Active Learning Framework For Long-term Network Traffic Classification

Jaroslav Pešek
CTU in Prague
Thakurova 9, Prague, Czech Republic
jaroslav.pesek@fit.cvut.cz

Dominik Soukup
CTU in Prague
Thakurova 9, Prague, Czech Republic
soukudom@fit.cvut.cz

Tomáš Čejka
CESNET
Zikova 4 Prague, Czech Republic
cejkat@cesnet.cz

2023 IEEE 13th Annual Computing and Communication Workshop and Conference (CCWC) | 979-8-3503-3286-5/23/$31.00 ©2023 IEEE | DOI: 10.1109/CCWC57344.2023.10099065

***Abstract*—Recent network traffic classification methods benefit from machine learning (ML) technology. However, there are many challenges due to the use of ML, such as lack of high-quality annotated datasets, data drifts and other effects causing aging of datasets and ML models, high volumes of network traffic, etc. This paper presents the benefits of augmenting traditional workflows of ML training&deployment and adaption of the Active Learning (AL) concept on network traffic analysis. The paper proposes a novel Active Learning Framework (ALF) to address this topic. ALF provides prepared software components that can be used to deploy an AL loop and maintain an ALF instance that continuously evolves a dataset and ML model automatically. Moreover, ALF includes monitoring, datasets quality evaluation, and optimization capabilities that enhance the current state of the art in the AL domain. The resulting solution is deployable for IP flow-based analysis of high-speed (100 Gb/s) networks, where it was evaluated for more than eight months. Additional use cases were evaluated on publicly available datasets.**

***Index Terms*—Active Learning, Dataset Quality, Network traffic analysis.**

I. Introduction

Machine learning (ML) is modern technology, and researchers worldwide show its feasibility in many domains, including network traffic monitoring and analysis. The growing amount of end-to-end encryption in network traffic forces us to leave traditional methods of traffic analysis based on visibility into the Application layer (L7) (as it was feasible in the past, e.g., in [1]). Data decryption is not feasible in practice in terms of scale and flexibility. ML is a promising way to exploit advanced statistics derived from encrypted communication flows even on high speed communication links [2].

On the other hand, ML is dependent on the target domain and dataset that is used for training. In most cases, the dataset must be annotated for network traffic classification (for supervised ML). This requirement makes ML usage very complicated since we need to provide ground truth labels. Without reliably annotated datasets, it is not possible to gain relevant results from ML models.

It is common that the research experiments for scientific papers are performed using a fixed dataset, i.e., from one fixed (even short) period of time, which works perfectly in laboratory environments. However, network traffic evolves, as it is studied by Brabec et al. in [3], and we can argue that ML models and datasets are becoming obsolete in time [4]. Also, deployment in a different network can cause a performance decrease. As a result, even a perfectly annotated dataset with good performance will drop in accuracy due to aging. Therefore, we recognize an essential requirement to research methods for autonomous ML deployment, monitoring, and regular updates over time.

Fortunately, there is a concept of Active Learning (AL) as a sub-field of ML to deal with a huge amount of incoming data. The aim of AL is to reduce the initial need for labeled data records by intelligent querying labels during the training phase. This approach can continuously add new data instances and build relevant, up-to-date datasets for any target domain.

AL is defined as a general concept that leaves a wide space for methods and possibilities of how entities of AL are used (annotator, query strategy, ML model, and input data). Many state of the art approaches are incomplete or unavailable for our network traffic analysis use case; thus, some implemented system is missing for practical deployment in production. Therefore, we propose an Active Learning Framework (ALF) [1] focused primarily on network traffic analysis and designed for stream-based processing that allows online or offline dataset evaluation.

This paper describes the following main contributions:

- We propose a publicly available Active Learning Framework prototype — novel methods with a flexible and modular framework for network traffic classification:
  - based on Active Learning technology and state of the art principles and methods;
  - allowing for autonomous and continuous dataset creation and evolution;
  - supporting automatic updates, evaluation, and optimization of annotated datasets of network traffic;
  - supporting extensive monitoring of ML models and all stages in ALF workflow.
- Using ALF, we evaluate different AL query strategies during dataset updates and present new findings contrary to related work.
- We propose more types of annotators to enhance the annotation process within ALF.

