

Building Transparency in Deep Learning-Powered Network Traffic Classification: A Traffic-Explainer Framework

Riya Ponraj
University of Oregon

Ram Durairajan
University of Oregon, Link Oregon

Yu Wang
University of Oregon

ABSTRACT

Recent advancements in deep learning (DL) have significantly enhanced the performance and efficiency of traffic classification in networking systems. However, the lack of transparency in their predictions and decision-making has made network operators reluctant to deploy DL-based solutions in production networks. To tackle this challenge, we propose Traffic-Explainer, a model-agnostic and input-perturbation-based traffic explanation framework. By maximizing the mutual information between predictions on original traffic sequences and their masked counterparts, Traffic-Explainer automatically uncovers the most influential features driving model predictions. Extensive experiments demonstrate that Traffic-Explainer improves upon existing explanation methods by approximately 42%. Practically, we further apply Traffic-Explainer to identify influential features and demonstrate its enhanced transparency across three critical tasks: application classification, traffic localization, and network cartography. For the first two tasks, Traffic-Explainer identifies the most decisive bytes that drive predicted traffic applications and locations, uncovering potential vulnerabilities and privacy concerns. In network cartography, Traffic-Explainer identifies submarine cables that drive the mapping of traceroute to physical path, enabling a traceroute-informed risk analysis. Our implementation is publicly available at <https://anonymous.4open.science/r/TrafficExplainer-5E2E/README.md>.

ACM Reference Format:

Riya Ponraj, Ram Durairajan, and Yu Wang. 2025. Building Transparency in Deep Learning-Powered Network Traffic Classification: A Traffic-Explainer Framework. In *Proceedings of* . ACM, New York, NY, USA, 13 pages. <https://doi.org/10.1145/nnnnnnn.nnnnnnn>

1 INTRODUCTION

Network traffic classification [21, 48], which infers properties of network transmissions from packet flows between interconnected devices in networked systems, supports numerous critical tasks, such as application classification [21, 48], traffic localization [27], and network cartography [10, 28]. Recent advances in deep learning (DL) have led to powerful automated solutions for traffic classification. However, decisions made by DL models are based on learned features and lack explainability. Although earlier feature-engineering-based traffic classifiers, such as fingerprint matching [40, 42], are

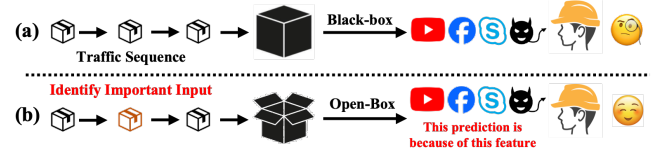


Figure 1: (a) Untransparent scenario: DL solutions that solely present predictions with no explanation lose the trust of network operators. (b) Transparent scenario: DL solutions that offer explanations for their decision-making gain by highlighting the critical unit driving the prediction of the traffic input sequence, earning the trust of network operators.

inherently explainable by manually extracting features (e.g., device and certificate) based on domain expertise, they are less effective and nonadaptive. DL-based models like ET-Bert [21] and TFE-GNN [48] operate in latent embedding spaces, where internal representations (i.e., hidden neurons) have no clear correspondence with human-understandable concepts or predicted labels. This opacity limits transparency and has made network operators hesitant to adopt DL-based solutions in traffic classification [11, 19]. In high-stakes environments, such as government networks or critical infrastructure, a DL model may flag traffic as suspicious or associated with prohibited applications (e.g., BitTorrent) without revealing whether the decision stemmed from payload patterns, header anomalies, or timing irregularities. Without such insight, network operators cannot verify actions like blocking or rerouting, complicating compliance and auditability.

To improve the transparency, conventional DL-based explanation methods [2, 43] can be naturally employed. For example, gradient-based methods such as Saliency Maps and Grad-CAM [35, 38, 46] can be applied to uncover the most determinant factors based on the input gradient. Despite these well-established explanation methods in the general DL literature [4, 23, 31, 47, 49], their adoption in traffic classification or even broader networking system domains remains limited. Although traditional rule-based approaches are self-explainable [19], their reliance on hand-crafted features makes them inflexible and rigid to generalize across different datasets or tasks. To date, only a few works have explicitly bridged explainability and networking systems. NetXplain [26] focuses solely on explaining traffic delays, and Hybrid Explainability [11] is restricted to post-hoc interpretation of non-DL models such as decision trees. There is still no unified framework that systematically explains the behavior of DL-based traffic classification models.

Given the criticality yet the nascent state of building the transparency of DL-powered traffic classification, this paper presents an explanation framework, Traffic-Explainer, designed explicitly for DL-based traffic classification applications. Achieving this goal requires addressing three key challenges. (1) *First, we need to determine the explanation object that balances specificity (i.e., encoding*

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2025 Association for Computing Machinery.
ACM ISBN 978-x-xxxx-xxxx-x/YY/MM...\$15.00
<https://doi.org/10.1145/nnnnnnn.nnnnnnn>

sufficient class-desired information to enable the explanation) and generalizability (i.e., being applicable across various traffic classification scenarios). Since many network traffic classification problems can be formulated as sequential classification (e.g., application classification based on sequential traffic packets and network cartography based on sequential round-trip times (RTTs)), Traffic-Explainer treats each sequence as the basic explanation object and aims to identify the most critical units within the sequence responsible for its classification. For instance, in traffic application classification, where sequences consist of bytes or packets, explanations should highlight the critical bytes and byte-byte interactions that determine the classification outcome. In network cartography, where traffic sequences consist of sub-segments annotated with RTTs across hops, effective explanations should pinpoint the specific RTT hop that best characterizes the underlying physical fiber cable carrying that particular traffic. (2) *Second, the underlying classifier should be capable of handling sequential data, achieving high performance, and remaining accessible to practitioners.* To this end, we apply our proposed Traffic-Explainer to explain predictions from two widely-used DL architectures: transformer, for its advanced capabilities on sequential data, and multi-layer perceptron (MLP), for its simplicity and usability. This design choice underscores the model-agnostic nature of our Traffic-Explainer. (3) *Thirdly, the generated explanations should be naturally interpretable to domain experts such as network operators.* To achieve this, our Traffic-Explainer formulates explanation generation as a mutual information maximization [47], aiming to identify the most informative input units (e.g., bytes and RTTs) that maximize interpretability from the network operator perspective. We summarize contributions as follows:

- **Novel Explanation Problem on Traffic Classification:** This work pioneers the explanation of DL-based traffic classification, aiming to enhance the transparency of DL-driven decisions and provide domain insights to assist network operators.
- **Systematic Explanation Framework for Traffic Classification:** We introduce Traffic-Explainer, a model-agnostic and input-perturbation-based explanation framework that identifies the most influential features by maximizing the mutual information between predictions on original inputs and their masked versions. Extensive experiments validate its effectiveness, efficiency, and transferability of explanations across DL models.
- **Three Real-world Applications of Traffic Classification:** We demonstrate the efficacy of Traffic-Explainer through three representative use cases. For application classification and traffic localization, Traffic-Explainer identifies the most characteristic bytes driving instance-level predictions, such as a given application or location. For network cartography, the generated explanations enable network operators to automatically pinpoint the most likely submarine cables traversed by a given traffic traceroute, resulting in more transparent logical-to-physical dependencies.

2 RELATED WORK

Explainable DL-based Network Traffic Classification. The ubiquity of network traffic coupled with a shortage of skilled operators has driven the adoption of deep learning for automated traffic classification [1, 11, 18, 19]. However, their black-box nature raises concerns about transparency and motivates the development

of explanation techniques. Explanation methods can be generally categorized into four types: gradient, perturbation, surrogate, and decomposition methods. Gradient methods analyze the gradients of model outputs with respect to input features [36]. Perturbation-based methods modify input data and observe changes in model outputs [32]. Surrogate methods, such as LIME [31], approximate model behavior using simpler interpretable models. Decomposition methods break down predictions into additive contributions of input features [12]. Despite their broad application in general DL [4, 23, 31, 47, 49], they have seen limited adoption in DL-powered solutions [11]. For instance, rule-based traffic classification is inherently explainable [19]. However, its static handcrafted logic lacks adaptability across diverse datasets and tasks. Notably, two ML-based networking explanation frameworks, such as NetXplain [26] and Hybrid Explainability [11], remain narrow in scope, where the former targets only traffic delay explanations, while the latter focuses on post-hoc interpretations for non-DL models. To date, no framework systematically explains DL behaviors of traffic classification. This gap motivates us to propose Traffic-Explainer, which adapts a perturbation-based masking strategy to identify key sequence units for explanation. We next review its three applications.

Application Classification and Traffic Localization. Traffic classification aims to categorize sequential traffic flows composed of byte sequences and has been used in application classification and traffic localization [6, 21, 27, 48]. Traditional methods rely on handcrafted features, such as traffic fingerprints and statistics, which require extensive domain expertise and are limited to specific traffic scenarios [22, 39]. Recent DL approaches (e.g., Transformers and Graph Neural Networks) to automatically learn representations for traffic classification [21, 48]. However, the black-box nature of traffic classification models makes their predictions difficult to interpret, highlighting the need for explanation techniques to identify the key components in each traffic sequence that drive instance-level predictions and characterize class-level signatures. For example, these explanation techniques are expected to reveal which byte patterns are most influential in distinguishing the "Chat" application from "P2P", or distinguishing locations where traffic occurs.

Network Cartography. Network cartography aims to establish the dependency between the logical traffic and the physical infrastructure layer in networking systems [8]. Given the routing of traffic along IP-level paths, the goal is to infer the physical paths (e.g., the terrestrial or submarine fiber-optic cables) traversed by the traffic. Prior efforts [5, 28] have relied on commonsense heuristics, such as speed-of-light filters, geographic proximity, and networking domain expertise. However, these signals are noisy, incomplete, and hard-coded [41]. For instance, cables owned by the same provider can be challenging to differentiate using ownership data alone, and IP geo-locating services are notoriously known to produce erroneous results. To address these limitations, recent work [30] has explored temporal patterns within traceroutes (i.e., the sequence of RTTs) to infer physical cable mappings. Building on this direction, our proposed Traffic-Explainer can be used as a post-hoc analysis tool to identify the most discriminative temporal features, such as a specific hop with a characteristic RTT, that reveal the property of the underlying physical cable. These features can serve as cable signatures, supporting risk assessment/resilience planning in networking design [10, 34].

