On the Feasibility of Inter-Flow Service Degradation Detection

Balint Bicski*‡ and Adrian Pekar*†‡
* 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@cujo.com, apekar@hit.bme.hu

***Abstract*—Hardware acceleration in modern networks creates monitoring blind spots by offloading flows to a non-observable state, hindering real-time service degradation (SD) detection. To address this, we propose and formalize a novel inter-flow correlation framework, built on the hypothesis that observable flows can act as environmental sensors for concurrent, non-observable flows. We conduct a comprehensive statistical analysis of this inter-flow landscape, revealing a fundamental trade-off: while the potential for correlation is vast, the most explicit signals (*i.e.*, co-occurring SD events) are sparse and rarely perfectly align. Critically, however, our analysis shows these signals frequently precede degradation in the target flow, validating the potential for timely detection. We then evaluate the framework using a standard machine learning model. While the model achieves high classification accuracy, a feature-importance analysis reveals it relies primarily on simpler intra-flow features. This key finding demonstrates that harnessing the complex contextual information requires more than simple models. Our work thus provides not only a foundational analysis of the inter-flow problem but also a clear outline for future research into the structure-aware models needed to solve it.**

***Index Terms*—Service degradation detection, Inter-flow analysis, Real-time monitoring, Network correlation, Hardware offloading**

I. Introduction

Service degradation (SD) detection in modern network infrastructure faces a fundamental trade-off between comprehensive monitoring and resource constraints. To achieve line-rate speeds, contemporary network devices employ dual-path architectures where network flows transition from CPU-based processing (the *observable state*) to specialized hardware for accelerated forwarding (the *non-observable state*) [1]. This transition, while essential for performance, offloads the flow from software visibility, creating a critical *monitoring blind spot* where traditional packet-level analysis becomes impossible.

Our previous research established the feasibility of *intra-flow* SD prediction, where a flow's own early characteristics are used to predict the likelihood of its future degradation [2], [3]. While valuable, this approach is fundamentally predictive—yielding only a probability of future issues—and cannot provide *immediate detection* for flows already in their non-observable state, a critical limitation for real-time network management.

This paper proposes an *inter-flow* (or cross-flow) detection method that addresses this gap. Our approach is founded on the principle that network SD often manifests as a system-wide phenomenon, impacting multiple concurrent flows [4], [5]. We hypothesize that by leveraging the temporal correlations between concurrent flows, we can use those in an observable state as environmental sensors to infer the real-time status of flows that are already non-observable.

To test this hypothesis, our work unfolds in two stages. First, we formalize the inter-flow framework and conduct a deep statistical characterization of the underlying data landscape to validate that the potential for correlation is practically significant and timely. Second, we perform a preliminary experimental evaluation by implementing a detection model using a straightforward feature construction method. This allows us to establish a crucial first performance benchmark and, more importantly, to uncover the practical challenges of harnessing this complex, high-dimensional data.

Our main contributions are:

1) The formalization of a novel *inter-flow correlation framework* to address the hardware acceleration monitoring gap.
2) A comprehensive *statistical analysis* that characterizes the inter-flow landscape and validates the potential for timely detection.
3) A preliminary *experimental evaluation* that establishes a performance benchmark and uncovers the limitations of using a simple feature *concatenation* strategy for this task.
4) A clear *roadmap for future research*, emphasizing the need for structure-aware models to unlock the full potential of inter-flow analysis.

Additionally, we complement these contributions with a comprehensive *digital artifact* containing detailed exploratory data analysis, feature engineering pipeline, and extended results.

The remainder of this paper is organized as follows: Section II reviews related work. Section III introduces our framework. Section IV presents the statistical analysis of the inter-flow landscape. Section V evaluates the detection model and discusses its limitations. Finally, Section VI concludes with future work directions.

II. Related Work

While extensive research exists on intra-flow analysis and general network correlation, the specific challenge of using observable flows to detect service degradation in concurrent

978-3-903176-75-1

non-observable flows remains largely unaddressed. Our work builds upon several distinct research areas while addressing this novel intersection.

