Early Detection of Network Service Degradation: An Intra-Flow Approach

Balint Bicski[*‡] and Adrian Pekar[*†‡]

[*]Department of Networked Systems and Services, Faculty of Electrical Engineering and Informatics,
Budapest University of Technology and Economics, Műegyetem rkp. 3., H-1111 Budapest, Hungary.
[†]HUN-REN-BME Information Systems Research Group, Magyar Tudósok krt. 2, 1117 Budapest, Hungary.
[‡]CUJO LLC, Budapest, Hungary.
Email: balint.bicski@edu.bme.hu, apekar@hit.bme.hu

Abstract—This research presents a novel method for predicting service degradation (SD) in computer networks by leveraging early flow features. Our approach focuses on the observable (O) segments of network flows, particularly analyzing Packet Inter-Arrival Time (PIAT) values and other derived metrics, to infer the behavior of non-observable (NO) segments. Through a preliminary evaluation, we identify an optimal O/NO split threshold of 10 observed delay samples, balancing prediction accuracy and resource utilization. Evaluating models including Logistic Regression, XGBoost, and Multi-Layer Perceptron, we find XGBoost outperforms others, achieving an F_1-score of 0.74, balanced accuracy of 0.84, and AUROC of 0.97. Our findings highlight the effectiveness of incorporating early flow features and the potential of our method to offer a practical solution for monitoring network traffic in resource-constrained environments.

Index Terms—Service degradation detection, Intra-flow analysis, Early flow features, Hardware offloading, Predictive modeling

I. INTRODUCTION

In the era of relentless digitization, residential networks are fundamental to digital activities, from streaming videos to real-time gaming. As the demand for uninterrupted, high-quality digital experiences grows, so does the need for resilient networks that can consistently meet these expectations [1], [2]. However, service degradation (SD), characterized by reduced network performance, poses a significant challenge. Factors like network congestion, inefficient data routing, or external interferences can lead to increased latency, buffering, or even outages [3]. Therefore, effective SD management is crucial for seamless digital experiences.

Research on SD detection has evolved over nearly two decades, exploring various metrics and methods. Early work analyzed round-trip time deviations to predict Internet SDs [4]. Subsequent studies have utilized end-to-end delay and loss measurements to assess network quality [5] and developed real-time packet loss monitoring systems [6]. Machine learning approaches have also been explored; for example, deep neural networks have been used to predict throughput and flow duration in campus networks [7], highlighting their potential in SD prediction.

Latency has been identified as a crucial SD indicator, particularly in applications like video streaming and cloud gaming [8], [9]. In cloud computing environments, methods to detect inter-VM interference and estimate performance degradation using multi-variable regression models have been proposed [10]. Additionally, controlled SD strategies have been introduced to manage network overloads, balancing service quality and resource utilization [11], [12].

Our research builds on these insights by focusing on SD detection in residential LAN environments, where network devices often have limited computational capabilities [13]. These devices typically use a dual-path architecture: a fast path for rapid hardware-based forwarding and a slow path for CPU-based processing [13]–[15]. Once specific criteria, like the completion of a TCP handshake, are met, flows transition from the CPU (slow path) to hardware (fast path), reducing the CPU workload but limiting detailed, packet-level monitoring [15]. Consequently, flows are split into observable (O) and non-observable (NO) segments, presenting a challenge in detecting SD events in the NO segments after offloading to hardware accelerators.

To address this challenge, we propose leveraging early flow features from the O segments to predict the status of the NO segments, thereby identifying potential SD in residential networks. Our approach, termed *intra-flow service degradation detection*, uses early flow characteristics, such as Packet Inter-Arrival Time (PIAT), to gain insights into network performance. We evaluate various models, including Logistic Regression, XGBoost, and Multi-Layer Perceptron (MLP), to determine their effectiveness in predicting SD events from the O segments. Results show that XGBoost, with an O/NO split threshold of 10 observed delays, offers the best performance—achieving an F_1-score of 0.74, balanced accuracy of 0.84, and AUROC of 0.97—balancing prediction accuracy with resource utilization.