979-8-3503-3286-5/23/$31.00 ©2023 IEEE

[1] https://github.com/CESNET/ALF

- We showcase experiments on publicly available datasets and on long-term deployment in the real network to prove the feasibility of ALF.

This paper is a broad extension of our previous work [5], which provided initial ideas for the quality of datasets and ML model deployment.

The organization of this work is as follows. Sec. II lists related works in active learning, quality of datasets, and related solutions. Sec. III contains details about the architecture and main features of the ALF system. Sec. IV lists the most common use cases in the networking domain we target and evaluation of ALF. Sec. V presents results we have achieved in a real network environment in online mode. Finally, Sec. VI contains identified findings from our experiments and Sec. VII concludes the paper.

II. Related Work

The autonomous creation of a dataset is a very challenging task for researchers in any domain due to the complicated labeling of a gathered dataset. One possible solution is semi-supervised learning which does not require labels for all items. This approach is demonstrated, for example, by Iliyasu et al. [6], who applied it to a classification task for encrypted network traffic. AL is a universal concept that helps build relevant datasets with labeled data. In this paper, we are focused on the evaluation of AL in the network traffic domain and the enhancement of the current state of the art regardless of supervised, semi-supervised, or unsupervised ML models.

Settles [7] provided the first comprehensive survey in this field. This survey includes available methods and scenarios that can be applied with AL. The author discusses a pool-based resp. stream-based approaches to AL, which are designed for processing an offline batch of data, resp. online continuous stream of data. Even though pool-based approaches are commonly chosen by researchers, the author recommended stream-based approaches as more appropriate for cases like ours. He also described basic AL query strategies (i.e., methods to select which data to annotate), and three of them are relevant for stream-based use cases (uncertainty sampling, information density, and random sampling). Recently, a new technical study regarding AL-based methods for network traffic classification has been published by Shahraki et al. [8]. This survey contains a summary of used AL methods in the network traffic domain. The authors reviewed several query strategies and empirically tested and evaluated them. Based on this study, we extended the set of query strategies with the Reinforcement AL (RAL) method and tested it in selected scenarios. Ju et al. [9] used the AL method together with concept drift. Wasserman et al. [10] introduced the RAL strategy, which utilizes feedback from an annotator. Cardoso et al. [11] define ranked batch for removing duplicates (in the sense of distance similarity) in a batch of selected flows in one iteration. The results of these papers were considered during AL evaluation on publicly available networking datasets in Sec. V.

Ginart et al. [12] introduced the need for post-deployment ML monitoring. This work proposes a new method with theoretical guarantees to detect changes in the input data stream and ML results, such as distribution or data drift. ML model monitoring is part of our framework too. Moreover, it is connected with AL. Therefore, it leverages information from annotators, dataset quality method AL loop, and ML model to automatically improve the dataset based on detected changes.

We delivered a universal and flexible framework, ALF, that implements known AL query strategies from relevant papers and combines state of the art knowledge referenced in the previous sections. We implemented strategies described by Settles and Shahraki et al. Based on our experiments, better results are obtained with a batch of flows. For informative selection in batch, we implemented Cardoso's ranked batch. We also implemented Wassermann's RAL with a referential implementation with non-significant changes. Principles introduced by Wassermann's RAL might be used in the future with more strategies. The AL core is extended by optimization and monitoring modules to allow post-deployment improvements of the dataset and ML model.

To our best knowledge, there is no publicly available software implementation of active learning technology applied and optimized for i) network traffic classification, ii) continuous evolving annotated network traffic datasets for ML, iii) support of research in dataset evaluation and optimization. Such ambitions were set for the design and development of the proposed ALF described in our paper.

III. Active Learning Framework (ALF)

In this work, we propose a framework that aims to improve the deployment and monitoring of ML models in real networks. This section provides a more detailed description of its design and benefits. Also, we compare our solutions with existing frameworks.