3 PRELIMINARY

Since many traffic classification problems, such as application classification, traffic localization, and network cartography in Section 2, operate on sequence-based traffic data, we adopt the sequence as the fundamental unit for DL-based traffic classification and the subsequent explanation. We now introduce notations and problems.

3.1 Mathematical Notations

Let $\mathcal{X} = \{(\mathcal{X}^i, \mathcal{Y}^i)\}_{i=1}^N$ represent a set of N traffic sequences, where the i^{th} sequence consists of a series of units $\mathcal{X}^i = \{\mathcal{X}_j^i\}_{j=1}^{|\mathcal{X}^i|}$ and its corresponding one-hot label $\mathcal{Y}^{i,*} \in \{0, 1\}^C$ with C being the class number. In the context of application classification and traffic localization, each sequence \mathcal{X}^i corresponds to a sequence of bytes, where each byte \mathcal{X}_j^i takes values in the range $[0, 255]$, and the label $\mathcal{Y}^{i,*}$ represents the downstream application (e.g., YouTube, Skype, and Facebook) or traffic locations (e.g., US, China, and India). In network cartography, each sequence \mathcal{X}^i consists of logical-layer measurements, where each measurement \mathcal{X}_j^i is a real-valued RTT, and the label $\mathcal{Y}^{i,*}$ corresponds to the traversed physical infrastructure (e.g., fiber links). Assuming the traffic classifier as g_{Θ_g} (e.g., transformer and MLP) and the proposed Traffic-Explainer as h_{Θ_h} , we formulate the explanation problem of sequence-level classification in the following.

3.2 Problem Statement

Given a traffic sequence \mathcal{X}^i , we aim to:

- **Learn an optimal classifier** $g_{\Theta_g^*}: \mathcal{X}^i \rightarrow \mathcal{Y}^i$ so that the predicted class \mathcal{Y}^i matches the ground-truth class $\mathcal{Y}^{i,*}$.
- **Learn an optimal explainer** $h_{\Theta_h^*}: (\mathcal{X}^i, \mathcal{Y}^i, g_{\Theta_g^*}) \rightarrow \mathbf{M}^{i,*}$ so that the learned feature mask $\mathbf{M}^{i,*} \in [0, 1]^{|\mathcal{X}^i|}$ could maximally explain the prediction \mathcal{Y}^i by the classifier $g_{\Theta_g^*}$ over \mathcal{X}^i .

Notably, $\mathbf{M}^{i,*}$ denotes the feature mask over each unit in the sequence \mathcal{X}^i , where each value ranges between 0 and 1 and represents the importance of that unit in contributing to the model prediction. This problem setup can be generalized to many sequence-based traffic classification applications, such as application classification, traffic localization, and network cartography.

4 FRAMEWORK

Building on the above notations and problem formulation, we next briefly introduce our transformer-based traffic classifier since understanding its architecture informs the design of our proposed Traffic-Explainer. *Notably, Traffic-Explainer is an explanation model that differs from the classifier itself. Furthermore, Traffic-Explainer directly operates in the input space to identify the most influential features (i.e., units within a sequence), making it inherently model-agnostic.* While our primary classifier is based on the Transformer architecture, we also evaluate Traffic-Explainer using a simpler backbone, such as a multi-layer perceptron (MLP), to demonstrate its generality and adaptability across model types. Given the simplicity and well-known structure of MLPs, we omit a detailed discussion of their architecture and instead focus on briefly introducing the Transformer-based backbone.

4.1 Transformer-based Traffic Classifier

Transformer-based traffic classifier g_{Θ_g} comprises self-attention, feed-forward, and pooling layers, which are introduced next.

4.1.1 Unit Tokenization. Each sequence \mathcal{X}^i consists of a set of units, where unit \mathcal{X}_j^i is mapped to a d -dimensional embedding via a learnable embedding matrix $\mathbf{E} \in \mathbb{R}^{|\mathcal{V}| \times d}$, where $|\mathcal{V}|$ denotes the vocabulary size. For instance, in application classification, each byte b_k in a packet takes a value in $[0, 256]$, where 256 serves as a padding token to standardize packet lengths. We retrieve the embedding of the j^{th} unit as $\mathbf{e}_j^i = \mathbf{E}[\mathcal{X}_j^i, :]$, $\forall \mathcal{X}_j^i \in \mathcal{X}^i$. For real-valued RTT sequences in network cartography, the units are projected into a continuous embedding space via a learnable linear transformation.

4.1.2 Iterative Self-Attention and Feed-Forward. To capture dependencies among units, we apply self-attention. Since ordering information is essential (e.g., byte order in packets or temporal order in RTTs), we incorporate positional encoding Φ_j into each unit embedding. The transformer then applies self-attention over the sequence to produce contextual embeddings: $\{\mathbf{h}_j^i\}_{j=1}^{|\mathcal{X}^i|} = \text{Self-ATT}(\{\mathbf{e}_j^i, \Phi_j\})_{j=1}^{|\mathcal{X}^i|}$. These embeddings are subsequently passed through a feed-forward layer. This process is repeated iteratively across multiple transformer layers to progressively refine the representation.

4.1.3 Pooling and Classification. After obtaining contextual unit embeddings $\{\mathbf{h}_j^i\}_{j=1}^{|\mathcal{X}^i|}$, we aggregate them into a fixed-size sequence embedding via a pooling: $\mathbf{F}^i = \text{Pooling}(\{\mathbf{h}_j^i\}_{j=1}^{|\mathcal{X}^i|})$, where the typical pooling operation could be mean-pooling, and the obtained sequence embedding \mathbf{F}^i serves as the final embedding for classification. Given the ground-truth class $\mathcal{Y}^{i,*}$, we optimize the classifier parameters Θ_g by minimizing the cross-entropy loss: $\Theta_g^* = \arg \min_{\Theta_g} \sum_{i=1}^N \sum_{c=1}^C \mathcal{Y}_c^{i,*} \log \mathcal{Y}_c^i$ where $\mathcal{Y}^i \in \mathbb{R}^C$ is the predicted class distribution after applying softmax normalization and linear mapping on \mathbf{F}^i . This formulation ensures that our transformer-based classifier supports both byte/RTT sequences for application classification, country localization, and network cartography.

4.2 Traffic-Explainer

After introducing the traffic classifier, this section focuses on developing our proposed explanation framework, Traffic-Explainer. We aim to identify the most important units that are responsible for prediction decisions. Formally, given a sequence $\mathcal{X}^i = \{\mathcal{X}_j^i\}_{j=1}^{|\mathcal{X}^i|}$, we seek to extract a subset of units $\hat{\mathcal{X}}^i \subseteq \mathcal{X}^i$ that are most responsible for its prediction \mathcal{Y}^i . Since the notion of a fundamental unit varies across applications (e.g., bytes in traffic classification, RTTs in network cartography), Traffic-Explainer provides a generalizable approach by centering explanations on these core sequence components. For instance, bytes form the fundamental building blocks of traffic flows across diverse protocols, while RTTs offer ubiquitous network performance measurements. By identifying the most critical units for classification, our method enhances interpretability across heterogeneous domains. Moreover, our framework extends beyond individual units to capture unit-unit interactions by refining the masking mechanism to operate at the self-attention layer rather than the input level. We demonstrate the explanation effectiveness at the byte level and the byte-byte interaction level in Section 5.

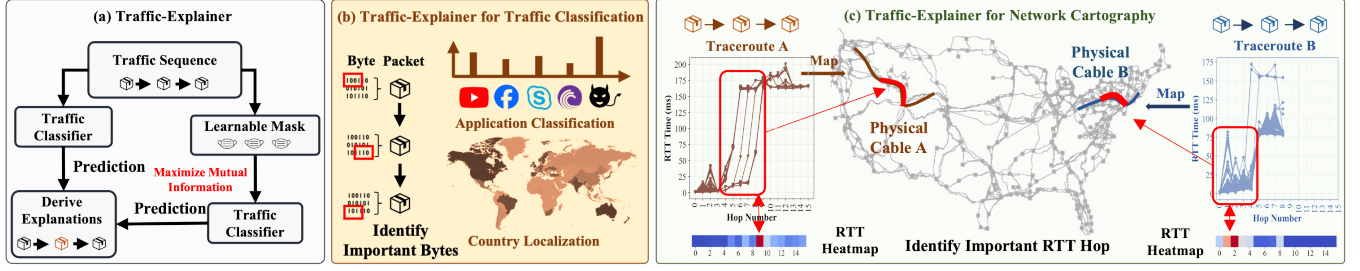


Figure 2: (a) Given a traffic sequence $\mathcal{X}^i = \{\mathcal{X}_j^i\}_{j=1}^{|\mathcal{X}^i|}$ consisting of $|\mathcal{X}^i|$ feature units, an MLP or Transformer-based traffic classifier first predicts the corresponding traffic label. To provide interpretability, our proposed Traffic-Explainer identifies the Top-K most influential input units that drive the model prediction by optimizing input masks via mutual information maximization. (b) In application classification and traffic localization, Traffic-Explainer produces explanations in the form of individual bytes in the traffic packet sequence that are most critical for the model’s decision. (c) In the network cartography, Traffic-Explainer highlights the specific RTT-hop in a traceroute sequence that corresponds to the physical submarine cable being traversed, enabling transparent mapping between logical observations and physical infrastructure.