A. Intra-Flow Service Degradation Detection

The foundation of our work is our previously established methodologies for flow-based SD detection. We define SD as statistically significant increases in LAN-side packet delay and jitter, identified using methods from [2]. Building on this, our prior work demonstrated the feasibility of *intra-flow SD prediction* [3]. In that approach, statistical features (e.g., mean and variance of delay) from a flow's first few observable packets were used to predict the likelihood of future degradation in the same flow, achieving a balanced accuracy of 0.84. However, its critical limitation—and the primary motivation for this paper—is that it is a *predictive* tool, not a real-time detection method. This prior work establishes the clear research gap we aim to fill and serves as the performance benchmark for our new inter-flow strategy.

B. Inter-Flow and Cross-Correlation Analysis

The limitations of single-flow analysis motivate exploring inter-flow dynamics. The concept of using correlation in networking is not new, but has been applied to different problems. For instance, some have correlated control and data plane traffic for security anomaly detection [6]. While effective, their focus on inter-plane correlation is distinct from our flow-to-flow approach. More recently, frameworks like FlowID have used hypergraphs to model inter-flow relationships for the purpose of traffic classification [7]. These studies validate the utility of analyzing relationships between flows, but their objective is classification, not the real-time detection of performance degradation.

C. Conceptual Parallels in Temporal Systems

The principle of leveraging temporal correlations is well-established in other complex domains, providing a strong conceptual parallel to our work. In multi-sensor systems, for example, researchers have constructed temporal correlation graphs between different sensor readings to treat anomaly detection as a graph classification problem, achieving F1-scores exceeding 0.90 [8]. Just as they leverage correlations between heterogeneous sensors to diagnose system-wide anomalies, our work leverages correlations between concurrent network flows to infer the health of the network. This supports our hypothesis that such an approach can reveal degradation patterns that are invisible to single-element analysis [9].

D. The Research Gap: Monitoring in Accelerated Networks

These distinct areas of research converge on a critical, unaddressed problem: the monitoring blind spot created by hardware acceleration in modern networks [10]. Once a flow is offloaded to a hardware path, its real-time performance metrics become invisible to standard monitoring tools like NetFlow/IPFIX [11], [12]. While the techniques above have established intra-flow prediction and explored inter-flow correlation for other tasks, to our knowledge, no prior work has specifically aimed to bridge this hardware-induced monitoring gap by using concurrent observable flows to infer the real-time degradation state of non-observable ones. This is the novel contribution of our paper.

III. Methodology

This section details the framework for our inter-flow analysis. We first explain the core problem of monitoring in hardware-accelerated networks, then define our O/NO segmentation framework, and finally, formalize the temporal correlation approach used to overcome the monitoring challenge.

A. The Hardware Acceleration Blind Spot

Modern network devices, such as residential gateways, rely on hardware acceleration to achieve high-speed packet forwarding. In this common architecture, a network flow's lifecycle is split across two processing paths. When a new flow begins, its first few packets are processed on the device's main CPU. In this *Observable (O) state*, software has full visibility, allowing for deep packet inspection and the measurement of performance metrics like per-packet delay.

To maintain line-rate performance and reduce CPU load, the device's software programs the hardware forwarding engine (the ASIC or "fast path") with a rule to handle all subsequent packets of that flow. The flow then transitions to the *Non-Observable (NO) state*. In this state, packets bypass the CPU entirely and are processed directly in hardware. This creates a critical *monitoring blind spot*: while the flow is active and carrying user traffic, the software has no visibility into its real-time performance. Our work is designed specifically to infer the performance of a flow during this non-observable phase.

B. The O/NO Segmentation Framework

Our work formalizes this process as the O/NO flow segmentation framework, building upon our previous work [3]. The transition from the O to the NO state occurs after a predefined number of a flow's initial delay measurements, m, have been observed. This study uses O/NO splits of $m \in \{5, 10\}$. The inter-flow approach presented here aims to shift from the predictive capabilities of our prior work to a real-time detection method for flows in the NO state.

C. Inter-Flow Correlation Principle

The core assumption driving our methodology is that SD often manifests as a system-wide phenomenon, impacting multiple concurrent flows. When network infrastructure experiences congestion or other bottlenecks, these conditions create detectable temporal correlations between the performance characteristics of flows. Our framework leverages this principle, using concurrent observable flows as distributed environmental "sensors" to infer the state of a non-observable target flow.

Fig. 1 provides a detailed illustration of this concept. A target flow, f_t, enters its non-observable state. The goal is to infer its status by analyzing the features of other flows $(f_a, f_b, \ldots, f_d)$ whose observable segments are active during the non-observable lifetime of f_t.