A. Overview

ALF is designed as a modular framework that can be completely customized. Generally, it provides an AL workflow that works with any input data format and ML model. However, this paper is mostly focused on the network traffic domain. As an input, we expect an initial dataset ($\mathcal{D}_0$) that can be even empty, annotator, target ML classifier, and unlabeled data. Based on the provided configuration, the framework can work in offline (pool-based) or online (stream-based) mode. This brings flexibility since we can process network data directly from the network interface or load stored traffic in pcap format. More details of data processing steps are described in Sec. III-B.

Dataset quality assessment is based on permutation testing introduced by Camacho et al. [13], and Wasielewska et al. [14]. It allows to evaluate the quality of each dataset version and compares the improvement for possible retraining of the classifier. The standard AL approach is focused on enhancing an existing dataset with new valuable data. Nevertheless, this would incrementally increase the size of the dataset, its redundancy, and its complexity, regardless of its quality.

ALF uses the concept of annotators to standardize the process of labeling. Generally, an annotator is a software module that is able to annotate a subset of input data (network flows in our case) based on some additional information from external sources (e.g., system logs, service logs, or any monitoring/auditing tools running at endpoint devices). Naturally, having a perfect annotator seems like an ideal solution to the whole classification task. However, in practice, the annotator is not always available for all cases and cannot be deployed to process the whole traffic. Therefore, it is assumed annotator is not able to provide ground truth to all data, and some AL query strategy is required to select the subset that can be and should be annotated. For example, an annotator based on information from endpoints cannot be used in the network environment with devices out of control. The use of decryption to annotate the traffic is not feasible at a large scale. Our idea is to train the ML model using an annotator in some controlled environment and, afterward, deploy and update classification models in production. This approach allows automated and consistent labeling that AL methods can leverage. Part of the ALF solution is a publicly available web plugin annotator [2] created during development that is able to annotate TLS traffic based on the browser information.

Collected metrics from all stages are used for monitoring classifier and dataset performance over time. These metrics include Matthews correlation coefficient, F_1 score, accuracy, recall, precision, and dataset quality score [14]. These values are visualized for different query strategies over time to compare.

From the technical point of the overview, ALF has been developed in Python language and was primarily intended for any UNIX-based operation system. The whole framework is implemented as a single Python module with several classes. These classes are cross-referenced and exchange data via class attributes. Based on this approach, ALF can be used as a standalone module but can also be integrated with other frameworks, e.g., NEMEA[3] [1] is a working example.

B. Architecture

The architecture of an Active Learning Framework for network classification includes several key components:

1) IP Flow dataset $\mathcal{D}_i$ is the set of IP flows that are captured and labeled. The data may include information such as source and destination IP addresses, packet size, traffic volume, etc. Preprocessing of input data for ML classification is part of ALF configuration.
2) The machine learning model is trained on $\mathcal{D}_i$ dataset of labeled IP flows and is used to predict the class of new, unlabeled flows. In our case, we employed exclusively supervised learning algorithms. Once new flows have been obtained, the model is retrained with the newly labeled data, and the process continues.

[2]https://github.com/jan-kala/WebTrafficAnnotator

[3]NEMEA system is an open source IP flow based tool for stream-wise analysis of network traffic.

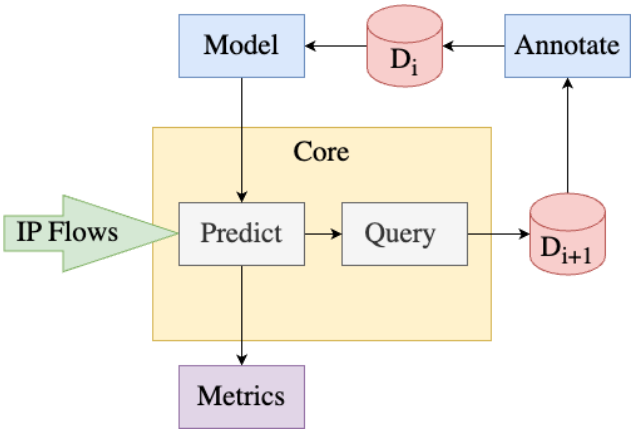

Fig. 1: ALF architecture. The "Core" part is sequential internally, otherwise the architecture is asynchronous. In the case of a cold start, predict part is skipped, and the query is required to be random.