Given a sequence of units \mathcal{X}^i , to identify the most important units, we initialize a learnable unit masking matrix $\mathbf{M}^i \in [0, 1]^{|\mathcal{X}^i|}$ with \mathbf{M}_j^i denoting the importance of unit j in contributing to the model prediction of \mathcal{X}^i to be \mathcal{Y}^i . The traffic classifier predicts with the masked sequence, i.e., its first self-attention layer becomes: $\{\hat{\mathbf{h}}_j^i\}_{j=1}^{|\mathcal{X}^i|} = \text{Self-ATT}(\{(\mathbf{e}_j^i * \sigma(\mathbf{M}_j^i), \Phi_j)\}_{j=1}^{|\mathcal{X}^i|})$ where σ is the sigmoid function mapping the mask score to a value between 0 and 1. The transformed units $\{\hat{\mathbf{h}}_j^i\}_{j=1}^{|\mathcal{X}^i|}$ are then aggregated via pooling followed by the linear classifier following Section 4.1. A higher value of \mathbf{M}_j^i after optimization by explanation objective indicates the higher importance of unit \mathcal{X}_j^i in predicting the sequence label. We next introduce explanation objectives at the local-instance/global-class levels for optimizing the unit-level masking \mathbf{M}^i .

4.2.1 Local Instance Explanation. For each sequence \mathcal{X}^i with predicted class $\mathcal{Y}^i = g_{\Theta_g^*}(\mathcal{X}^i)$, we optimize explanation by maximizing mutual information (MI) between the prediction \mathcal{Y}^i and its explanation $\hat{\mathcal{X}}^i$ (i.e., a subset of units):

$$\max_{\hat{\mathcal{X}}^i} \text{MI}(\mathcal{Y}^i, \hat{\mathcal{X}}^i) = H(\mathcal{Y}^i) - H(\mathcal{Y}^i | \hat{\mathcal{X}}^i) \quad (1)$$

For traffic sequence \mathcal{X}^i , MI quantifies the change in the prediction \mathcal{Y}^i when the sequence \mathcal{X}^i is masked to become the explained sub-sequence $\hat{\mathcal{X}}^i$. For example, if removing the unit $\hat{\mathcal{X}}_j^i$ strongly decreases the prediction probability of $\max_{c \in \mathcal{C}} \mathcal{Y}_c^i$, the unit $\hat{\mathcal{X}}_j^i$ is naturally a good counterfactual explanation for the prediction of sequence \mathcal{X}^i . Maximizing the mutual information between the predicted label distribution \mathcal{Y}^i and explanation (i.e., the masked sequence) $\hat{\mathcal{X}}^i$ equals minimizing the conditional entropy $H(\mathcal{Y}^i | \hat{\mathcal{X}}^i)$:

$$\mathbf{M}^{i,*} = \arg \min_{\mathbf{M}^i} H(\mathcal{Y}^i | \hat{\mathcal{X}}^i) = \arg \min_{\mathbf{M}^i} -\mathbb{E}_{\mathcal{Y}^i | \mathcal{X}^i} [\log P_{g_{\Theta_g^*}}(\hat{\mathcal{Y}}^i | \hat{\mathcal{X}}^i)]. \quad (2)$$

The explanation for prediction \mathcal{Y}^i is thus a subsequence of units \mathcal{X} that minimizes uncertainty of $g_{\Theta_g^*}$, following the intuition that *the explanation should be the ones that if only keep features identified by the explainer and remove all other input features, the model would become more confident about its original prediction distribution.*

Rather than explaining through the model confidence, the network operators sometimes care about “*why does the trained traffic classifier predict a certain class label?*”. Therefore, we modify the conditional entropy objective in Eq. (2) towards the class label c :

$$\mathbf{M}^{i,*}(c) = \arg \min_{\mathbf{M}^i} - \sum_{j=1}^C \mathbb{1}[\mathcal{Y}_j^i = c] \log P_{g_{\Theta_g^*}}(\hat{\mathcal{Y}}_j^i | \hat{\mathcal{X}}^i). \quad (3)$$

The explanation for prediction is thus a subsequence of units $\hat{\mathcal{X}}$ that, if only keep them and remove all other units in the sequence, would maximize the prediction score of the model towards its original predicted class (as compared to its original prediction distribution as before). We empirically find that this objective slightly outperforms the previous confidence objective in explanation.

4.2.2 Global Class Explanation. Both previous explanations are only at the instance level. In real-world applications, what is more interesting is why the model always makes certain predictions about a group of instances, e.g., the ones belonging to the same class. This motivates us to explain the predictions at the global class level. For example, identifying the most important units driving predictions towards a certain class across all instances.

$$\mathbf{M}^* = \arg \min_{\mathbf{M}} - \sum_{i=1}^N \sum_{j=1}^C \mathbb{1}[\mathcal{Y}_j^i = c] \log P_{g_{\Theta_g^*}}(\hat{\mathcal{Y}}_j^i | \hat{\mathcal{X}}^i) \quad (4)$$

4.2.3 Mask Regularization. Blindly optimizing the mask \mathbf{M} following the above explanation-based objectives Eq. (2)–(4) may lead to trivial explanations, e.g., $\hat{\mathcal{X}} = \mathcal{X}$, where all units in the original sequence are tagged important, as it would naturally encompass information necessary to explain the model prediction. Moreover, a widely adopted assumption in feature selection and sparse representation learning is that model prediction should primarily be attributed to a subset of the inputs [20]. To prevent the trivial use of the entire sequence as the explanation and also consider the sparsity principle, we impose a predefined budget B to limit the magnitude of the explanation mask. We weighted combine the explanation loss $\mathcal{L}^{\text{Explain}}$ and the budget constrain loss:

$$\mathbf{M}^* = \arg \min_{\mathbf{M}} \mathcal{L} = \alpha_1 \text{ReLU}(\|\mathbf{M}\|_1 - B) + \alpha_2 \mathcal{L}^{\text{Explain}}, \quad (5)$$

where $\mathcal{L}^{\text{Explain}}$ could refer to any of the previous three explanation losses defined in Eq (2)-(4). *Intuitively, minimizing \mathcal{L} identifies the most informative units while maximally excluding units irrelevant to the explanation, thereby refining the final explanation.*

Although the presented framework only focuses on identifying the most critical units, it can be easily tailored to identify the most important unit-unit interactions by masking the unit self-attention. Therefore, we also include this level of explanation in Table 1.

5 EXPERIMENTS

In this section, we conduct experiments to evaluate the effectiveness, efficiency, and transferability of the explanation discovered by Traffic-Explainer. We introduce experimental settings below.

5.1 Experimental Settings

5.1.1 Application Tasks and Dataset Collection. We demonstrate the practical usage of our proposed Traffic-Explainer through three application tasks, including their objectives and dataset collection, explanation baselines, evaluation strategies, and hyperparameter settings. Comprehensive details are in Appendix A.

- **Application 1 - Application Classification [21, 48]:** This task classifies the application (e.g., YouTube, Skype) based on traffic flows represented as byte sequences. The explanation identifies the most influential bytes that drive the predicted application. Following [48], we validate the proposed Traffic-Explainer on four datasets: ISCX-VPN, ISCX-NonVPN, ISCX-Tor, and ISCX-NonTor. We use SplitCap to obtain bidirectional flows and increase the training samples in ISCX-Tor by dividing each flow into 60-second non-overlapping blocks.
- **Application 2 - Traffic Localization [27, 45]:** aims to identify the country where a given sequence of network traffic occurred. This is conceptually framed similarly to traffic classification, where given a network flow (hex-encoded data), it predicts the country label of its origin or destination. The explanation is to identify the most critical bytes that contribute to the model prediction of the country. Two datasets, IOS-Cross Platform and Android-Cross Platform [27, 42], are used to showcase Traffic-Explainer in identifying traffic country localization. These two datasets comprise user-generated data for 215 Android and 196 iOS apps in the US, China, and India.
- **Application 3 - Network Cartography [29]:** focuses on mapping traffic traceroute data to physical submarine cables. Each data instance represents the routing of traffic along IP-level paths, with the goal of inferring the corresponding physical paths (e.g., terrestrial or submarine fiber-optic cables). The explanation task aims to identify the most critical RTT hops that contribute to the mapping of traffic to physical cables. We collect traceroutes from predefined source locations to destinations. Using RIPE Atlas probes [16], we select target countries and direct probes to a set of international servers. Over a three-day collection period, we gathered 5,000 traceroutes across 10 unique source-destination pairs, which serve as classification labels. Among these, three key classes include: (1) Seattle, US to Yokohama, Japan (submarine route); (2) Seattle, US to San Jose, US (terrestrial route); and (3) Virginia Beach, US to San Sebastian, France (submarine route). Details about dataset statistics are presented in Appendix A.

5.1.2 Explanation Baselines. To benchmark the explanation by the proposed Traffic-Explainer, in the first application classification, we compare it with five explanation baselines: **Random**-we randomly select the Top-K bytes or byte-byte interactions and treat them as the explanation; **Saliency Map/Gradient-based Methods** [38]-we calculate the gradient of the output prediction with respect to either each byte or each byte-byte interaction and then select the ones with top-K gradient magnitude as the explanation; **Self-Attention** [14]-we use the transformer attention as the importance score for each byte-byte interaction and select the top-K ones as the explanation; **LIME** [31]-A local surrogate model is trained by perturbing the input (e.g., masking byte values) and observing the model's output. The learned weights of the surrogate model approximate the importance of each byte or byte value, from which the Top-K most influential ones are selected; **SHAP** [23]-SHAP estimates the contribution of each byte or byte-byte interaction to the model's output using Shapley values from cooperative game theory. The Top-K most important features are selected based on their absolute Shapley values.

5.1.3 Evaluation Metrics. Four explanation evaluation metrics are used: Fidelity, Accuracy, Counterfactual Fidelity, and Counterfactual Accuracy [3?]. For each instance, we extract the Top-K most important bytes as the explanation and assess how the model prediction changes when these bytes are either (1) removed for counterfactual fidelity and counterfactual accuracy evaluation or (2) exclusively retained for fidelity and accuracy evaluation. Due to space limitations, we present their comprehensive definition and computation equation in the Appendix B.