Our main contributions in this paper are as follows:

- A novel approach for predicting SD in network flows using early flow features to infer behavior in NO parts.
- Introduction and evaluation of intra-flow SD detection, leveraging information within a flow for prediction in resource-constrained environments.
- Identification of an optimal O/NO split threshold, balancing prediction accuracy and practical implementation in network devices with hardware offloading.
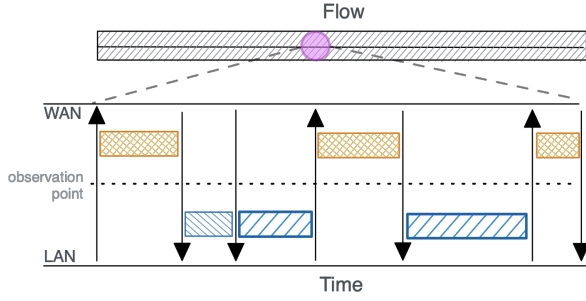
Fig. 1: Visual representation of network traffic flow showing the vertical separation between WAN and LAN delays.

Fig. 2: Illustration of the horizontal separation of a network flow into O and NO segments.

The rest of this paper is organized as follows: Section II presents our method to identify SD events. Section III evaluates the predictive power of O parts of network flows in a binary classification setting. Section IV concludes the paper.

II. METHODOLOGY

A. Vertical Separation of Delays

Network flow packets can be categorized into LAN and WAN directions, with PIATs measured for each. PIAT quantifies the time interval between the arrivals of two consecutive packets within a network flow at the flow meter. We derive LAN-side delays (and jitter) from this metric that serve as foundational metrics of our methodology [16]. The concept of *vertical separation* focuses on isolating LAN-side delays to accurately assess local network conditions, avoiding variability from WAN influences. Fig. 1 illustrates this separation, representing packet directionality as arrows and showing LAN PIATs in blue and WAN delays in orange.

Assuming that delays caused by local endpoints are negligible compared to those induced by broader network conditions and that the response to the last packet in a burst is instantaneous (a condition typically met with TCP traffic) the LAN PIATs marked with a thick blue border in Fig. 1 correspond to the latency introduced by traversing the LAN, thus defining the LAN-side delays that we use in our analysis.

B. Horizontal Separation of Flows

Flows in a network can be split into *Observable* (O) and *Non-observable* (NO) segments based on a threshold θ, which defines the number of packets monitored by the software layer before offloading to hardware. If a flow's packet count p is:

$$\text{State of Flow} = \begin{cases} \text{Observable} & \text{if } p \leq \theta \\ \text{Non-observable} & \text{if } p > \theta \end{cases}$$

For $||f_i|| \leq \theta$, all packets are observed; for $||f_i|| > \theta$, the flow is divided into $O_p = \{p_1, \ldots, p_\theta\}$ (observed) and $NO_p = \{p_{\theta+1}, \ldots, p_{||f_i||}\}$ (unobserved). This *horizontal separation*, visualized in Fig. 2, highlights the point where a flow transitions from software to hardware processing, marked by the threshold θ.
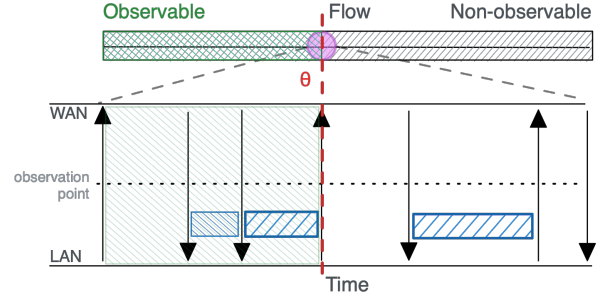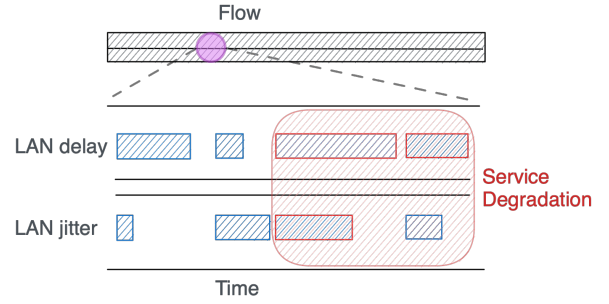

Fig. 3: An SD event in a flow with a MSL of 2.

C. Refining Observability of LAN Delays

The θ parameter does not ensure a consistent number of observable LAN delays due to the exclusion of WAN delays and the occurrence of packet bursts. To refine the focus on LAN delays, we introduce a threshold m specific to LAN delays. The O part of the flow consists of LAN delays up to m, while delays beyond this are considered non-observable. If a flow has fewer delays than m, it is fully observable.