3) Query strategy is used to select which new, unlabeled flows to annotate with labels. This can be done using a variety of methods, such as uncertainty based, random, or reinforcement learning based.
4) Annotator provides a process of annotating new, unlabeled flows with labels, typically done by human experts. ALF is ready to work with various types of annotators based on configuration. The annotator can be a direct part of ALF, or it can be hosted on a remote endpoint or server.
   In our architecture, we had to deal with the cold start problem. This is the situation in that we have no initialization dataset available. In the case where there is no initial dataset, the process starts with annotating random flows until a satisfying number of flows is collected, and then the model is trained with that labeled dataset. During the cold start phase, no model and no prediction are used. The architecture is displayed in Fig. 1.
5) The postprocessing provides an operation over the training dataset. On the one hand, there is undersampling to reduce the $\mathcal{D}_i$ size, and on the other hand, this module checks the quality of the dataset for which permutation tests are employed.

C. Existing AL Frameworks

During the development of the proposed framework, we evaluated existing solutions to compare improvements and novel features. Results are part of Tab. I. In general, most available frameworks implement AL query strategies based on the available state of the art for the particular domain. However, almost none of these frameworks include novel directions that are important for the autonomous creation of datasets and deployment of ML models in the real environment. This includes the evaluation of the Quality of Datasets (QoD), which is important in the postprocessing phase of the AL loop. Several frameworks contain support for ML performance

TABLE I: Comparison with existing AL solutions in terms of features for network traffic classification environment.

AL Framework	Stream-based	Pool-based	QoD	Failover Ready	Monitoring/Metrics	Network Traffic Evaluation
ALF	✓	✓	✓	✓	✓/✓	✓
modAL [15]	✓	✓	✗	✗	✗/✗	✗
ALiPy [16]	✗	✓	✗	✓	✗/✓	✗
AlpacaTag [17]	✗	✓	✗	✗	✗/✓	✗
Baal [18]	✗	✓	✗	✗	✗/✓	✗
Libact [19]	✗	✓	✗	✗	✗/✗	✗

metrics for evaluation and visualization of experiment results. Nevertheless, more detailed and continuous information about the dataset, AL iterations, and framework logs for monitoring over time is missing. Most frameworks are ML agnostic, so available query strategies can be used for any use case. However, our framework is the only one that has been tested and designed for network traffic classification, including novel features. Lastly, AL is a continuous service that should allow restarts and recovery from any processing phase. This is also one of the enhancements we focus on during development.

IV. Targeted Real Use Cases

ALF already contains novel features that are important for the network traffic environment. This section describes the use cases we target that help evaluate the proposed framework and its features.

A. Classification Improvement (even from the empty dataset)

ALF instance is able to run even in such a configuration with an empty or insufficiently small initial dataset. An annotator with a suitable query strategy then works as a generator of a completely new initial dataset inside ALF. With this setup, it is possible to generate annotated datasets automatically from the real network traffic, incrementally update them, and evolve ML models. This functionality saves a significant amount of time and trust in created datasets. Also, with this deployment, we can quickly identify if the classification task is suitable for the ML algorithm with selected features and acceptable performance or more detailed investigation is needed.

B. ML Model Development

It is generally complicated to build a precise ML model and keep it updated for a long time. ALF can be used as an annotator and build an ML model quickly with the low effort since we assume the query strategy selects the most informative samples. This helps to keep the dataset up to date over time. With continuous monitoring and dataset quality evaluation, we can track evolving parameters of the ML model and dataset. In future research, we would like to focus more on monitoring metrics to provide more detailed insights that will help to optimize AL and ML parameters.

C. Dataset optimization

Many publicly available datasets are focused on their large size, which is considered as a quality parameter. However, in a real deployment, it is preferred to have a minimal size of the dataset that provides the same performance score as the original dataset [5]. Using the query strategy, we can start with the original dataset and select the most informative items to create the new dataset. Once the performance score of the ML model is the same as for the original dataset, we have the optimal dataset size for the selected use case.