5.1.4 Implementation Details and Hyperparameters. We train the traffic classifier using the following hyperparameters: 1,000 training epochs; batch size selected from {64, 512, 4096}; learning rate from {0.001, 0.01}; and dropout rate from {0.2, 0.5}. For the first application, we preprocess traffic sequences by limiting each to a maximum of 50 packets, with each packet truncated to 150 payload bytes and 40 header bytes, following [48]. For the second application, we adopt the preprocessing and configuration from [27]. For the third application, we manually collect 5,000 traceroutes spanning 10 different submarine cables and split the data into training/validation/test sets using a 70%/15%/15% ratio. Once the Traffic Classifier is trained, its predictions are used to derive explanations via mutual information optimization at both the byte and byte-byte levels. Explanations are represented as score masks, either vectors in \mathbb{R}^{257} (for individual bytes) or matrices in $\mathbb{R}^{257 \times 257}$ (for byte-byte interactions). We rank these scores to select the Top-K bytes or interactions and evaluate explanation quality using fidelity and counterfactual accuracy. Further implementation details are provided in Appendix C.

5.2 Application Classification

5.2.1 Local-Instance Explanation. In Table 1, we evaluate Traffic-Explainer against Random/Saliency Map/SHAP/LIME for byte-level explanations, and against Random/Self-Attention for byte-byte interactional level explanations. This choice reflects the nature of each baseline: Self-Attention captures pairwise interactions and is unsuitable for byte-level explanation, while Saliency Map, SHAP, and LIME are not designed to model pairwise dependencies, making them infeasible for byte-byte interactional level explanation.

Object	Explainer		ISCX-VPN				ISCX-nonVPN				ISCX-Tor				ISCX-nonTor			
			Fid	Acc	C-Fid	C-Acc	Fid	Acc	C-Fid	C-Acc	Fid	Acc	C-Fid	C-Acc	Fid	Acc	C-Fid	C-Acc
Byte Level	Random	1%	31.9	31.2	1.27	4.46	22.5	22.8	1.5	10.6	13.2	9.2	0.57	17.8	43.6	43.9	0.29	2.88
		5%	35.7	35.7	4.50	8.30	25.6	25.8	4.60	13.4	14.4	12.6	3.50	21.3	42.4	42.7	1.80	3.40
		10%	39.5	40.8	9.60	12.1	28.6	28.9	7.90	12.9	16.1	12.6	5.80	23.0	42.9	43.2	4.50	6.10
	Saliency Map	1%	33.8	35.0	6.40	9.60	23.3	23.5	0.76	10.1	31.6	22.4	29.3	42.5	37.9	38.1	3.00	4.90
		5%	69.4	72.0	50.3	52.2	40.0	39.5	14.9	18.7	32.8	23.0	28.7	36.2	31.8	31.8	12.5	13.2
		10%	71.3	72.0	53.5	52.9	44.6	43.3	22.0	24.1	46.0	36.8	43.7	52.9	51.8	51.5	21.7	21.8
	SHAP	1%	33.8	33.8	0.00	5.73	22.8	23.0	0.00	10.4	9.5	0.00	0.00	9.5	44.2	44.5	0.00	2.63
		5%	33.8	33.8	0.00	5.73	22.8	23.0	0.00	10.4	9.5	0.00	0.00	9.5	44.2	44.5	0.00	2.63
		10%	33.8	33.8	0.00	5.73	22.8	23.0	0.00	10.4	9.5	0.00	0.00	9.5	44.2	44.5	0.00	2.63
	LIME	1%	80.9	<u>79.6</u>	<u>54.1</u>	<u>54.8</u>	<u>60.0</u>	<u>57.5</u>	<u>24.1</u>	<u>26.1</u>	56.9	45.4	<u>42.5</u>	51.7	<u>51.4</u>	<u>51.5</u>	<u>10.1</u>	<u>10.8</u>
		5%	<u>94.9</u>	<u>92.4</u>	<u>69.4</u>	<u>70.1</u>	<u>75.7</u>	<u>71.1</u>	<u>42.8</u>	<u>40.2</u>	<u>68.4</u>	<u>54.6</u>	<u>68.4</u>	<u>73.6</u>	<u>69.0</u>	<u>68.8</u>	<u>21.0</u>	<u>20.7</u>
		10%	<u>96.8</u>	<u>93.0</u>	<u>78.3</u>	<u>76.4</u>	<u>86.3</u>	<u>79.5</u>	<u>60.3</u>	<u>55.7</u>	<u>79.9</u>	<u>66.1</u>	<u>77.6</u>	<u>79.9</u>	<u>81.0</u>	<u>80.6</u>	<u>31.1</u>	<u>30.4</u>
	Traffic-Explainer	1%	82.8	81.5	58.6	56.7	65.3	65.1	29.1	33.2	<u>44.3</u>	<u>39.1</u>	49.4	<u>50.6</u>	74.5	74.8	20.2	20.9
		5%	97.5	92.4	82.8	79.6	92.4	84.6	78.5	74.9	92.0	77.6	86.8	88.5	96.1	95.4	71.1	70.5
		10%	98.7	93.0	84.1	80.9	96.2	87.3	79.8	76.5	97.7	81.0	86.8	90.2	97.9	96.0	77.2	76.8
Byte-Byte Level	Random	1%	28.7	27.4	0.00	5.70	34.9	35.7	0.51	10.4	16.7	16.7	0.57	<u>18.4</u>	17.7	17.4	0.83	2.70
		5%	35.0	33.8	0.64	5.10	40.5	40.3	1.30	10.1	27.6	31.6	3.50	19.5	26.0	25.8	0.29	2.80
		10%	38.9	36.9	1.90	5.10	52.2	51.9	2.00	10.4	31.6	36.2	4.00	20.7	<u>55.3</u>	<u>55.0</u>	0.42	2.70
	Self-Attention	1%	<u>93.6</u>	93.6	<u>8.90</u>	<u>10.2</u>	<u>93.9</u>	89.1	<u>8.90</u>	<u>12.7</u>	<u>53.5</u>	<u>56.9</u>	<u>9.20</u>	<u>14.4</u>	<u>37.9</u>	<u>38.1</u>	<u>3.00</u>	<u>4.90</u>
		5%	<u>93.6</u>	93.0	74.5	76.4	<u>97.0</u>	89.1	43.5	44.1	<u>69.5</u>	<u>61.5</u>	<u>46.0</u>	<u>52.3</u>	<u>31.8</u>	<u>31.8</u>	<u>12.5</u>	<u>13.2</u>
		10%	95.5	93.6	79.0	80.9	<u>98.2</u>	89.6	56.5	56.7	<u>71.3</u>	<u>61.5</u>	<u>59.8</u>	<u>64.4</u>	51.8	51.5	21.7	21.8
	Traffic-Explainer	1%	97.5	<u>92.4</u>	56.1	<u>52.2</u>	96.0	<u>87.1</u>	38.7	34.4	74.7	63.8	73.0	69.0	96.8	94.6	48.5	47.6
		5%	98.1	<u>92.4</u>	<u>58.6</u>	<u>58.0</u>	99.0	<u>88.6</u>	<u>37.0</u>	<u>36.2</u>	93.7	75.3	74.7	70.1	97.1	94.6	60.4	60.0
		10%	<u>94.9</u>	<u>89.2</u>	<u>56.7</u>	<u>58.0</u>	98.7	<u>88.4</u>	<u>38.5</u>	<u>39.8</u>	98.3	79.9	74.7	73.0	96.9	94.4	61.3	61.1

Table 1: Comparison of local-instance explanation at Byte and Byte-Byte levels. The best/runner-up results under the same byte budget are in bold and underlined. Our proposed Traffic-Explainer generates explanations of higher quality.

For byte-level explanations, the performance margin is significant. Traffic-Explainer requires only 5% of bytes (approximately $256 \times 5\% \approx 10$ bytes) to explain more than half of the networking application predictions. This finding suggests that certain bytes are particularly salient and closely associated with each network class, which aligns with the principles of sparsity in deep representation learning and supports our design choice of incorporating mask regularization. Among the four baselines, the Saliency Map and LIME outperform the Random and SHAP. The Saliency Map is inferior to Traffic-Explainer because it only considers the individual importance of bytes, overlooking their cooperative effect in contributing to the final prediction. In contrast, Traffic-Explainer jointly optimizes the mask over the whole sequence to identify the Top-K important bytes, thereby inherently capturing their interactions. The weaker explanation performance of LIME relative to Traffic-Explainer is likely due to its reliance on a simple surrogate model, which struggles to approximate the complex decision boundaries of the sequence traffic classifier. In comparison, Traffic-Explainer optimizes a learned mask on the original trained model, preserving the architecture function. Moreover, as the explanation size increases (i.e., the amount of masked bytes decreases), the explanation performance increases. We also verify the efficiency of Traffic-Explainer by empirically measuring the time for explaining each individual instance. On average, Traffic-Explainer takes 2.52s/2.12s/1.90s/2.10s for generating the explanation (i.e., the byte mask) for each traffic instance across the above four datasets. Traffic-Explainer applies a mask on the input data, without imposing additional complexity, as the underlying traffic classifier remains the same.

For byte-byte level explanations, Traffic-Explainer achieves high explanation in most cases. However, it performs slightly below the Self-Attention Baseline on the ISCX-VPN and ISCX-NonVPN datasets when evaluated using Acc and C-Fid metrics. We hypothesize that this is because the Self-Attention Baseline selects Top-K byte-byte interactions from all layers of the transformer, thereby exerting a stronger influence on the model’s decision process. In contrast, Traffic-Explainer defines fixed pairwise interactions whose importance remains constant across all layers. This design choice, despite being less effective in manipulating model predictions after applying the attention mask, aligns more closely with the intrinsic characteristics of network traffic, where certain byte pairs, such as protocol indicators and IP header fields, consistently co-occur across flows within the same application or country.

5.2.2 Global-Class Explanation. While local-instance explanation is valuable for understanding the model’s decision-making for each network traffic, certain applications demand understanding of a group of instances by Eq. (4). In these cases, identifying common patterns that influence a group can reveal rules or insights that go beyond individual cases. Therefore, we extend our analysis to include global-class level explanations shown in Table 2. In general, our Traffic-Explainer still achieves the best explanation performance. Compared with the instance-level explanation in Table 1, the overall decreasing quality of the class-level explanation as compared to the instance-level explanation in Table 2 might be due to the increased variability inherent in aggregating explanations across a broader set of network traffic sequences.