D. Heuristic Detection of Service Degradation Events

With refined observability, SD events can be detected using a heuristic based on latency and jitter behavior over time [16]. Different application types have unique requirements for extreme delay and jitter occurrences, defined by a *Minimum Sequence Length* (MSL) of such events. An SD event begins with an extreme delay and jitter followed by a sequence of extreme delays. Fig. 3 shows an SD event with MSL of 2.

E. Handling Split SD Events

SD events can span the O and NO parts of flows, requiring careful handling to maintain detection accuracy. Three scenarios arise during horizontal separation:

- The SD event is entirely within the O part ($SD_{\text{end}} < O_{\text{end}}$).
- The SD event is entirely within the NO part ($SD_{\text{start}} > O_{\text{end}}$).
- The SD event splits between O and NO parts ($SD_{\text{start}} < O_{\text{end}}$ and $SD_{\text{end}} > O_{\text{end}}$).
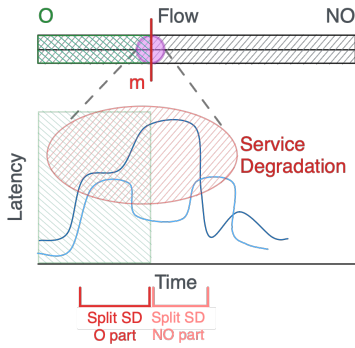
Fig. 4: Illustration of a split SD event in a flow.

Fig. 6: Intra-flow SD detection.

In the last case, in order to preserve potentially crucial information we chose to keep these events by splitting them in two. Fig. 4 illustrates this scenario.

However, this process makes these split events indistinguishable from apparent SD events that are shorter than MSL (i.e. they do not develop into real SD events) and either cross the O/NO boundary or end right at the end of the O part. Such a split is depicted in Fig. 5. Aiming for consistency, we marked these the same way as real split SD events.
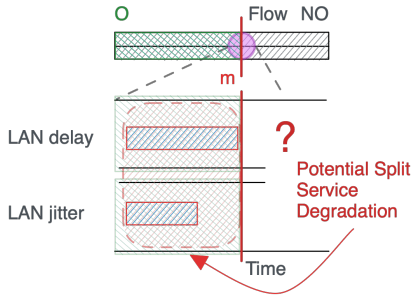

Fig. 5: Illustration of a potential split SD event.

To analyze the splits, we calculate the *split SD ratio* as:

$$f_{\text{split SD ratio}} = \frac{f_{|\text{partial SD event}|}}{f_{\text{MSL}}},$$

where f is a flow. This ratio helps quantify how close a partial event is to meeting the SD event criteria.

F. Intra-Flow Service Degradation Detection

Accurate SD detection in resource-constrained edge devices requires leveraging the observable (O) flow segments to infer the state of the non-observable (NO) segments. We propose an *intra-flow service degradation detection* method that uses early flow features from the O part to predict SD events in the NO part. Fig. 6 illustrates this approach.

The goal is to perform a binary classification using the O segment data to predict SD presence in the NO segment, offering a solution tailored to the limitations of edge devices that cannot continuously monitor entire flows.
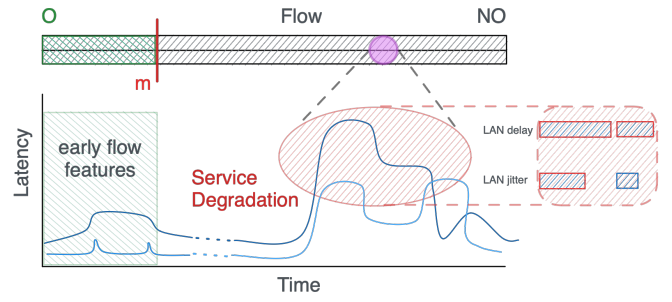
III. PRELIMINARY EVALUATION AND PERFORMANCE ANALYSIS

We assess the potential of the O parts of flows to predict SD events in the NO parts using binary classification models to determine SD event presence or absence in the NO part.

A. Dataset

We use the dataset from [16], [17], which consists of TCP network flows from five consecutive days in a university dormitory network. The dataset includes:

- Aggregated flow features and packet-level characteristics for the first 255 packets of each flow.
- LAN delay and jitter values extracted using vertical separation (Section II-A).
- Marked SD events with application-specific MSL thresholds.

The analysis focuses on flows measured at *Location 2* for result transferability.