TABLE II: P-value tables for merged DoH datasets. P-values above significance level 0.01 are marked in red, and lower p-values are replaced by a dot as described in [13]. The results confirm the good quality of the dataset even after the merge operation since at least one model is not signification for all permutation levels.

	50%	25%	10%	5%	1%
KNN	.	.	.	.	.
SVM	.	.	.	.	0.18
DT	.	.	.	.	0.20
RF	.	.	.	.	0.10
XGB	.	.	.	.	.

D. Dataset merge

It is very common to have more versions of datasets that could be used for ML detection. However, it is challenging to identify how to merge more datasets to keep the most valuable items. ALF uses query strategies to process each dataset and select the most informative items to build the final dataset.

For dataset merge, we assume cold start, and thus we randomly select samples to $\mathcal{D}_0$ from both datasets. Then we use ALF to select the most informative samples iteratively from both datasets. In this scenario, we do not need an annotator since samples are already annotated.

For a demonstration, we combined two DoH (DNS over HTTPS protocol) datasets of IP flows from two different time periods, i.e., these datasets are different enough yet define the same classification task. To evaluate the quality of the merged dataset, we used a p-value table implemented based on permutations testing [13] that is part of ALF, see Tab. II. The achieved result corresponds with the original datasets.

V. Evaluation in Real Environment

This section describes our experiments that were done to demonstrate selected use cases and the benefits of ALF. New findings and comparison with existing papers are part of Sec. VI.

A. Real Network

ALF framework has already been experimentally deployed for pilot testing on a flow collector in national research and

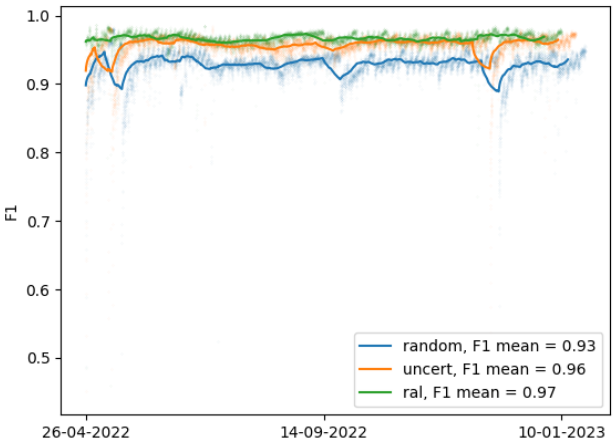

Fig. 2: Comparison of results from three approaches for illustration, showing that the RAL strategy (defined by Wassermann et al. in [10]) outperforms both randomly selecting flow samples and a strategy based on entropy measurement. However, all strategies keep the internal machine learning model performance relatively stable and resistant to data drift. The most stable approach is RAL; across all investigated strategies, the least stable is random.

education. CESNET2[4] network infrastructure. The network has many network lines in the range from 10 Gb/s to 100 Gb/s connecting CESNET2 to other backbone networks. On average, the monitoring probes at the perimeter of the network generate about 300,000 IP flow records per second during peak hours. Naturally, we were forced to sample the flows since we were not able to handle such a high throughput. In upcoming versions, we intend to increase the performance and thus reduce the need for sampling. This pilot deployment improved debugging and revealed new requirements and feature requests.

We selected the DoH protocol classification, which is a protocol for DNS queries encapsulated in the HTTPS protocol. This problem was chosen because deterministic blacklist classification is straightforward and reliable to prove the novel benefits of ALF.

We selected three approaches for best illustration, shown in Fig. 2. We launched the experiments in April 2022 and are tracking the results continuously until January 2023. We can see randomly selecting flow samples leads to significantly worse results. Uncertainty based strategy with entropy measuring leads to much better results. However, the best result we observed came from the RAL strategy defined by Wassermann et al. in [10], which is a state-of-the-art strategy based on principles of reinforcement learning approach. In the experiment, we used undersampling at the end of each AL cycle, and at the same time, we constrained the size of the $\mathcal{D}_i$ to avoid being overwhelmed by the size of the database, which might bring a negative performance effect.