Object	Explainer		ISCX-VPN				ISCX-nonVPN				ISCX-Tor				ISCX-nonTor			
			Fid	Acc	C-Fid	C-Acc	Fid	Acc	C-Fid	C-Acc	Fid	Acc	C-Fid	C-Acc	Fid	Acc	C-Fid	C-Acc
Byte Level	Saliency Map	1%	35.7	35.7	14.0	15.9	25.9	19.5	27.6	40.2	27.9	26.3	8.10	13.7	24.0	24.3	3.46	4.71
		5%	66.9	69.4	51.6	54.1	30.5	24.1	36.2	42.5	41.5	41.5	23.3	26.8	58.5	58.8	16.5	17.2
		10%	75.2	75.2	56.1	57.3	51.2	48.9	53.5	57.5	43.0	41.3	33.4	34.9	83.4	84.0	29.0	29.3
	Net-Explainer	1%	76.4	76.4	40.8	41.4	16.7	12.6	11.5	21.8	25.3	24.3	4.56	11.1	45.6	46.1	3.92	5.04
		5%	94.9	93.0	68.2	68.2	32.2	24.1	33.9	40.8	51.7	47.1	23.3	26.3	81.2	81.6	29.3	29.3
		10%	98.7	94.9	75.2	75.2	64.9	55.8	74.7	78.2	83.0	78.0	47.3	47.1	91.5	91.2	52.6	52.6

Table 2: Comparison of global class explanation. Traffic-Explainer exhibits higher quality in its generated explanation. We omit results for Random due to its inferior performance in Table 1, and LIME/SHAP due to their focus on local explanations.

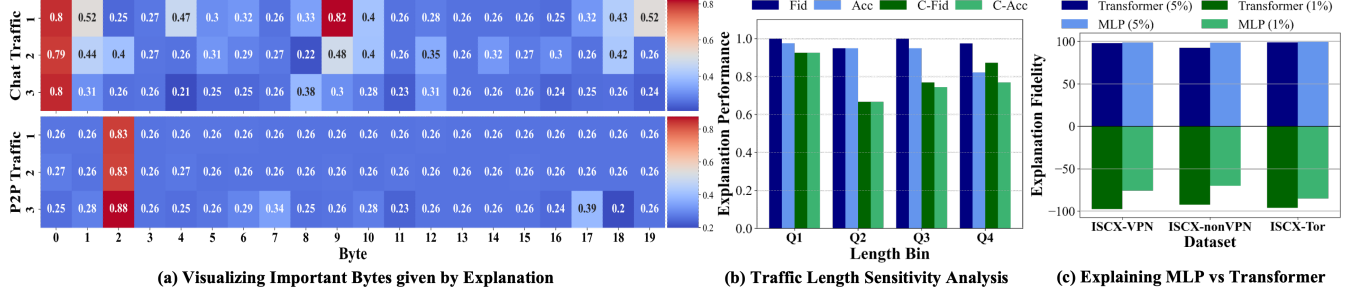


Figure 3: (a) By visualizing the byte importance for each networking traffic sequence in Chat and P2P applications, network operators can assess how the identified critical bytes correspond to the actual content carried by the traffic, thereby validating the predicted application. (b) We observe a slightly decreasing traffic explanation quality as the sequence length grows. (c) Traffic-Explainer achieves consistently higher explanation performance across both MLP/Transformer-based network classifiers.

5.2.3 Explanation Visualization. After quantitatively validating the quality of explanations generated by our proposed Traffic-Explainer, we qualitatively visualize the byte importance scores for three network sequences from Chat and P2P applications. Figure 3(a) illustrates a distinct byte importance pattern, where the 0th/1st/9th bytes are crucial for identifying traffic of application Chat, while the 2nd byte plays the most significant role in distinguishing traffic of P2P application. The explanation also differs across different flow sequences, even though they belong to the same traffic application.

5.2.4 Sensitivity Analysis of Traffic Length. We conduct a sensitivity analysis to evaluate how the length of network traffic sequences impacts explanation quality. Specifically, on the ISCX-VPN dataset, we group traffic sequences into four bins by the total number of bytes (combining both header and payload) for each network traffic sequence and measure the average explanation performance in Figure 3(b). We observe a slight decrease in explanation quality as sequence length increases, although the pattern is not strictly monotonic. We attribute this to the increased difficulty of identifying necessary bytes within longer sequences. As the sequence grows, the larger number of interacting byte pairs makes it harder for Traffic-Explainer to recognize the contribution of specific bytes.

5.2.5 Model-Agnostic Nature of Traffic-Explainer on MLP/Transformer Classifiers. To demonstrate the generality of our Traffic-Explainer, we further apply it to the MLP-based traffic classifier and compare its explanation fidelity against the Transformer-based counterpart under 1%/5% byte budgets across three datasets. In Figure 3(c), Traffic-Explainer achieves similarly high explanation performance between MLP and transformer backbone, confirming its model-agnostic effectiveness.

5.3 Traffic Localization

In this section, we present the second application of Traffic-Explainer: traffic localization. This use case highlights both the interpretability benefits and potential adversarial risks of applying our Traffic-Explainer. When users access the internet from specific geographic regions, such as a particular country, their traffic patterns inherently reflect regional infrastructure and locality characteristics [25]. Traffic-Explainer captures these patterns by automatically identifying the most influential bytes within a traffic sequence that contribute to predicting the country of origin. Following the experimental setting of network application classification in Table 1, we present the explanation performance of traffic localization in Table 4 in Appendix D. Overall, our Traffic-Explainer continues to demonstrate superior explanation performance.

At the global class level, we aim to evaluate the causal and transferable nature of the identified explanations. We perform a byte-swapping experiment across traffic sequences occurring at different country locations. Specifically, we swap the top 10% most important bytes, identified by Traffic-Explainer, between traffic flows belonging to different countries and examine how this affects the output of various classifiers, including Transformer, ET-Bert [21], and MLP. As shown in Figure 4(a), this manipulation consistently causes the classifiers to change their predictions toward the target country associated with the swapped bytes, demonstrating a strong transformation rate. *The success of these transformations, even when using classifiers different from the one used for applying Traffic-Explainer during explanation generation, indicates that the identified bytes are not merely model-specific explanations but causal features that truly characterize country-level traffic patterns.* Note

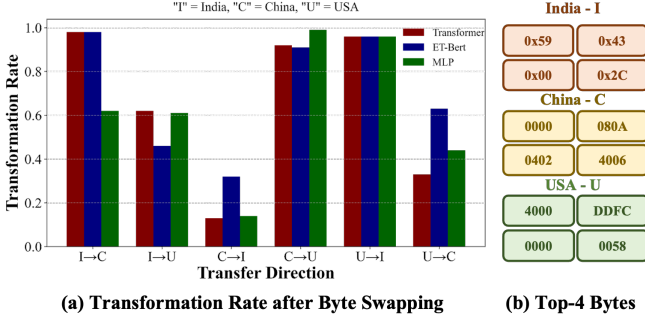


Figure 4: (a) After swapping Top 10% of explanatory bytes by Traffic-Explainer between traffic sequences from different country locations, all classifiers (Transformer/ET-BERT [21]/MLP) exhibit strong class transformation rates. The fact that the transformation also works on models other than the one used for explanation suggests that the identified bytes are truly causal for predicting the country, not just artifacts of the Transformer model. (b) Visualizing Top-4 most important bytes for each country-level traffic class.

that our byte-swapping operation involves exchanging 16-bit hex values between two valid sequences, so the checksums remain correct, and the network traffic is still valid in the real world after swapping bytes. Figure 3(b) visualizes the most influential bytes for representative countries such as India, China, and the USA. These visualizations offer interpretable insights into regional traffic signatures and can inform more precise, geo-aware networking services. Practical applications include region-specific content delivery, targeted advertising, geolocation-based authentication, geofencing, and optimized routing in multiplayer gaming environments [33]. Beyond technical utility, Traffic-Explainer also provides value from a social-good perspective. In regions with government surveillance or censorship, this tool helps users protect their privacy by identifying and obfuscating location-revealing byte patterns. For instance, journalists in sensitive areas could use Traffic-Explainer to deduce information about vulnerable bytes and prevent their actual location from being detected, reducing surveillance risks and cyber-attacks.

5.4 Network Cartography

In this section, we present the third application of Traffic-Explainer: network cartography. Specifically, we aim to validate whether the traceroute-to-submarine cable mapping correctly identifies the key hop in the round-trip time sequence for accurate mapping. As network signals travel through physical cables, the RTT is measured at each hop. Longer physical cables generally result in higher RTTs at the corresponding hops, and submarine cables, which are significantly longer than terrestrial ones, exhibit distinct RTT patterns. This observation leads us to frame the traceroute mapping task as a submarine cable classification problem, where each traceroute sequence is associated with the specific submarine cable it traverses. To enhance transparency, we apply Traffic-Explainer to identify the RTT hops most determinative in the classification decision. These identified hops are cross-validated against the ground-truth RTT sequence: if they align with the hops exhibiting significant RTT spikes, it confirms the reliability of our mapping model.

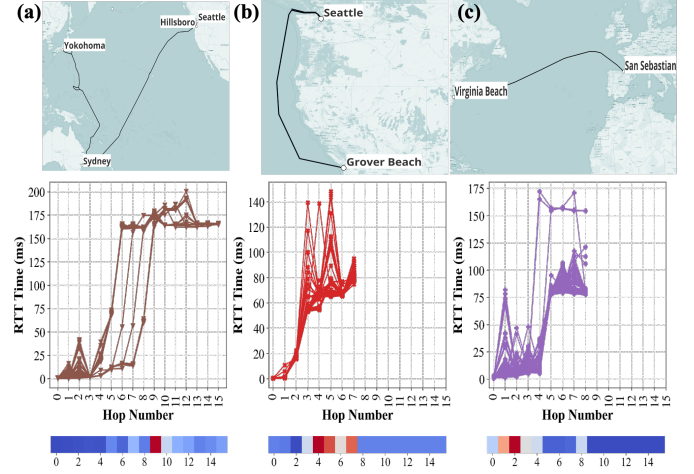


Figure 5: Top - Three distinct submarine cables used for transmitting three groups of traffic traceroute: (a) Seattle, US → Yokohama, Japan; (b) Seattle, US → San Jose, US; (c) Virginia Beach, US → San Sebastian, France. Middle - Sequences of RTT measurements for different traceroute paths. Bottom - Explanation by identifying the most responsible cable for inferring the physical route of a traffic path.