B. Data Preparation

Data from Monday to Wednesday was used for training, and Thursday to Friday for testing. Only flows with an O/NO split were included. Input features include:

- Statistical measures (min, max, median, mean, std) of delays and jitters in the O part.
- Individual delay and jitter values, SD event counts, and attributes of the longest SD event in the O part.
- Application, category, location, connection type, and $f_{\text{split SD ratio}}$.

We performed one-hot encoding on categorical variables and used Standard Scaler for scaling. Table I summarizes the input sizes and feature counts for different O/NO splits.

C. Experimental Design and Evaluation Metrics

We evaluated the following models:

- Baseline predictors (null, all-true, random),
- Heuristic models (SD-Based Predictor, Split SD Metric-Based Predictor),
- Machine learning models (Logistic Regression, XGBoost, Multi-Layer Perceptron (MLP)).

Grid Search and 5-fold cross-validation were used to optimize model parameters. Performance was evaluated using

TABLE I: Train and Test Sizes with Input Feature Counts for Different O/NO Splits

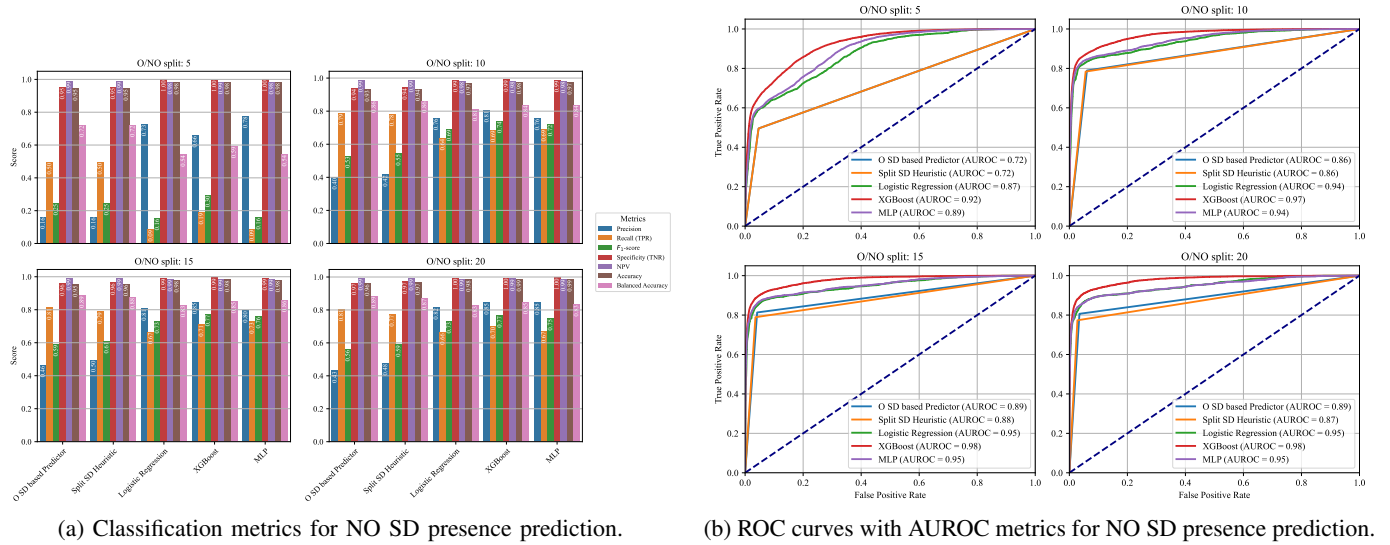| O/NO Split	Train Size	Test Size	Input Feature Count
5	905,407 flows	507,359 flows	141 features (including 6 app. cat. names, 101 app. name, 9 location, 2 connection type OH encoded features)
10	264,183 flows	154,021 flows	140 features (including 6 app. cat. names, 90 app. name, 9 location, 2 connection type OH encoded features)
15	168,463 flows	99,290 flows	144 features (including 6 app. cat. names, 84 app. name, 9 location, 2 connection type OH encoded features)
20	122,814 flows	73,043 flows	152 features (including 6 app. cat. names, 82 app. name, 9 location, 2 connection type OH encoded features)

(a) Classification metrics for NO SD presence prediction.

(b) ROC curves with AUROC metrics for NO SD presence prediction.

Fig. 7: Performance at different O/NO split thresholds.

metrics: *precision*, *recall*, *F_1-score*, *specificity*, *NPV*, *accuracy*, and *balanced accuracy*.