[4]CESNET2 is a national research and education network infrastructure in the Czech Republic

TABLE III: Comparison of various query strategies during the experiments with DeCrypto dataset.

Strategy	Average final F_1	Average query time $[sec/it]$
Unranked uncertainty	0.952	0.001
Ranked uncertainty	0.919	2.116
Unranked density	0.699	2.592
Ranked density	0.642	3.069
KL divergence	0.919	2.820
Random	0.860	0.001

B. Public Dataset DeCrypto

The public dataset DeCrypto [20] is used in [21]. This dataset was created to design a detector of cryptominers' communication. It contains 2,024,903 flows collected on the national CESNET2 network. Results are in Tab. III. We used strategies appropriate for offline experiments thus we omitted RAL since it is designed for long-time stream-wise runs. We notice the poor results for strategies based on information density (Ranked and Unranked density). Uncertainty based strategies are apparently superior to random strategy in this use case.

C. Public Dataset CTU-13

Another battery of experiments was performed on the CTU-13 dataset. The CTU-13 dataset is a publicly available resource for network security research, specifically for intrusion detection systems (IDS) evaluation. The CTU-13 dataset is widely used by researchers in the field of network security and is considered a standard benchmark dataset for evaluating the performance of IDS systems. It contains flows labeled as botnet traffic and benign traffic [22]. We mirrored experiments we did on the DeCrypto dataset. The results (visualized in Fig. 3 with time progression, all methods except random strategy obtained F_1 score approaching 1.0) were similar with one significant difference, which is that similarity based methods were not nearly as unsuccessful. While we did not find any reason why they should be considered relevant, they certainly did not give as weak results as in the previous battery of experiments. The explanation offered is that the botnet is very different from benign traffic and is, therefore, easily separable.

VI. Findings and Experience

Hereby we describe the main findings that emerged from the experiments.

A. Similarity based strategies

Approaches based on similarity, such as ranked batch and information density, are methods of active learning where the model selects the data points for labeling based on their similarity to the existing labeled data. However, it was shown that these approaches do not work well in the practice of classifying IP flows. One reason for this may be that IP flows can be highly diverse and may not be easily grouped into distinct clusters of similar data points. Moreover, these methods are extremely computationally intensive.

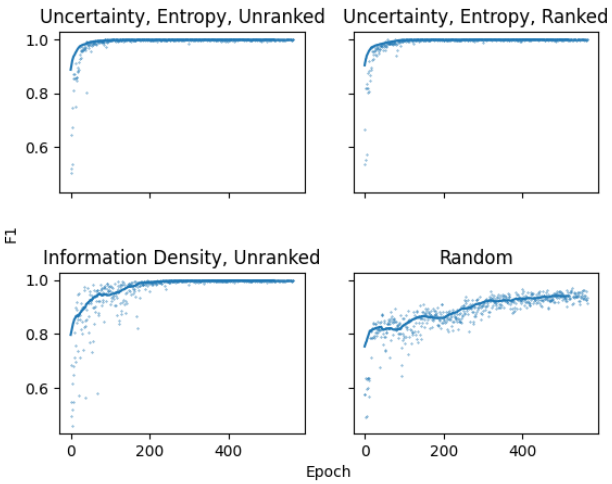

Fig. 3: Comparison of active learning query strategies for IP flow classification: The top-left plot shows the performance of unranked uncertainty sampling with entropy measuring, while the top-right plot shows the performance of ranked uncertainty sampling with entropy measuring. The bottom-left plot shows the performance of the information density strategy, while the bottom-right plot shows the performance of the random strategy as a baseline. The results indicate that uncertainty-based strategies perform best, with the information density strategy also showing good convergence to the optimum, while the random strategy is unstable and does not converge to the optimal performance.