We collaborate with domain experts from Link Oregon to collect 5,000 traceroute measurements traversing 15 distinct submarine cables using tools such as RIPE Atlas and CAIDA Ark [13]. Each traceroute consists of a round-trip time sequence that reflects the physical path of the probe, including both terrestrial and submarine cable segments. Figure 5 highlights three representative traceroutes, with the corresponding RTT sequences shown below each. Notably, the RTT patterns exhibit distinct sequential structures across different submarine cables. Leveraging these patterns, our transformer-based traffic classifier demonstrates exceptional classification performance, as shown in Table 5. To ensure the network cartography (mapping traceroute to physical submarine cable) is based on the distinct RTT spike patterns for each traceroute, we apply Traffic-Explainer to automatically identify the most influential RTT hop responsible for the traceroute mapping. We visualize the heatmap scores for each traceroute hop under the RTT sequence in Figure 5. The darker red regions, indicating higher importance, align well with the RTT spikes. This alignment confirms that our mapping approach successfully captures RTT spikes as discriminative features for characterizing the underlying physical path.

6 CONCLUSION

Despite the successful adoption of deep learning-powered models in network traffic classification, most of them focus on boosting performance without considering the underlying reasons behind their decision-making process, which jeopardizes the transparent adoption of DL solutions by network operators. This motivates us to introduce an explanation framework, Traffic-Explainer, to explain traffic predictions by identifying the important units and unit-unit interactions in given sequences. Extensive real-world evaluations validate the effectiveness of Traffic-Explainer in both local-instance and global-class predictions, demonstrating its potential to advance transparent network management.

REFERENCES

- [1] Mujaheed Abdullahi, Yahia Baashar, Hitham Alhussian, Ayed Alwadain, Nor-shakirah Aziz, Luiz Fernando Capretz, and Said Jadid Abdulkadir. 2022. Detecting cybersecurity attacks in internet of things using artificial intelligence methods: A systematic literature review. *Electronics* 11, 2 (2022), 198.
- [2] Amina Adadi and Mohammed Berrada. 2018. Peeking inside the black-box: a survey on explainable artificial intelligence (XAI). *IEEE access* 6 (2018).
- [3] Nourah Alangari, Mohamed El Bachir Menai, Hassan Mathkour, and Ibrahim Almosallam. 2023. Exploring evaluation methods for interpretable machine learning: A survey. *Information* 14, 8 (2023), 469.
- [4] Julia Amann, Alessandro Blasimme, Effy Vayena, Dietmar Frey, Vince I Madai, and Precise4Q Consortium. 2020. Explainability for artificial intelligence in healthcare: a multidisciplinary perspective. *BMC medical informatics and decision making* 20 (2020), 1–9.
- [5] Scott Anderson, Logman Salamation, Zachary S. Bischof, Alberto Dainotti, and Paul Barford. 2022. iGDB: connecting the physical and logical layers of the internet. *ACM Meas* (2022), 433–448. <https://doi.org/10.1145/3517745.3561443>
- [6] Ahmad Azab, Mahmoud Khasawneh, Saed Alrabae, Kim-Kwang Raymond Choo, and Maysa Sarsour. 2024. Network traffic classification: Techniques, datasets, and challenges. *Digital Communications and Networks* 10, 3 (2024), 676–692.
- [7] Jbodria2023benchmarking Francesco Bodria, Fosca Giannotti, Riccardo Guidotti, Francesca Naretto, Dino Pedreschi, and Salvatore Rinzivillo. [n. d.]. Benchmarking and survey of explanation methods for black box models. *Data Mining and Knowledge Discovery* ([n. d.]).
- [8] Ovidiu Dan, Vaibhav Parikh, and Brian D Davison. 2021. IP geolocation using traceroute location propagation and IP range location interpolation. In *Companion Proceedings of the Web Conference 2021*. 332–338.
- [9] Ovidiu Dan, Vaibhav Parikh, and Brian D. Davison. 2022. IP Geolocation through Geographic Clicks. *ACM Trans. Spatial Algorithms Syst.* 8, 1, Article 2 (March 2022), 22 pages. <https://doi.org/10.1145/3476774>
- [10] Ramakrishnan Durairajan, Paul Barford, Joel Sommers, and Walter Willinger. 2015. InterTubes: A Study of the US Long-haul Fiber-optic Infrastructure. In *Proceedings of the 2015 ACM SIGCOMM Conference*. London, United Kingdom.
- [11] Abdurraheem Elfandi, Hannah Sagaly, Ramakrishnan Durairajan, and Walter Willinger. 2024. Bootstrapping Trust in ML4Nets Solutions with Hybrid Explainability. In *2024 3rd ACM Workshop on Practical Adoption Challenges of ML for Systems*. ACM.
- [12] Qizhang Feng, Ninghao Liu, Fan Yang, Ruixiang Tang, Mengnan Du, and Xia Hu. 2023. Degree: Decomposition based explanation for graph neural networks. *arXiv preprint arXiv:2305.12895* (2023).
- [13] Manaf Gharaibeh, Anant Shah, Bradley Huffaker, Han Zhang, Roya Ensafi, and Christos Papadopoulos. 2017. A look at router geolocation in public and commercial databases. In *Proceedings of the 2017 Internet Measurement Conference*. 463–469.
- [14] Yaru Hao, Li Dong, Furu Wei, and Ke Xu. 2021. Self-attention attribution: Interpreting information interactions inside transformer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 12963–12971.
- [15] Claudia Hauff and Geert-Jan Houben. 2012. Placing images on the world map: a microblog-based enrichment approach. *ACM Meas* (2012), 691–700. <https://doi.org/10.1145/2348283.2348376>
- [16] Thomas Holterbach, Cristel Pelsser, Randy Bush, and Laurent Vanbever. 2015. Quantifying Interference between Measurements on the RIPE Atlas Platform. *ACM Meas* (2015), 437–443. <https://doi.org/10.1145/2815675.2815710>
- [17] Hao Jiang, Yaoqing Liu, and Jeanna N Matthews. 2016. IP geolocation estimation using neural networks with stable landmarks. In *2016 IEEE conference on computer communications workshops (INFOCOM WKSHPs)*. IEEE, 170–175.
- [18] Nektaria Kaloudi and Jingyue Li. 2020. The ai-based cyber threat landscape: A survey. *ACM Computing Surveys (CSUR)* 53, 1 (2020), 1–34.
- [19] Jared Knofczynski, Ramakrishnan Durairajan, and Walter Willinger. 2022. ARISE: A Multitask Weak Supervision Framework for Network Measurements. *IEEE Journal on Selected Areas in Communications* 40, 8 (2022).
- [20] Jundong Li, Kewei Cheng, Suhang Wang, Fred Morstatter, Robert P Trevino, Jiliang Tang, and Huan Liu. 2017. Feature selection: A data perspective. *ACM computing surveys (CSUR)* 50, 6 (2017), 1–45.
- [21] Xinjie Lin, Gang Xiong, Gaopeng Gou, Zhen Li, Junzheng Shi, and Jing Yu. 2022. Et-bert: A contextualized datagram representation with pre-training transformers for encrypted traffic classification. In *Proceedings of the ACM Web Conference 2022*. 633–642.
- [22] Chang Liu, Longtao He, Gang Xiong, Zigang Cao, and Zhen Li. 2019. Fs-net: A flow sequence network for encrypted traffic classification. In *IEEE INFOCOM 2019-IEEE Conference On Computer Communications*. IEEE, 1171–1179.
- [23] Scott M Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. *Advances in neural information processing systems* 30 (2017).
- [24] Scott M Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. *Advances in neural information processing systems* 30 (2017).
- [25] Edward J Malecki. 2002. The economic geography of the Internet's infrastructure. *Economic geography* 78, 4 (2002), 399–424.
- [26] David Pujol Perich, José Rafael Suárez-Varela Maciá, Shihan Xiao, Bo Wu, Alberto Cabellos Aparicio, and Pere Barlet Ros. 2021. Netxplain: Real-time explainability of graph neural networks applied to networking. *ITU Journal on future and evolving technologies* 2, 4 (2021), 57–66.
- [27] Chen Qian, Xiaochang Li, Qinqing Wang, Gang Zhou, and Huajie Shao. 2024. Net-Bench: A Large-Scale and Comprehensive Network Traffic Benchmark Dataset for Foundation Models. *arXiv preprint arXiv:2403.10319* (2024).
- [28] Alagappan Ramanathan and Sangeetha Abdu Jyothi. 2023. Nautilus: A Framework for Cross-Layer Cartography of Submarine Cables and IP Links. *Proc. ACM Meas. Anal. Comput. Syst.* 7, 3, Article 46 (Dec. 2023), 34 pages. <https://doi.org/10.1145/3626777>
- [29] Alagappan Ramanathan and Sangeetha Abdu Jyothi. 2024. Towards Efficient and Scalable Internet Cross-Layer Mapping. In *Proceedings of the CoNEXT on Student Workshop 2024* (Los Angeles, CA, USA). *ACM Meas*, 11–12. <https://doi.org/10.1145/3694812.3699931>
- [30] Alagappan Ramanathan and Sangeetha Abdu Jyothi. 2025. Leveraging Traceroute Inconsistencies to Improve IP Geolocation. *arXiv preprint arXiv:2501.15064* (2025).
- [31] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "Why should I trust you?" Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. 1135–1144.
- [32] Marko Robnik-Šikonja and Marko Bohanec. 2018. Perturbation-based explanations of prediction models. *Human and Machine Learning: Visible, Explainable, Trustworthy and Transparent* (2018), 159–175.
- [33] Krzysztof Rusek, José Suárez-Varela, Paul Almasan, Pere Barlet-Ros, and Albert Cabellos-Aparicio. 2020. RouteNet: Leveraging graph neural networks for network modeling and optimization in SDN. *IEEE Journal on Selected Areas in Communications* 38, 10 (2020), 2260–2270.
- [34] Mario A Sanchez, Fabian E Bustamante, Balachander Krishnamurthy, Walter Willinger, Georgios Smaragdakis, and Jeffrey Erman. 2014. Inter-domain traffic estimation for the outsider. In *Proceedings of the 2014 Conference on Internet Measurement Conference*. 1–14.
- [35] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. 2017. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*. 618–626.
- [36] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. 2020. Grad-CAM: visual explanations from deep networks via gradient-based localization. *International journal of computer vision* 128 (2020), 336–359.
- [37] Satadal Sengupta, Hyojoon Kim, and Jennifer Rexford. 2022. Continuous in-network round-trip time monitoring. *ACM Meas* (2022). <https://doi.org/10.1145/3544216.3544222>
- [38] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. 2013. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034* (2013).
- [39] Payap Sirinam, Mohsen Imani, Marc Juarez, and Matthew Wright. 2018. Deep fingerprinting: Undermining website fingerprinting defenses with deep learning. In *Proceedings of the 2018 ACM SIGSAC conference on computer and communications security*. 1928–1943.
- [40] Vincent F Taylor, Riccardo Spolaor, Mauro Conti, and Ivan Martinovic. 2017. Robust smartphone app identification via encrypted network traffic analysis. *IEEE Transactions on Information Forensics and Security* 13, 1 (2017), 63–78.
- [41] Kedar Thiagarajan, Esteban Carisimo, and Fabián E Bustamante. 2025. The Aleph: Decoding Geographic Information from DNS PTR Records Using Large Language Models. *Proceedings of the ACM on Networking CoNEXT1* (2025).
- [42] Thijs Van Ede, Riccardo Bortolameotti, Andrea Continella, Jingjing Ren, Daniel J Dubois, Martina Lindorfer, David Choffnes, Maarten Van Steen, and Andreas Peter. 2020. Flowprint: Semi-supervised mobile-app fingerprinting on encrypted network traffic. In *Network and distributed system security symposium (NDSS)*, Vol. 27.
- [43] Sahil Verma, John Dickerson, and Keegan Hines. 2020. Counterfactual explanations for machine learning: A review. *arXiv preprint arXiv:2010.10596* (2020).
- [44] Caleb Wang, Ying Zhang, Esteban Carisimo, Qianli Dong, Ram Durairajan, and Fabián E. Bustamante. 2025. Threading the Ocean: Mapping Digital Routes Across Submarine Cables using Calypso. In *ACM SIGCOMM*.
- [45] Qinqing Wang, Chen Qian, Xiaochang Li, Ziyu Yao, and Huajie Shao. 2024. Lens: A Foundation Model for Network Traffic. *arXiv preprint arXiv:2402.03646* (2024).
- [46] Yongjie Wang, Tong Zhang, Xu Guo, and Zhiqi Shen. 2024. Gradient based Feature Attribution in Explainable AI: A Technical Review. *arXiv preprint arXiv:2403.10415* (2024).
- [47] Zhitao Ying, Dylan Bourgeois, Jiaxuan You, Marinka Zitnik, and Jure Leskovec. 2019. Gnnexplainer: Generating explanations for graph neural networks. *Advances in neural information processing systems* 32 (2019).
- [48] Haozhen Zhang, Le Yu, Xi Xiao, Qing Li, Francesco Mercaldo, Xiapu Luo, and Qixu Liu. 2023. TFE-GNN: A temporal fusion encoder using graph neural networks for fine-grained encrypted traffic classification. In *Proceedings of the ACM web conference 2023*. 2066–2075.

