

Identification of Path Congestion Status for Network Performance Tomography using Deep Spatial-Temporal Learning

Chengze Du^{a,*}, Zhiwei Yu^a, Xiangyu Wang^a

^a*Beijing University of Posts and Telecommunications (BUPT), Beijing, 100876, China*

Abstract

Network tomography plays a crucial role in assessing the operational status of internal links within networks through end-to-end path-level measurements, independently of cooperation from the network infrastructure. However, the accuracy of performance inference in internal network links heavily relies on comprehensive end-to-end path performance data. Most network tomography algorithms employ conventional threshold-based methods to identify congestion along paths, while these methods encounter limitations stemming from network complexities, resulting in inaccuracies such as misidentifying abnormal links and overlooking congestion attacks, thereby impeding algorithm performance. This paper introduces the concept of Additive Congestion Status to address these challenges effectively. Using a framework that combines Adversarial Autoencoders (AAE) with Long Short-Term Memory (LSTM) networks, this approach robustly categorizes (as uncongested, single-congested, or multiple-congested) and quantifies (regarding the number of congested links) the Additive Congestion Status. Leveraging prior path information and capturing spatio-temporal characteristics of probing flows, this method significantly enhances the localization of congested links and the inference of link performance compared to conventional network tomography algorithms, as demonstrated through experimental evaluations.

Keywords: Network Tomography, Path Congestion Status, End-to-End Measurement, LSTM, Adversarial Autoencoders

1. INTRODUCTION

As networks grow in size and complexity [1], sophisticated and minimally invasive tools for network performance assessment become ever more necessary. Network tomography [2] is an innovative approach that leverages end-to-end measurements(e.g., delays or packet loss rates) to infer the status of internal network links by using information from probing between source-end hosts.

Measuring the information of an end-to-end path performance directly determines whether the performance status information of the internal links in the target network can be successfully inferred. When the end-to-end path performance measurement is under perfect error-free conditions, the performance parameters of the internal network links can theoretically be accurately inferred. Nevertheless, in real network environments, end-to-end measurements are influenced by various error factors, forcing the path performance parameters to deviate from their true values with a certain error degree. Therefore, improving the overall detection accuracy of end-to-end paths is necessary to ensure the reliability and effectiveness of inferring the performance parameters of network tomography links and to obtain more accurate path-performance parameter measurement data.

End-to-end path performance parameter measurements can be conducted either through active measurement or

passive observation. Although the latter avoids introducing any detection load cost, its limitations in operability and controllability have promoted active measurement, which is currently more commonly used.

Each method has a varying requirement for end-to-end path performance measurement information. Specifically, early network tomography relied on highly invasive probing to deduce specific properties of links, assuming that if two paths traverse a single, shared lossy link, their measured path loss rates would be equal. For example, network Boolean tomography [3] employed lighter probing methods to infer the Boolean status of links based on rough path information. Such schemes rely on heuristic thresholds based on delay and packet loss performance to determine paths' binary congestion status (congested or none-congested). New frameworks, such as Range tomography [4], utilize multiple detection data from end-to-end paths, e.g., packet loss rates and available bandwidth, to balance probing cost and accuracy.

Based on a simple observation in normal network scenarios, a congested link will cause all paths passing through it to become congested. Therefore, network Boolean tomography extracts the link status by reversing the binary path status through methods like Maximum-Likelihood estimation or greedy inference. Consider the example network shown in Figure 1, where paths $\{y_1, y_2, y_3\}$ traverse through five links $\{x_1, x_2, x_3, x_4, x_5\}$. In traditional Boolean tomography, a congested link causes all paths pass-