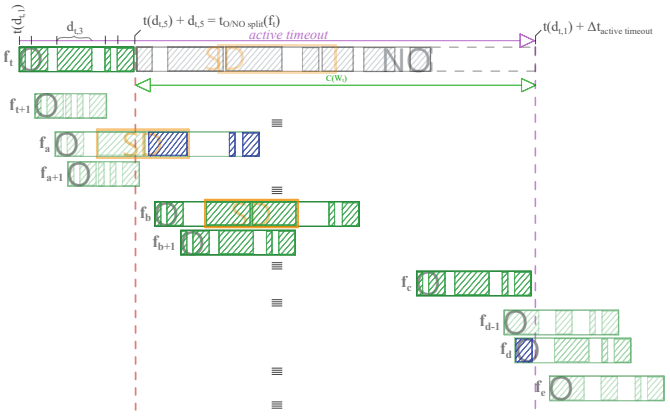


Fig. 1: The inter-flow detection concept: leveraging the observable delay measurements from concurrent flows to infer the state of a target flow in its non-observable phase.

D. Temporal Correlation Framework

We formalize this relationship by defining a *Correlation Window* W_t for each target flow f_t. This window begins the moment f_t transitions to its non-observable state and has a duration $\Delta t = \textit{active timeout} - \Delta t_O$, where Δt_O is the time the target flow spent in its observable state. This definition ensures the window covers the maximum possible lifetime of the flow's non-observable segment before it expires from the flow cache.

A *covering flow*, f_k, is any flow that has at least one observable delay measurement within W_t. To specify this formally, we first define $F_O(W_t)$ as the set of all flows that are in their observable state at some point during the window W_t, and $F_{NO}(W_t)$ as the set of flows in their non-observable state. The correlation space $C(W_t)$ is then defined as the set of all pairs (f_t, f_k) that meet the temporal overlap conditions specified in Equations (1) and (2). Here, $d_{x,y}$ represents the value (duration) of the y^{th} delay measurement of flow f_x, while $t(d_{x,y})$ denotes the timestamp marking the beginning of that delay measurement.

$$C(W_t) = \{(f_t, f_k) : f_t \in F_{NO}(W_t) \ \wedge \ f_k \in F_O(W_t)\} \quad (1)$$

$$\forall k \ \exists i \in \{1,\ 2,\ \ldots,\ m\} \ \wedge \ \exists j \in \{1,\ 2,\ \ldots,\ m\} : \quad (2)$$
$$i \leq j \ \wedge \ t(d_{t,m}) + d_{t,m} \leq t(d_{k,i}) \ \wedge \ t(d_{k,j}) + d_{k,j} \leq t(d_{t,0}) + \Delta t$$

This general framework is designed to be comprehensive. As Fig. 1 illustrates, it accommodates all types of temporal overlaps. For instance, some covering flows like f_a and f_d only *partially overlap* the correlation window. In these cases, our framework considers only the delay measurements that fall within the window (highlighted in blue). Other flows, like f_b and f_c, are *fully contained* within the window. The framework also accounts for *non-contributing* flows, such as f_{t+1} and f_e, which are active but have no observable delay measurements inside W_t.

IV. Inter-Flow Coverage Statistics

Before building a detection model based on the framework in Section III, it is essential to first validate its core assumption: *does the necessary data for inter-flow correlation actually exist in a real-world setting, and is it timely enough to be useful?* This Section addresses these questions by presenting a statistical characterization of the inter-flow landscape. We applied our framework to a large dataset of network flows, using an O/NO split of 10, to characterize the practical properties of the covering flows.

The following analysis is primarily descriptive, designed to characterize the raw data landscape and validate our core assumptions before moving to a predictive modeling phase. Consequently, rather than focusing on specific correlation coefficients such as Pearson or Spearman—which test for linear or monotonic relationships between variables—our analysis centers on the empirical distributions of direct counts and temporal offsets. Furthermore, we use the raw, non-normalized values of the underlying metrics (*e.g.*, per-packet delays) to ensure our characterization reflects the natural scale and distribution of events in the network.

A. Flow Overlap Quantification

First, we sought to understand the volume of potential data available for correlation. Fig. 2 shows the distribution of the number of covering flows per target flow. The opportunity for correlation is immense: a typical target flow is overlapped by thousands of concurrent flows, with a mean of 2836.2. This confirms that the raw material for inter-flow analysis exists in abundance.

