal of Network and Computer Applications*, vol. 183–184, p. 102 985, Jun. 2021, ISSN: 1084-8045. DOI: 10.1016/j.jnca.2021.102985.

[8] D. Soukup *et al.*, "Machine learning metrics for network datasets evaluation," in *ICT Systems Security and Privacy Protection*. Springer Nature Switzerland, 2024, pp. 307–320, ISBN: 9783031563263. DOI: 10.1007/978-3-031-56326-3_22.

[9] M. Douze *et al.*, *The faiss library*, Sep. 6, 2024. DOI: 10.48550/arXiv.2401.08281. [Online]. Available: http://arxiv.org/abs/2401.08281 (visited on 12/09/2024).

[10] A. Habibi Lashkari *et al.*, "Characterization of tor traffic using time based features:" in *Proceedings of the 3rd International Conference on Information Systems Security and Privacy*, Porto, Portugal: SCITEPRESS - Science and Technology Publications, 2017, pp. 253–262, ISBN: 978-989-758-209-7. DOI: 10.5220/0006105602530262.

[11] J. Dai *et al.*, "GLADS: A global-local attention data selection model for multimodal multitask encrypted traffic classification of IoT," *Computer Networks*, vol. 225, p. 109 652, Apr. 1, 2023, ISSN: 1389-1286. DOI: 10.1016/j.comnet.2023.109652.

[12] W. Wang *et al.*, "Malware traffic classification using convolutional neural network for representation learning," in *2017 International Conference on Information Networking (ICOIN)*, Jan. 2017, pp. 712–717. DOI: 10.1109/ICOIN.2017.7899588.

[13] Y. Hu *et al.*, "Effect analysis of malicious flow classification model based on representation learning on network flow anomaly detection," 2024.

[14] S. Jorgensen *et al.*, "Extensible machine learning for encrypted network traffic application labeling via uncertainty quantification," *IEEE Transactions on Artificial Intelligence*, vol. 5, no. 1, pp. 420–433, Jan. 2024, ISSN: 2691-4581. DOI: 10.1109/TAI.2023.3244168.

[15] R. J. Babaria *et al.*, *FastFlow: Early yet robust network flow classification using the minimal number of time-series packets*, Apr. 2, 2025. DOI: 10.1145/3727115. [Online]. Available: http://arxiv.org/abs/2504.02174 (visited on 04/17/2025).

[16] A. Pektaş *et al.*, "Deep learning to detect botnet via network flow summaries," *Neural Computing and Applications*, vol. 31, no. 11, pp. 8021–8033, Nov. 1, 2019, ISSN: 1433-3058. DOI: 10.1007/s00521-018-3595-x.

[17] I. Vaccari *et al.*, *MQTTset*, 2020. [Online]. Available: https://www.kaggle.com/datasets/cnrieiit/mqttset (visited on 07/19/2024).

[18] U. Do *et al.*, "Evaluation of XAI algorithms in IoT traffic anomaly detection," in *2024 International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*, Feb. 2024, pp. 669–674. DOI: 10.1109/ICAIIC60209.2024.10463357.

[19] A. Nascita *et al.*, "Improving performance, reliability, and feasibility in multimodal multitask traffic classification with XAI," *IEEE Transactions on Network and Service Management*, vol. 20, no. 2, pp. 1267–1289, Jun. 2023, ISSN: 1932-4537. DOI: 10.1109/TNSM.2023.3246794.

[20] M. MontazeriShatoori *et al.*, "Detection of DoH tunnels using time-series classification of encrypted traffic," in *DASC/PiCom/CBDCom/CyberSciTech*, Aug. 2020, pp. 63–70. DOI: 10.1109/DASC-PICom-CBDCom-CyberSciTech49142.2020.00026.

[21] M. T. Jafar, "Analysis and investigation of malicious DNS queries using CIRA-CIC-DoHBrw-2020 dataset," *Manchester Journal of Artificial Intelligence and Applied Sciences*, vol. 2, no. 1, Apr. 26, 2021, ISSN: 2634-1034.

[22] Y. N. Kunang *et al.*, "Attack classification of an intrusion detection system using deep learning and hyperparameter optimization," *Journal of Information Security and Applications*, vol. 58, p. 102 804, May 1, 2021, ISSN: 2214-2126. DOI: 10.1016/j.jisa.2021.102804.

[23] D. Arp *et al.*, "Dos and don'ts of machine learning in computer security," presented at the 31st USENIX Security Symposium (USENIX Security 22), 2022, pp. 3971–3988, ISBN: 978-1-939133-31-1.