*Corresponding Email: ducz0338@bupt.edu.cn

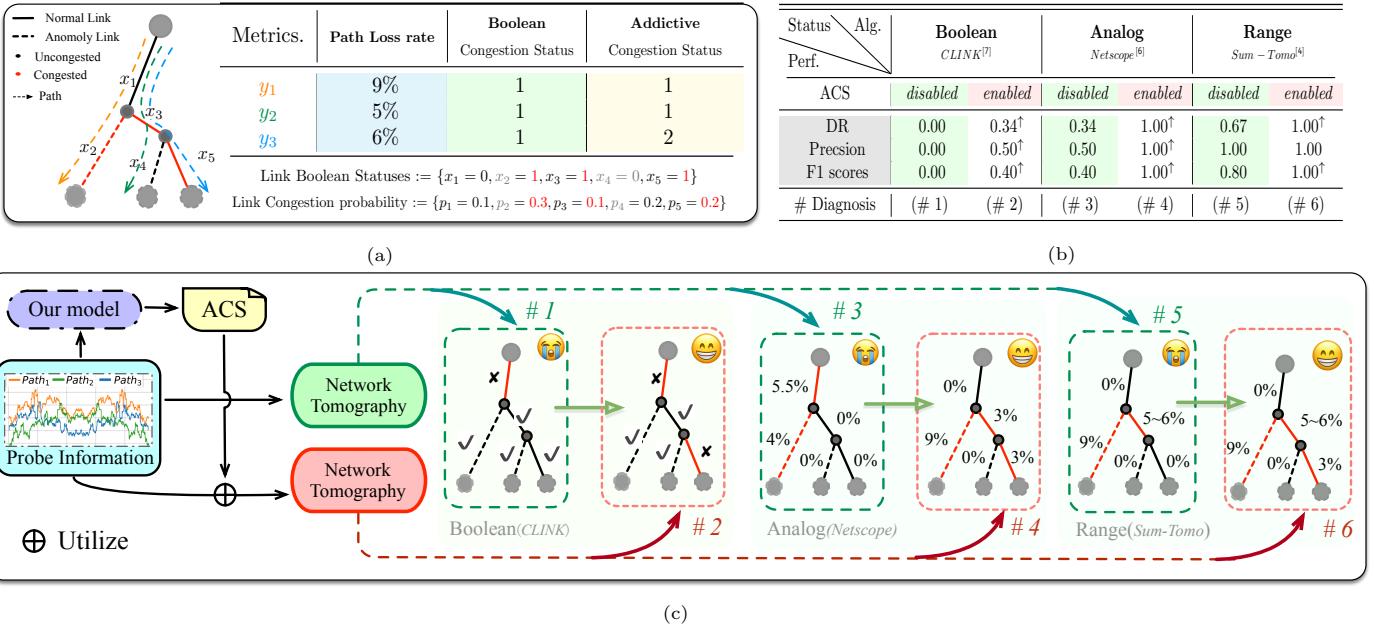


Figure 1: Illustration of Addictive Congestion Status (ACS) based network boolean tomography for congested link identification. (a) A network composed of four links and three paths. The observed status of each path is quantified by their loss rates. (b) Diagnosis performance comparisons of different network tomography schemes. (c) Illustration of the enhancement when applying ACS for path status quantification compared to conventional boolean ones.

ing through it to become congested, leading to binary path states $B = R \vee X$. However, this approach is ill-posed, as multiple sets of congested links (solutions) often match the observed congestion on the end-to-end paths, limiting its diagnostic performance.

It should be noted that once a path traverses multiple congested links, its packet loss rate and communication delay are much higher than those of a single congested link. For instance, in Figure 1(a), path y_1 has a 9% packet loss rate traversing one congested link, while path y_3 shows a 6% loss rate passing through two congested links. By leveraging this difference, we propose Additive Congestion Status $A^+ = R \cdot X$ to quantify the number of congested links in each path. Through deep learning inference that captures both spatial dependencies between paths and temporal patterns in congestion, we can more accurately identify the true congestion pattern. As illustrated in Figure 1(b), this additional information helps reduce the solution space and improves diagnostic accuracy, achieving better precision and recall compared to Boolean approaches.

Based on this, this paper distinguishes and identifies the congestion status of paths by proposing a metric called Addictive Congestion Status and a method to measure it. Qualitatively, the metric encompasses three status: none-congested, single-congested, and multi-congested. Quantitatively, it represents the number of congested links on the path. Furthermore, we develop a new observation framework integrating adversarial autoencoders [5] with Long Short-Term Memory (LSTM) networks to refine the network tomography process. This hybrid approach leverages the adversarial autoencoder's ability to learn complex, non-

linear representations of network paths and the LSTM's proficiency in capturing temporal dependencies within the path measurements. This strategy accurately identifies the Addictive Congestion Status qualitatively and quantitatively. Finally, we validate the contribution of Addictive Congestion Status to existing methods by selecting three classic network tomography algorithms: Netscope [6], CLINK [7], and Sum-Tomo [4], which respectively belong to Analog Tomography, Boolean Tomography, and Range Tomography. The results demonstrate a significant improvement in the accuracy of network tomography techniques regarding congested link localization and link performance inference. The major contributions of our work are summarized as follows:

- **Novel Hybrid Model** We introduce a new method that combines adversarial autoencoders (AAE) with LSTM networks to redefine the Addictive Congestion Status (ACS) and improve network traffic classification and quantification by leveraging spatio-temporal characteristics.
- **Enhanced Network Tomography Performance** The proposed approach significantly improves network tomography, reducing false and missed congestion detections and improving metrics such as recall, precision, F1 score, and normalized root mean square error (NRMSE) in link performance inference.