However, the figure also reveals a critical challenge: *sparsity of the most explicit signals*. When we filter these covering flows to count only those that also contain a SD event, the mean count drops by an order of magnitude to 290.9. This finding indicates that while the network is dense with activity, explicit degradation events are far less common, suggesting that a successful model cannot rely solely on finding co-occurring SDs.

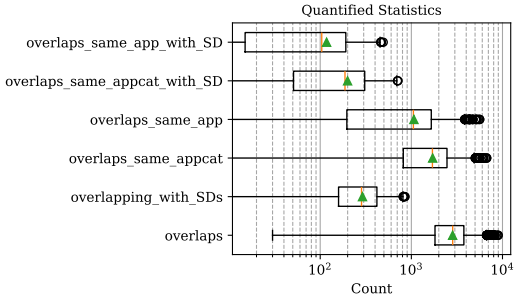


Fig. 2: Distribution of covering flow counts per target flow. The plot shows a high volume of general overlaps but a much lower count for overlaps that also contain SD events, highlighting a signal sparsity challenge.

B. Timeliness of Correlated Signals

Next, we analyzed how quickly this inter-flow information becomes available. Fig. 3 displays the distribution of the time delay from a target flow's O/NO split to the start of the first covering flow. A generic covering flow appears almost immediately, with a mean delay of just 1.43 seconds. This confirms that some form of contextual data is available in near real-time.

The challenge, however, is again revealed when focusing on the most valuable signals. The time to the first covering flow that contains an SD event is significantly longer, with a mean of 98 seconds. For many target flows, which may only last a few minutes, this critical signal could arrive too late to be useful for real-time intervention. This finding strongly motivates the need for a modeling approach that can extract predictive information from all available covering flows, not just the rare ones that also exhibit explicit SD.

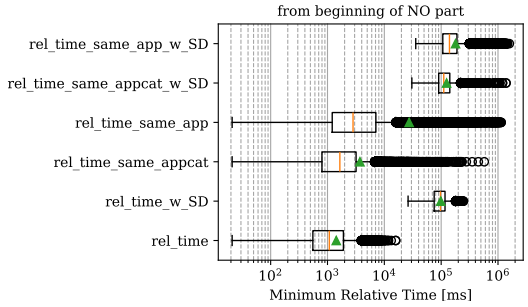


Fig. 3: Distribution of time to the first covering flow. While generic overlaps appear quickly, the first SD-containing overlap is often significantly delayed.

C. Temporal Alignment of SD Events

Despite the challenges of sparsity and delay, our most critical analysis validates the framework's potential for real-time detection by examining the optimal temporal alignment of co-occurring SD events. To quantify this, for each target flow containing one or more SD events, we identified all SD events across all of its covering flows. We then computed the temporal distance between the center-point of every possible target-SD/covering-SD pair. From this set of all pair-wise distances for a given target flow, we selected the single *minimum value* to represent the "best temporal alignment" for that flow. This method ensures we capture the most optimistic case of correlation between any two degradation events.

As shown in Fig. 4, the distribution of these minimum-distance values is notably skewed negative, with a median offset of -66 seconds. This is a powerful finding: it demonstrates that the SD event in a covering flow often *precedes* the event in the target flow, even under the most optimistic alignment. This confirms that the signal in concurrent flows is not just correlated but is often predictive, offering a window of opportunity for timely, or even pre-emptive, detection. This result provides the core empirical justification for our inter-flow framework and motivates the development of a detection model to harness this predictive potential.

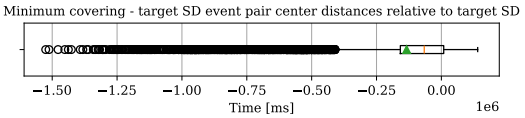


Fig. 4: Distribution of the temporal distance between the center of a target SD event and the center of its best-aligned covering SD event. The negative skew indicates that the covering SD event often precedes the target SD.

V. Experimental Evaluation of an Inter-Flow Model

The statistical analysis in Section IV confirmed that while the data for inter-flow correlation is abundant and holds predictive potential, it is also complex, sparse, and temporally misaligned. Guided by these insights, we designed an experiment to test a specific, pragmatic implementation of our framework. The goal was to assess if a standard machine learning model could harness this challenging data, establishing a performance benchmark and uncovering the practical limitations of a straightforward approach.