- [49] Haiyan Zhao, Hanjie Chen, Fan Yang, Ninghao Liu, Huiqi Deng, Hengyi Cai, Shuaiqiang Wang, Dawei Yin, and Mengnan Du. 2024. Explainability for large language models: A survey. *ACM Transactions on Intelligent Systems and Technology* 15, 2 (2024), 1–38.

A DATASETS

Seven datasets are used to verify our proposed Traffic-Explainer.

- **Application 1 - ISCX VPN/nonVPN** [48]: A public traffic dataset including ISCX-VPN/non-VPN datasets. The ISCX-VPN is collected over virtual private networks (VPNs), used for accessing some blocked websites or services, and is difficult to recognize due to the obfuscation technology. Conversely, the traffic in ISCX-nonVPN is regular and not collected over VPNs.
- **Application 1 - ISCX Tor/NonTor** [48]: ISCX Tor-nonTor is a public dataset, and the ISCX-Tor dataset is collected over the onion router, whose traffic can be difficult to trace. Besides, ISCX-nonTor is also regular and not collected over the onion router.
- **Application 2 - IOS/Android** [27]: The Cross-Platform dataset comprises user-generated data for 215 Android and 196 iOS apps. The iOS apps were gathered from the top 100 apps in the US, China, and India App Stores. The Android apps originate from the top 100 apps in the Google Play Store in the US and India, plus from the top 100 apps of the Tencent MyApp and 360 Mobile Assistant stores, as Google Play is unavailable in China. Each app was executed between three and ten minutes while receiving real user inputs. We use this dataset to evaluate our method’s performance with user-generated data and the performance between different operating systems.
- **Application 3 - Network Traceroute Data**, we collect traceroutes from predefined sources to destinations. Using RIPE Atlas probes [16], we select target countries and direct probes to international servers. Over a three-day period, we collected 5000 unique source-to-destination traceroutes using 15 submarine cables, including Seattle, US to Yokohama, Japan; Seattle, US to San Jose, US; and Virginia Beach, US to San Sebastian, France.

We follow [48] to preprocess the first four synthetic traffic flow datasets and follow [27] to preprocess the last two real-world traffic flow datasets. The comprehensive statistics of each dataset is presented in Table 3. Note that to avoid any overfitting bias, unlike [48], we also add the validation split following ratio: 80%/10%/10%.

B COMPUTATION OF EVALUATION METRIC

As our contributions involve both a scalable traffic classifier and a trustworthy traffic explainer, our evaluation metrics should also consider these two aspects: one for evaluating the traffic classification performance and the other for evaluating the explanation quality. We use Accuracy (Acc) and F1-macro to evaluate the traffic classification performance to avoid the bias caused by the imbalance of traffic flow instances. To evaluate the explanation quality, we first apply Traffic-Explainer to select the top-K important bytes, and then we re-calculate the model predictions with the updated flow sequences by either removing or keeping those selected important bytes. For i^{th} instance, assuming $\mathcal{Y}^{i,*}$ represents its ground-truth label, \mathcal{Y}^i represents its predicted label with the original traffic flow sequence, $\mathcal{Y}^{F,i}$ is its updated prediction after keeping only the Top-K important bytes and masking all others, and $\mathcal{Y}^{CF,i}$ is its updated prediction after removing the Top-K important bytes and keeping

all others. Then, the following four explanation evaluation metrics can be calculated as:

- **Fidelity (Fid)**: The percentage of updated predictions that equal the original model predictions after keeping only the Top-K important bytes: $\text{Fid} = \frac{1}{N} \sum_{i=1}^N \mathbb{I}(\mathcal{Y}^{F,i} = \mathcal{Y}^{i,*})$.
- **Accuracy (Acc)**: The percentage of updated predictions that equal the ground-truth labels after keeping only the Top-K important bytes: $\text{Acc} = \frac{1}{N} \sum_{i=1}^N \mathbb{I}(\mathcal{Y}^{F,i} = \mathcal{Y}^{i,*})$.
- **Counterfactual-Fidelity (C-Fid)**: The percentage of updated predictions that equal the original model predictions after removing the Top-K important bytes: $\text{C-Fid} = \frac{1}{N} \sum_{i=1}^N \mathbb{I}(\mathcal{Y}^{CF,i} = \mathcal{Y}^{i,*})$.
- **Counterfactual-Accuracy (C-Acc)**: The percentage of updated predictions that equal the ground-truth labels after removing the Top-K important bytes: $\text{C-Acc} = \frac{1}{N} \sum_{i=1}^N \mathbb{I}(\mathcal{Y}^{CF,i} = \mathcal{Y}^{i,*})$.

C IMPLEMENTATION DETAILS OF NETWORK CARTOGRAPHY

The Traffic-Explainer is further used to address the traffic mapping challenge in network cartography, where the objective is to establish cross-layer dependencies between logical IP routes and the underlying physical infrastructure. We collaborate with domain experts to collect traceroute data using tools such as RIPE Atlas and CAIDA Ark [13]. After preprocessing, we employ IP geolocation services to associate IP addresses with precise physical locations, incorporating latitude and longitude coordinates [15]. Additionally, we analyze traceroutes that traverse both terrestrial and submarine fiber-optic networks, classifying them based on RTT characteristics. The key insight is that traffic flowing through the same physical link—such as a specific submarine cable—should exhibit similar RTT patterns. Traffic-Explainer helps uncover these patterns, providing a holistic view of how different fiber-optic pathways influence traffic behavior and performance. We detail in four steps below.

Step 1: Data Collection. The data collection process involves gathering traceroutes from predefined source locations to designated destinations. Using RIPE Atlas probes [16], we select target countries and direct probes to a consistent set of international servers. For example, in a case study analyzing transpacific traffic, we selected the United States as the source and Japan as the destination. Over a three-day period, we collected nearly 5000 traceroute data, including Palo Alto, US to Yokohama, Japan; Rockville, Maine, US to Amsterdam, Netherlands; and Vancouver, Canada to Washington, US. These traceroutes traverse both submarine and terrestrial fiber-optic networks, enabling cross-layer analysis of network infrastructure.

Step 2: Data Preprocessing and Geolocation. Once collected, the traceroute data is preprocessed to extract key hop-level details, including hop number, IP address, and source-destination pair. To assess the consistency of multiple traceroutes between the same endpoints, we visualize RTT variations across different probe times (see middle line charts in Figure 5). These visualizations help identify potential deviations in traffic flow and latency variations that indicate the use of different physical network paths – especially submarine cables, as indicated by the sharp increase in RTTs.