The remainder of this paper is organized as follows. Section 2 reviews related work in network tomography approaches and deep learning applications. Section 3 introduces the network model and presents our proposed

framework, including the Additive Congestion Status and probing flow mechanism. Section 4 details our methodology for combining LSTM networks with adversarial autoencoders. Section 5 presents our experimental evaluations and comparative analysis. Finally, Section 6 concludes the paper and discusses future research directions.

2. RELATED WORK

Network tomography methods have evolved significantly over time. Analog network tomography uses invasive probing methods to obtain as many messages as possible to solve the under-constrained linear equations between links and paths [8]. Most methods employ additional optimization and constraints to enhance their performance [9; 6; 10; 11; 12]. It should be noted that analog methods impose high requirements in probing and computational resources [3]. Besides, Boolean tomography adopts binary path measurements to deduce the link status ('Bad' or 'Normal') [13; 3; 14; 15]. This method reduces the data volume and privacy intrusions at the expense of measurement precision [16]. However, considering that the adaptability of different network services to different congestion levels is vastly different, some new network tomography frameworks, such as Range tomography, identify the boolean status and the congestion range [4]. Furthermore, some recent papers [17; 18] focus on attack methods targeting network tomography monitoring, which increase the forwarding delay on victim links to manipulate the inferred results of network tomography. Recently, machine learning approaches have shown promising results in network tomography. Several studies [19; 20; 21] employ deep neural architectures like Graph Neural Networks to infer network structures or predict path performance, offering enhanced scalability and adaptability to diverse network conditions. Our combined LSTM-AAE framework significantly improves network tomography accuracy by properly identifying path congestion status through spatio-temporal learning approaches.

Long Short-Term Memory [22] (LSTM) networks are a specialized form of recurrent neural networks uniquely designed to address the challenges associated with learning from data sequences [23]. Their architecture enables them to remember information for long periods, essential for capturing the temporal dependencies inherent in network data. Several existing works have demonstrated the appealing performance of LSTM in prediction and classification problems [24; 25; 26; 27; 28; 29], demonstrating a very low error rate in predicting epidemic diseases [30; 31; 32]. Subsequently, LSTM has been integrated with other modules. For instance, Firzt [33] combined GNN and LSTM to predict COVID-19 cases in Germany. Chen [34] proposed GC-LSTM, a Graph Convolution Network (GC) embedded LSTM for end-to-end dynamic link prediction. Li [29] combined convolutional neural networks (CNN) and long short-term memory to predict the Nev sales and charge infrastructure ownership, which shows the

best performance in using computing power. Sun [28] introduced a deep learning model based on LSTM to improve the shortcomings of existing automatic driving trajectory planning systems, which enhances both the accuracy and safety of the output trajectories. Alternatively, adversarial-LSTM and LSTM-adversarial autoencoders use LSTM to train the latent space produced by autoencoder [35], and our work is different from them.

3. PRELIMINARIES

3.1. Network Model

Network topology is considered an undirected graph $G = (\mathcal{N}, \mathcal{L})$, where nodes \mathcal{N} can be routers, switches, and hosts. The source node is typically the host that sends the data flow, while the edge node usually receives the data flow. The direct connection between nodes is called a link l , and the set of all links in the topology is \mathcal{L} . A data flow from the sending host to the receiving host often passes through multiple nodes, traversing several unknown links. Besides, a path p is an ordered set of links between a pair of source and edge nodes, and the set of all paths in the topology is \mathcal{P} .

To better express the relationship between paths and links, a routing matrix R is employed to elucidate. The routing matrix is a two-dimensional Boolean array of size $|\mathcal{P}| \times |\mathcal{L}|$, where the row coordinates represent path numbers, and the column coordinates are link numbers. If the number in the i -th row and j -th column of R is 1, the i -th path in the network topology includes the j -th link. The set of row coordinates with the number 1 in the n -th column vector represents the set of link coordinates that the n -th path routes through.

Before clarifying the Path Congestion Statues, it is necessary to state the following three assumptions:

1. R does not change during the measurement process, and observations that do not match R are discarded.
2. The states of the links are independent of each other and do not affect each other.
3. The network status remains stable and does not change drastically over a period of time.