A. Experimental Setup

For this initial feasibility study, we made a key design choice to create a tractable and consistent feature space for our models. This choice was directly motivated by the findings of our statistical analysis.

The experiments leverage the large-scale dataset from our prior work [2], which was captured from multiple residential gateways over 5 consecutive days. The ground truth labels, indicating the presence of SD events, were generated using the robust Z-score and IQR-based analysis established in that work. For this study, we created a balanced dataset of 100,000 flows for training and a separate 100,000 for testing.

- *Flow Selection:* While our general framework (Section III) can accommodate all covering flows, for this experiment we constrained the input to manage the high dimensionality revealed in our analysis. For each target flow, we selected only the *first 30 covering flows that were fully contained* within the correlation window. This provides a consistent input vector for the model but knowingly accepts a trade-off, discarding potentially useful data from partial overlaps and later-arriving flows.
- *Feature Vector Construction:* The model input was a single, wide feature vector formed by *concatenating* the target flow's feature set with those of its 30 selected covering flows. The intra-flow set comprised 17 features—delay, jitter, observable SD event statistics, and application/connection type (categorical features one-hot encoded)—plus a dynamic set of delay and jitter metrics based on the O/NO split. Each covering flow contributed the same raw metrics and its relative start time, excluding one-hot encoded application/connection type features.

The resulting vector contained $14+2m+4_{OH\ encoded}+30\cdot(14+2m)$ features, plus covering counts per application type to capture distribution without per-flow one-hot encoding. This yielded 948 features for O/NO split 5 and 1248 for split 10. Detailed cardinalities are provided in the digital artifact [13].

We trained three primary models: a regularized Logistic Regression, a feed-forward Multi-Layer Perceptron (MLP), and a gradient-boosted tree model using the XGBoost library. All models were optimized using a 5-fold cross-validated Grid Search to find the best hyperparameters. Standard scaling was applied on the input features of MLP. This experimental setup was designed to answer a crucial question: can a model architecture that performed well on the simpler, single-flow problem of our previous work [3] adapt to learn from this much more complex, high-dimensional inter-flow feature set?

B. Results and Discussion

We trained several models for two primary tasks: classifying the presence of a SD event, and regressing its specific characteristics. The results present a nuanced picture of success and failure.

For the classification task, the XGBoost model proved to be the most effective. As shown in Fig. 5, it achieved a high AUROC of 0.96 and a strong balanced accuracy of 0.82 for the O/NO split of 10. On the surface, these strong top-line metrics suggest that the inter-flow model is viable and successful.

However, this initial success is misleading. The model's performance collapsed on more granular regression tasks. As detailed in Table I, the errors for predicting SD event characteristics like start and end times were on the order of minutes, with high MAPE values. This stark contrast between classification success and regression failure strongly indicated that the model was not learning the underlying temporal dynamics of the inter-flow data.

TABLE I: Regression metrics for the XGBoost model on flows containing SD events (O/NO=10).

Target Metric	MAE (s)	RMSE (s)	MAPE
SD count	0.48	0.68	0.42
Longest SD length (s)	207.0	309.6	0.79
Longest SD start (s)	98.7	171.4	0.47
Longest SD end (s)	282.3	406.2	0.46

The definitive insight came from a *feature importance analysis* of the successful XGBoost classification model, conducted using SHAP (SHapley Additive exPlanations) shown in Fig. 6. Despite being provided with hundreds of inter-flow features from the 30 covering flows, the model learned to almost completely *ignore them*. Its predictions were overwhelmingly based on the intra-flow features of the target flow itself.

This finding reveals a fundamental mismatch between the problem's complexity and the model's capabilities. By concatenating features, we created an extremely high-dimensional and sparse feature space. A model like XGBoost, which builds decisions by splitting on individual features, struggles to find meaningful relationships across hundreds of weakly correlated inputs. Furthermore, this simple vector provides no structural or temporal information; from the model's perspective, the delay metrics of flow #3 are completely independent of the metrics of flow #4. Lacking this relational context, the model defaults to the features it understands: the strong, familiar, and lower-dimensional signals from the target flow itself.