For geolocation, we use services such as IPGeolocation, IPLocation.net, and IPinfo to determine the approximate physical location of each hop [9]. The extracted geographic coordinates are then

Dataset	Type	# Train/Val/Test Seq.	# Packet	# Byte	Task	# Label
ISCX-VPN	Header Payload	1,231/154/157	34.20±15.23 24.62±18.99	1,351±612.4 Byte 2,828±2,742 Byte	Application Classification	6
ISCX-NonVPN	Header Payload	3,140/392/395	25.23±15.66 14.88±16.05	1,009±626.3 Byte 1,641±2,057 Byte	Application Classification	6
ISCX-Tor	Header Payload	1,354/169/174	43.03±14.92 40.94±17.02	1,721±596.6 Byte 6,093±2,543 Byte	Application Classification	8
ISCX-NonTor	Header Payload	19,179/2,398/2,400	26.00±16.77 13.60±17.06	1,040±670.7 Byte 1,740±2,411 Byte	Application Classification	8
IOS	Header Payload	776,156/97,803/97,803	/ /	25.74±2.740 Byte 28.27±29.93 Byte	Traffic Localization	3
Android	Header Payload	1,086,909/135,691/135,692	/ /	25.66±2.670 Byte 28.84±31.04 Byte	Traffic Localization	3
Traceroute	RTT	10%/10%/80%	/ /	/ /	Network Cartography	15

Table 3: Statistics of Network Traffic Sequence Datasets. The first four for application classification are from [48], while the next two for traffic localization are collected from [27]. The last one for traceroute is collected using RIPE Atlas probes.

Explainer	Budget	IOS				Android			
		Fid	Acc	C-Fid	C-Acc	Fid	Acc	C-Fid	C-Acc
Random	1%	39.63%	39.46%	2.60%	3.93%	18.12%	18.05%	1.18%	1.62%
	5%	40.71%	40.32%	2.88%	4.07%	24.43%	24.39%	1.81%	2.28%
	10%	50.62%	50.33%	3.79%	5.04%	31.38%	31.09%	2.20%	2.66%
Saliency Map	1%	72.51%	71.83%	15.91%	16.52%	<u>75.42%</u>	<u>75.34%</u>	13.43%	14.09%
	5%	92.64%	91.68%	39.97%	40.40%	90.17%	89.87%	22.30%	22.67%
	10%	95.44%	94.39%	46.64%	46.95%	94.45%	94.03%	25.75%	25.96%
Traffic-Explainer	1%	45.43%	44.47%	<u>5.33%</u>	<u>5.70%</u>	76.85%	76.13%	<u>11.37%</u>	<u>11.46%</u>
	5%	94.61%	92.93%	39.05%	38.96%	96.55%	95.73%	33.39%	33.26%
	10%	97.92%	96.43%	62.28%	62.34%	99.16%	98.57%	44.91%	44.80%

Table 4: Comparison of local-instance explanation at Byte levels for traffic localization task. Our proposed Traffic-Explainer generates explanations of higher quality. Traffic-Explainer achieves better explanation than Saliency Map.

mapped using QGIS, providing a spatial representation of network paths. This geolocation data is crucial for identifying submarine and terrestrial infrastructure used by the traffic, helping establish physical dependencies in network routing.

Step 3: Mapping Traceroutes to Submarine Infrastructure. To map traceroutes to submarine cable infrastructure, we analyze RTT characteristics and geolocation data. First, we identify RTT spikes along traceroutes, as submarine cables typically introduce higher latency due to increased physical distance and transmission delay. Next, we overlay the preprocessed traceroute paths onto QGIS, mapping their geographic coordinates to known submarine cable locations. Additionally, we perform DNS reverse lookups to correlate IP addresses with submarine cable domains, further validating the inferred physical pathways. By linking domain names to submarine cable infrastructure, we identify the most probable submarine routes taken by the traffic, which are collected as a few labeled training sets for instantiating the semi-supervised submarine cable mapping/classification based on RTT characteristics. The

hypothesis is that traffic flowing through the same submarine cable exhibits similar RTT patterns [37]. This classification enables a more refined understanding of how submarine cables influence network performance, congestion, and routing decisions.

Step 4: Traffic-Explainer for Route Interpretation. The mapped traceroute data is further analyzed using Traffic-Explainer, which enhances the interpretation of traffic flow through submarine and terrestrial networks. By overlaying traceroutes onto submarine cable maps, we verify their alignment with known physical routes. Visualizing RTT spikes along the path provides deeper insights into where traffic transitions onto submarine cables, highlighting latency variations across terrestrial vs. submarine network paths.

More broadly, by leveraging Traffic-Explainer, network operators improve network observability and gain a deeper understanding of how submarine cables impact traffic latency, routing behaviors, and performance. This approach significantly enhances their ability to map logical-layer traffic onto the underlying physical infrastructure, advancing the field of network cartography.

Metric	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Correct	1	202	110	226	233	211	235	103	428	94	226	107	237	402	394	469
Total	239	202	110	227	233	213	255	103	428	95	232	107	237	402	397	469
Accuracy (%)	0.42	100.00	100.00	99.56	100.00	99.06	92.16	100.00	100.00	98.95	97.41	100.00	100.00	100.00	99.24	100.00

Table 5: Network cartography performance of traceroute mapping across different source-destination pairs.

D ADDITIONAL RESULTS

D.1 Networking Cartography Performance

We collected approximately 5,000 traceroute measurements over a three-day period across 15 unique source-destination pairs, with each pair associated with a specific submarine cable. To train a traceroute mapping model that classifies a given traceroute to its corresponding source-destination pair, we labeled 5% of the traceroutes within each group and conducted a classification experiment. As shown in Table 5, the model achieves high accuracy, confirming a strong correlation between physical cable paths and logical RTT patterns. These results further motivate our RTT-based explanation learning, where we derive a mask to automatically identify the RTT hops most indicative of the underlying physical cable. In Figure 5, Traffic-Explainer successfully identifies the most characteristic hop, typically corresponding to the largest RTT jump, highlighting the transition point associated with the submarine cable.

E REAL-WORLD IMPLICATIONS

A natural question to consider is: *what are the real-world implications and operational benefits of deploying the Traffic-Explainer framework?* In this section, we describe how Traffic-Explainer provides operational value to network operators and researchers alike across different operational settings. The framework is designed to offer transparent, interpretable insights into deep learning model predictions, thereby closing a critical gap between high-performing but opaque models and the need for accountability and transparency in production network environments [24, 48].

From a general perspective, Traffic-Explainer enhances transparency in deep learning-powered networking systems by providing explanations at both the instance and class levels. For network operators performing monitoring, securing, and optimizing networks, such explanations are essential. In fact, it allows operators to understand not only *what* the model predicts but also *why* it makes that decision [24]. This transparency supports tasks like debugging misclassifications, detecting anomalies, validating compliance with policy or legal requirements, and even auditing model behavior for drift over time. By highlighting the most influential input features (e.g., bytes in packet sequences or RTT hops in traceroute data), Traffic-Explainer acts as a diagnostic tool, empowering operators to make informed, actionable decisions [26].

E.1 Traffic Application Classification

In the context of network application classification, Traffic-Explainer allows operators to identify which specific byte patterns within traffic flows are driving classification decisions. This enables deeper traffic forensics. For example, if a flow is classified as “YouTube” operators can inspect whether the decision was influenced by protocol-specific headers, certificate fingerprints, or payload markers [48].

This insight is valuable for enforcing application-specific policies, detecting evasion techniques (e.g., apps mimicking other protocols), or even discovering previously unknown traffic signatures. The byte-level visibility into model decisions also reduces the reliance on operator know-how (e.g., hand-engineered features), which are traditionally brittle and require constant manual updates.

E.2 Traffic Country Localization

For network country localization, Traffic-Explainer helps surface the regional indicators embedded in traffic data that contribute to country-level classification. This is especially relevant in scenarios where regulatory compliance or geopolitical constraints are in play [17]. For instance, operators can verify whether certain packets are correctly identified as originating from restricted or sanctioned regions, based on explainable byte patterns rather than blind model outputs. Beyond compliance, this capability can also be used to detect traffic masking techniques, such as VPN or proxy usage, and to uncover implicit privacy risks. Importantly, Traffic-Explainer can aid users in understanding which parts of their traffic expose location information, thus enabling the design of more privacy-preserving communication strategies (e.g., privacy-preserving protocols).

E.3 Network Cartography

In the case of network cartography, Traffic-Explainer provides interpretable mappings between logical observations (traceroute sequences) and physical infrastructure (such as submarine cables) [10, 44]. By identifying the RTT hops most responsible for inferring a traffic path’s physical route, Traffic-Explainer enables a new level of visibility into network topology and routing behavior. This can assist operators in diagnosing path anomalies, assessing the resilience of routing paths, and planning for failure scenarios (both benign and malicious). The ability to highlight specific RTT spikes corresponding to cable segments (e.g., transoceanic hops) is particularly useful for submarine risk assessment and capacity planning.

F ETHICAL CONCERN AND MITIGATION

Traffic-Explainer is designed to enhance transparency in network traffic classification, not to function as a surveillance tool. However, its capability to automatically uncover fine-grained packet features and traceroute paths may inadvertently expose sensitive user information. For instance, traffic localization could reveal a user’s approximate geographic location (e.g., the country). Additionally, the discovery of country-specific byte patterns could be adversarially exploited to spoof or falsify location information. These risks are particularly concerning in sensitive regions or under strict regulatory regimes, where misuse could lead to unwarranted surveillance or the de-anonymization of individuals. To mitigate these

concerns, we propose several safeguards: (1) User consent mechanisms to ensure detailed traffic analysis is performed only with explicit permission; (2) Limiting geographic inference granularity, providing only coarse region-level hints rather than precise locations; (3) Masking or omitting identifiable byte patterns from any publicly released outputs; and (4) Institutional oversight, such as

ethics board review or compliance audits, for any system deployment. By implementing these safeguards and adhering to strict ethical standards, researchers and network operators can responsibly leverage Traffic-Explainer’s transparency benefits without compromising user privacy and informing risks.