B. Threshold of uncertainty

Another finding is that threshold in uncertainty based approaches, where the model selects data points for labeling based on a score of confidence, have been found to have no significant effect on the performance of the system. This finding is in contrast to the work of Settles [7] and others. Given this, future research might focus on developing alternative methods of selecting data points for labeling that are better suited to the task of classifying IP flows.

C. Random query

A final finding in the field of active learning for network monitoring is that random query strategies, where the model selects data points for labeling randomly, are fast and work sufficiently well but are easily surpassed by uncertainty based strategies such as least confident scoring, entropy scoring, reinforcement learning strategy, etc.

D. Generalization of AL deployment

In conclusion, our experiments have shown that the methods are not universally valid and that the best results are obtained by carefully choosing the appropriate configuration for each use case. We observed that these results might not be generalizable to other situations, and more research is needed to understand the optimal configuration for different use cases. Overall, it is important to keep in mind that the approach and configuration should be chosen carefully to ensure the best results are obtained.

VII. Conclusion

The academic community has known the active learning principle for many years. However, its application in network traffic analysis is not common. Network monitoring and traffic analysis (e.g., for network security purposes) are crucial areas that can benefit from machine learning technology. The active learning approach allows for continuous updates of both datasets and machine learning models. Therefore, we have developed the new Active Learning Framework (ALF) to address the needs related to machine learning applications in the network traffic analysis domain. ALF is designed to process a stream of extended flow data (commonly used in practice to monitor large network infrastructures) and to evolve datasets and machine learning models. Contrary to other AL solutions, ALF includes novel enhancements, such as monitoring and dataset quality capabilities, that are very beneficial for the network traffic classification domain. The proposed enhancements were tested on publicly available datasets and real network traffic over eight months.

The reason for ALF uses instead of some alone annotator lies in the performance limits, computational or time complexity of the annotating (i.e., labeling process), and finally, the possibility to retrieve ground truth information that is not always available for all data in practice. Therefore, ALF is meant to train machine learning models using available information by the annotator so that such models can be deployed in other network environments where no annotator can be used.

We believe the current version of ALF will help to accelerate research activities in areas like quality of dataset assessment, research of replacement strategies and dataset optimization, and so on.

Acknowledgement

This research was funded by the Ministry of the Interior of the Czech Republic, grant No. VJ02010024: Flow-Based Encrypted Traffic Analysis, and by the Ministry of Education, Youth and Sport of the Czech Republic, grant No. LM2023054: e-Infrastructure.

References

[1] T. Cejka, V. Bartos, M. Svepes, Z. Rosa, and H. Kubatova, "Nemea: A framework for network traffic analysis," in *2016 12th International Conference on Network and Service Management (CNSM)*, 2016, pp. 195–201.

[2] Z. Tropková, K. Hynek, and T. Čejka, "Novel https classifier driven by packet bursts, flows, and machine learning," in *2021 17th International Conference on Network and Service Management (CNSM)*, 2021, pp. 345–349.

[3] J. Brabec, T. Komárek, V. Franc, and L. Machlica, "On model evaluation under non-constant class imbalance," *Computational Science – ICCS 2020*, vol. 12140, pp. 74 – 87, 2020.

[4] J. Lu, A. Liu, F. Dong, F. Gu, J. Gama, and G. Zhang, "Learning under concept drift: A review," *IEEE Transactions on Knowledge and Data Engineering*, vol. 31, no. 12, pp. 2346–2363, 2019.

[5] D. Soukup, P. Tisovčík, K. Hynek, and T. Čejka, "Towards evaluating quality of datasets for network traffic domain," in *2021 17th International Conference on Network and Service Management (CNSM)*, 2021, pp. 264–268.