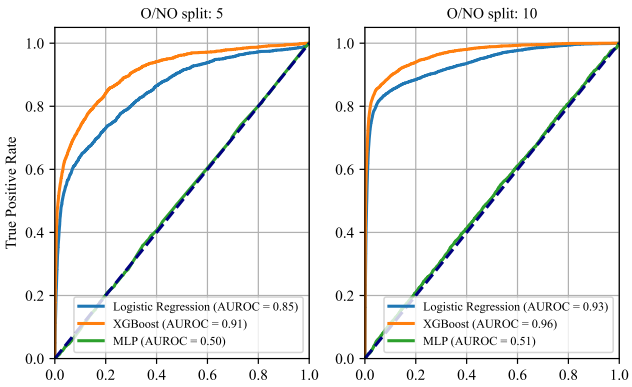
This is the central finding of our experimental evaluation: the model achieved good classification scores by simply replicating the performance of the intra-flow model using the target flow's own data, and it failed at regression because it never learned from the rich contextual data that could have helped characterize the events. The same model architecture that succeeded on the simpler intra-flow problem failed to adapt to the more complex inter-flow task. This does not invalidate our framework's principle, but it proves that a naive feature representation—in this case, simple concatenation—is insufficient for a standard model to learn from the sparse, high-dimensional, and time-shifted context of concurrent flows. This experiment successfully reveals that the primary challenge of inter-flow analysis lies not in the data's existence, but in its effective representation and modeling.

This finding provides a clear roadmap for future research. The logical path forward is to explore architectures inherently designed for relational and temporal data. Graph Neural Networks (GNNs) are particularly well-suited to this problem. A GNN could represent each flow as a node in a dynamic graph, with node attributes defined by the flow's observable features and edges representing temporal proximity or application similarity. Through message passing, a GNN could learn a rich, contextual representation of the network's local state, directly addressing the limitations of the simpler model. Furthermore, while this study was limited to temporal correlations, future work should explore spatial correlation by building graphs that connect flows across multiple network vantage points, a step toward capturing the localized nature of some degradation events.

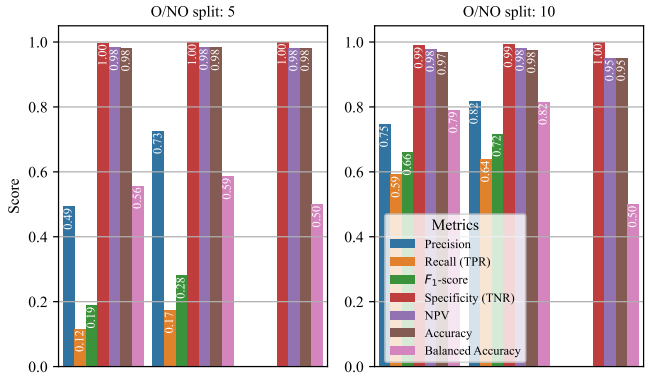
VI. Conclusion and Future Work

In this paper, we introduced a novel inter-flow correlation framework to address monitoring blind spots caused by hardware acceleration. Our primary contribution is a dual finding that bridges theory and practice. First, a comprehensive statistical analysis validated our framework's core premise: a predictive signal for SD does exist in concurrent network flows. Second, our experimental evaluation demonstrated that harnessing this signal is non-trivial. A standard machine learning model, despite its success in simpler intra-flow contexts, failed to leverage the complex inter-flow data when it was presented via simple feature concatenation.

This result crystallizes the true challenge of inter-flow analysis: it is not a data availability problem, but one of feature representation and modeling complexity. Our work therefore provides a foundational analysis and a clear, empirically-grounded justification for future research into the more sophis-



(a) ROC Curves show high AUROC for XGBoost.



(b) Classification metrics for O/NO=10 split.

Fig. 5: Classification performance for the inter-flow detection model. While the top-line metrics such as AUROC and Balanced Accuracy appear strong, the overall performance did not show an improvement over purely intra-flow models, providing the first hint of the model's limitations.