D. Classification Performance

Fig. 7 shows the achieved results for the selected O/NO threshold splits. Fig. 7a compares the classification metrics, while Fig. 7b depicts the Receiver Operating Characteristic (ROC) curves and corresponding areas under the curves (AUROC).

1) Overall Performance: All trained models significantly outperformed the random, all-true, and all-null predictors. The null predictor achieved over 95% accuracy but only 0.5 balanced accuracy, reflecting a high prevalence of flows without SD events in the NO part.

All models at every threshold showed excellent specificity, NPV, and accuracy (over 95%, many approaching 100%), indicating their effectiveness in correctly classifying flows without SD in the NO part. This suggests consistent behavior for flows with no SD events in the O part, nor split SDs, as evidenced by high TNR and NPV values. The high accuracy reflects the data imbalance toward these types of flows. However, models struggled to correctly identify flows with non-observable SDs, where these events were less frequent.

2) Model Comparison: Heuristic models (SD Based Predictor and Split SD Heuristic) showed moderate effectiveness with similar prediction results across all thresholds. They achieved the highest recall and balanced accuracy (50% and 72% at threshold 5, and around 0.8 and over 0.85 at higher thresholds). However, they underperformed in precision and

F_1-score compared to other models, indicating that while many flows with non-observable SD events have SD in their O parts, many flows with SD in the O part or guessed split SD events do not develop SD in the NO part.

The sophisticated models (Logistic Regression, XGBoost, and MLP) had much higher precision but lower recall, especially at threshold 5. For instance, XGBoost reached 66% precision (over 72% for the others) but had a recall of only 19%, and just 9% for Logistic Regression and MLP. This resulted in low F_1-scores (30% for XGBoost and about half for the others) and balanced accuracy around 55-60%, suggesting that lower thresholds do not provide enough information to predict the more volatile behavior in the NO part effectively.

3) Impact of O/NO Split Threshold: At the O/NO split threshold of 10, prediction metrics were more balanced for all models. The heuristics maintained high recall (78%) and balanced accuracy (86%) but had limited improvement in precision (under 45%). The sophisticated models achieved more balanced results, with all metrics above 64%. XGBoost and MLP had a recall of 69% and balanced accuracy of 84%, with XGBoost achieving over 80% precision and the highest F_1-score (74%).

Beyond threshold 10, splits of 15 and 20 showed only marginal gains. XGBoost reached 85% precision and a 77% F_1-score, but further increases led to minor or even decreased improvements, likely due to fewer flows being considered. This indicates that higher thresholds offer no significant advantages.

4) ROC Curve Analysis: The ROC curves show similar behavior to the patterns identified earlier. The simple heuristics demonstrate clear inferiority to the other models, even at the O/NO split threshold of 5 (with AUROC scores of 72% compared to over 87% for the other models). The superior robustness of the XGBoost model is evident at every threshold, exceeding 0.9 in AUROC at the O/NO split of 5, reaching 0.97 by threshold 10 and 0.98 by 15.

Logistic Regression and MLP exhibit similar patterns with highly correlated ROC curves and nearly identical AUROC values. The lack of significant improvement beyond threshold 10 is also evident in this figure.

5) Summary of Findings: All models outperformed the baseline predictors, underscoring the effectiveness of our approach. The XGBoost model offered the best balance between high precision, recall, and balanced accuracy, making it the most suitable for predicting non-observable SD events. Threshold 10 emerged as the optimal O/NO split point, offering a strong trade-off between predictive performance and computational efficiency. Lower thresholds, like 5, lacked sufficient information for reliable predictions, while higher thresholds, such as 15 and 20, provided only marginal improvements, leading to diminishing returns.

6) Limitations: Our study, while promising, has certain limitations. The analysis was conducted on a specific network environment, and further validation in diverse settings would enhance the generalizability of our findings. The accuracy of predictions may vary depending on the nature of the flow and the specific characteristics of SD events. While our model performs well in predicting the presence of SD events, predicting their exact characteristics (such as duration or severity) remains a challenge. Additionally, the method's effectiveness is influenced by the choice of the O/NO split threshold and the features selected for analysis. Despite these limitations, our intra-flow SD detection approach offers a novel and resource-efficient method for maintaining network quality in challenging environments.

IV. Conclusion

This paper introduced a novel method for predicting service degradation (SD) in networks by leveraging early flow features, particularly packet inter-arrival times and related metrics, from the observable (O) segments to infer the behavior in non-observable (NO) segments. We identified an optimal O/NO split threshold of 10 observed delays, which provides a balance between prediction accuracy and resource efficiency.