[6] A. S. Iliyasu and H. Deng, "Semi-supervised encrypted traffic classification with deep convolutional generative adversarial networks," *IEEE Access*, vol. 8, pp. 118–126, 2020.
[7] B. Settles, "Active learning literature survey," 2009.
[8] A. Shahraki, M. Abbasi, A. Taherkordi, and A. D. Jurcut, "Active learning for network traffic classification: A technical study," *IEEE Transactions on Cognitive Communications and Networking*, vol. 8, no. 1, pp. 422–439, 2022.
[9] J. Lu, A. Liu, F. Dong, F. Gu, J. Gama, and G. Zhang, "Learning under concept drift: A review," *IEEE Transactions on Knowledge and Data Engineering*, vol. 31, no. 12, pp. 2346–2363, 2019.
[10] S. Wassermann, T. Cuvelier, and P. Casas, "RAL - Improving Stream-Based Active Learning by Reinforcement Learning," in *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML-PKDD) Workshop on Interactive Adaptive Learning (IAL)*. [Online]. Available: https://hal.archives-ouvertes.fr/hal-02265426
[11] T. N. Cardoso, R. M. Silva, S. Canuto, M. M. Moro, and M. A. Gonçalves, "Ranked batch-mode active learning," vol. 379, pp. 313–337. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0020025516313949
[12] T. Ginart, M. Jinye Zhang, and J. Zou, "Mldemon:deployment monitoring for machine learning systems," in *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, G. Camps-Valls, F. J. R. Ruiz, and I. Valera, Eds., vol. 151. PMLR, 28–30 Mar 2022, pp. 3962–3997. [Online]. Available: https://proceedings.mlr.press/v151/ginart22a.html
[13] J. Camacho and K. Wasielewska, "Dataset quality assessment in autonomous networks with permutation testing," in *IFIP/IEEE International Symposium on Integrated Network Management (IM)*, 2022.
[14] K. Wasielewska, D. Soukup, T. Čejka, and J. Camacho, "Dataset Quality Assessment with Permutation Testing Showcased on Network Traffic Datasets," 6 2022. [Online]. Available: https://www.techrxiv.org/articles/preprint/Dataset_Quality_Assessment_with_Permutation_Testing_Showcased_on_Network_Traffic_Datasets/20145539
[15] T. Danka and P. Horvath, "modAL: A modular active learning framework for Python," available on arXiv at https://arxiv.org/abs/1805.00979. [Online]. Available: https://github.com/modAL-python/modAL
[16] Y.-P. Tang, G.-X. Li, and S.-J. Huang, "Alipy: Active learning in python," 2019. [Online]. Available: https://arxiv.org/abs/1901.03802
[17] B. Y. Lin, D.-H. Lee, F. F. Xu, O. Lan, and X. Ren, "AlpacaTag: An active learning-based crowd annotation framework for sequence tagging," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*. Florence, Italy: Association for Computational Linguistics, Jul. 2019, pp. 58–63. [Online]. Available: https://aclanthology.org/P19-3010
[18] P. Atighehchian, F. Branchaud-Charron, J. Freyberg, R. Pardinas, L. Schell, and G. Pearse, "Baal, a bayesian active learning library," https://github.com/baal-org/baal/, 2022.
[19] Y.-Y. Yang, S.-C. Lee, Y.-A. Chung, T.-E. Wu, S.-A. Chen, and H.-T. Lin, "libact: Pool-based active learning in python," National Taiwan University, Tech. Rep., Oct. 2017, available as arXiv preprint https://arxiv.org/abs/1710.00379. [Online]. Available: https://github.com/ntucllab/libact
[20] R. Plný, K. Hynek, and T. Čejka, "Datasets of cryptomining communication," Oct. 2022, Acknowledgements This research was funded by the Ministry of Interior of the Czech Republic, grant No. VJ02010024: Flow-Based Encrypted Traffic Analysis and also by the Grant Agency of the CTU in Prague, grant No. SGS20/210/OHK3/3T/18 funded by the MEYS of the Czech Republic. [Online]. Available: https://doi.org/10.5281/zenodo.7189293
[21] ——, "DeCrypto: Finding Cryptocurrency Miners on ISP Networks," in *Secure IT Systems*, ser. Lecture Notes in Computer Science, H. P. Reiser and M. Kyas, Eds. Springer International Publishing, pp. 139–158.
[22] S. García, M. Grill, J. Stiborek, and A. Zunino, "An empirical comparison of botnet detection methods," vol. 45, pp. 100–123. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167404814000923