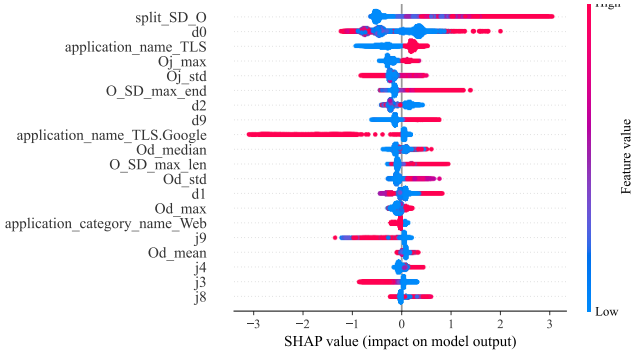


Fig. 6: SHAP analysis of the classification model trained for O/NO split 10. Features with `d` prefix relate to delay measurements, while those with `j` are jitter related. `split_SD_O` shows how close are the delay and jitter measurements at the end of the O part to becoming an SD event. Features from covering flows would have the `cov_` prefix on them.

ticated, structure-aware models required to solve this important problem.

Data and Code Availability

To ensure the reproducibility of our results, all data, analysis scripts, and code for generating figures are publicly available at [13]. This includes supplementary analyses not presented in the main text.

Acknowledgment

Supported by the János Bolyai Research Scholarship of the Hungarian Academy of Sciences. This work has been part of Celtic-Next project RAI-6Green: Robust and AI Native 6G for Green Networks with project-id: C2023/1-9 funded by 2024-1.2.6-EUREKA-2024-00009. This work was also supported by projects TKP2021-NVA-02 (financed under the TKP2021-NVA scheme) and 2024-1.2.6-EUREKA-2024-00009 (financed under the 2024-1.2.6-EUREKA scheme), implemented with support provided by the Ministry of Culture and Innovation of Hungary from the National Research, Development and Innovation Fund.

References

[1] O. Michel *et al.*, "The programmable data plane: Abstractions, architectures, algorithms, and applications," *ACM Computing Surveys*, vol. 54, no. 4, pp. 1–36, 2021. DOI: 10.1145/3447868.

[2] B. Bicski and A. Pekar, "Unveiling latency-induced service degradation: A methodological approach with dataset," *IEEE Access*, vol. 12, pp. 128 097–128 116, 2024. DOI: 10.1109/access.2024.3456588.

[3] B. Bicski and A. Pekar, "Early detection of network service degradation: An intra-flow approach," in *2024 20th International Conference on Network and Service Management (CNSM)*, 2024, pp. 1–5. DOI: 10.23919/cnsm62983.2024.10814309.

[4] L. D. Valdez *et al.*, "Cascading failures in complex networks," *Journal of Complex Networks*, vol. 8, no. 2, 2020. DOI: 10.1093/comnet/cnaa013.

[5] A. Smolyak *et al.*, "Mitigation of cascading failures in complex networks," *Scientific Reports*, vol. 10, no. 1, 2020. DOI: 10.1038/s41598-020-72771-4.

[6] B. AsSadhan *et al.*, "Network anomaly detection using a cross-correlation-based long-range dependence analysis," *International Journal of Network Management*, vol. 30, no. 6, 2020. DOI: 10.1002/nem.2129.

[7] J. Zhou *et al.*, *Multi-view correlation-aware network traffic detection on flow hypergraph*, 2025. arXiv: 2501.08610 [cs.CR].

[8] H. Li *et al.*, "Correlation-based anomaly detection method for multi-sensor system," *Computational Intelligence and Neuroscience*, vol. 2022, pp. 1–13, 2022. DOI: 10.1155/2022/4756480.

[9] M. Saqr, "Temporal network analysis: Introduction, methods and analysis with r," in *Learning Analytics Methods and Tutorials*. 2024, pp. 541–567. DOI: 10.1007/978-3-031-54464-4_17.

[10] D. Firestone *et al.*, "Azure accelerated networking: Smartnics in the public cloud," in *Proceedings of the 15th USENIX Conference on Networked Systems Design and Implementation (NSDI '18)*, 2018, pp. 51–64.

[11] P. Aitken *et al.*, *Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of Flow Information*, RFC 7011, 2013. DOI: 10.17487/RFC7011.

[12] S. Panchen *et al.*, *InMon Corporation's sFlow: A Method for Monitoring Traffic in Switched and Routed Networks*, RFC 3176, 2001. DOI: 10.17487/RFC3176.

[13] FlowFrontiers, *Inter-Flow Service Degradation Detection - Digital Artifacts*, https://github.com/FlowFrontiers/ServDeg-Inter, 2025.