Our results showed that the XGBoost model performed best, achieving an F_1-score of 0.74, a balanced accuracy of 0.84, and an AUROC of 0.97. These findings underscore the potential of using early flow features for effective SD detection in resource-constrained environments. Future work will explore the extension of this approach to detect other network anomalies and

validate the model across different network environments and traffic types to enhance its robustness and applicability.

References

[1] J. Cito *et al.*, "Identifying root causes of web performance degradation using changepoint analysis," in *Web Engineering*. 2014, pp. 181–199. DOI: 10.1007/978-3-319-08245-5_11.

[2] P. Vuletić *et al.*, "Localization of network service performance degradation in multi-tenant networks," *Computer Networks*, vol. 168, p. 107 050, 2020. DOI: 10.1016/j.comnet.2019.107050.

[3] P. Casas *et al.*, "Unveiling network and service performance degradation in the wild with mplane," *IEEE Communications Magazine*, vol. 54, no. 3, pp. 71–79, 2016. DOI: 10.1109/mcom.2016.7432151.

[4] A. Bremler-Barr *et al.*, "Predicting and bypassing end-to-end internet service degradations," *IEEE Journal on Selected Areas in Communications*, vol. 21, no. 6, pp. 961–978, 2003. DOI: 10.1109/JSAC.2003.814614.

[5] A. Abdelkefi *et al.*, "Assessing the service quality of an internet path through end-to-end measurement," *Computer Networks*, vol. 70, pp. 30–44, 2014. DOI: https://doi.org/10.1016/j.comnet.2014.04.016.

[6] H. Wu *et al.*, "Lossdetection: Real-time packet loss monitoring system for sampled traffic data," *IEEE Trans. Netw. Service Manag.*, vol. 20, no. 1, pp. 30–45, 2023. DOI: 10.1109/TNSM.2022.3203389.

[7] C. Hardegen *et al.*, "Predicting network flow characteristics using deep learning and real-world network traffic," *IEEE Trans. Netw. Service Manag.*, vol. 17, no. 4, pp. 2662–2676, 2020. DOI: 10.1109/TNSM.2020.3025131.

[8] A. Amaral *et al.*, "Prediction of video quality degradation on a cloud gaming platform," in *2023 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)*, 2023, pp. 1–5. DOI: 10.1109/BMSB58369.2023.10211223.

[9] Y. Li *et al.*, "Fleet: Improving quality of experience for low-latency live video streaming," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 33, no. 9, pp. 5242–5256, 2023. DOI: 10.1109/TCSVT.2023.3243901.

[10] L. Pons *et al.*, "Cloud white: Detecting and estimating qos degradation of latency-critical workloads in the public cloud," *Future Generation Computer Systems*, vol. 138, pp. 13–25, 2023. DOI: https://doi.org/10.1016/j.future.2022.08.012.

[11] B. Xian *et al.*, "Prediction-based and provident service degradation in optical data center networks," in *2016 15th International Conference on Optical Communications and Networks (ICOCN)*, 2016, pp. 1–3. DOI: 10.1109/ICOCN.2016.7875717.

[12] W. Hou *et al.*, "Service degradability supported by forecasting system in optical data center networks," *IEEE Systems Journal*, vol. 13, no. 2, pp. 1514–1525, 2019. DOI: 10.1109/JSYST.2018.2821714.

[13] A. Zulfiqar *et al.*, "The slow path needs an accelerator too!" *ACM SIGCOMM Computer Communication Review*, vol. 53, no. 1, pp. 38–47, 2023. DOI: 10.1145/3594255.3594259.

[14] E. C. Molero *et al.*, "Hardware-accelerated network control planes," in *Proceedings of the 17th ACM Workshop on Hot Topics in Networks*, ser. HotNets '18, 2018. DOI: 10.1145/3286062.3286080.

[15] D. Cerovic *et al.*, "Fast packet processing: A survey," *EEE Commun. Surv. Tutor.*, vol. 20, no. 4, pp. 3645–3676, 2018. DOI: 10.1109/comst.2018.2851072.

[16] B. Bicski and A. Pekar, "Unveiling latency-induced service degradation: A methodological approach with dataset," *IEEE Access*, pp. 1–1, 2024. DOI: 10.1109/ACCESS.2024.3456588.

[17] FlowFrontiers, *Servie Degradation Detection - Digital Artifacts*, http://github.com/FlowFrontiers/ServDeg-Dataset, 2024.