3.2. Probing Flow

The probing flow mechanism was initially used to measure a path's available bandwidth by identifying it based on the characteristics of the probing flow. It is categorized into a GAP model and RATE model based on the construction of the probing flow [36], which is achieved by increasing the probing flow size to induce self-congestion while considering the probing flow size at the moment of self-congestion as the available bandwidth. This approach often exacerbates the congestion situation on links.

The probing flow features include packet delay size, variance of all delays, changes in packet gaps, and packet loss rate. This work aims to capture the congestion status of a

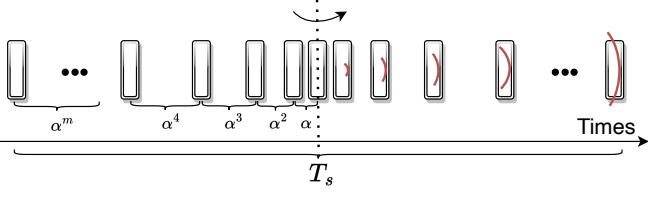


Figure 2: Illustration of the probing flow.

path through these characteristics, namely, how many links along the path are congested. Fig. 2 highlights that the developed framework adopts an exponential packet-sending mechanism where the interval between each packet sent is $\alpha^m, \alpha^{m-1}, \alpha^2, \alpha$. Compared to constant intervals, exponential intervals use the same number of packets but are less intrusive [37]. Thus, we present α to limit the average probing flow rate to 1%. For experimental verification, refer to Section 5.

For each probe packet i , with a send time t_{i_s} and receive time t_{i_r} , we concatenate the packet delay ($t_{i_r} - t_{i_s}$), the ratio of packet reception interval $(t_{i+1_r} - t_{i_r})/(t_{i+1_s} - t_{i_s})$, and the packet loss rate to form the probing flow. Sending M consecutive probe packets constitutes one probing action, which takes time T_s . We conduct N probing actions within the probing period T_P , where $T_P = \sum_i^N T_s$. The values of M and N are chosen to satisfy the requirements of the network state's dynamic changes.

4. METHODOLOGY

4.1. Addictive Congestion Status

The proposed model involves two states for link status: congested and uncongested, with the path state determined by the collective states of all links that comprise it. If any link within a path is congested, the entire path is considered congested. Conversely, a path is none-congested only if all its links are in a none-congested state. We introduce x_i and y_j to represent the boolean states of the i -th link and the j -th path, respectively. Specifically, $x_i = 1$ indicates that the i -th link is congested, and $x_i = 0$ indicates that the link is not congested, while we have

$$y_j \triangleq \begin{cases} 0, & \text{none-congested for } \sum_{l_i \leq p_j} x_i = 0; \\ 1, & \text{congested for } \sum_{l_i \leq p_j} x_i \geq 1, \end{cases} \quad (1)$$

where $l_i \leq p_j$ means the i -th link routed by the j -th path.

For the sake of the qualitative analysis, the **Addictive Congestion Status** (ACS) is introduced and comprises three types: none-congested, single-congested, and multiple-congested. Additionally, the \mathcal{A}_j^+ directly corresponds to the number of congested links in the j -th path and is mathematically formulated as follows:

$$\mathcal{A}_j^+ = \sum_{l_i \leq p_j} x_i, \quad (2)$$

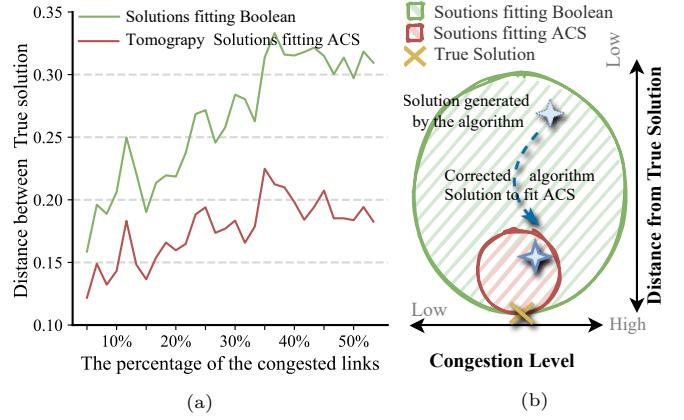


Figure 3: A brief comparison between the boolean and ACS based tomography. (a) Average distance between the tomography result and the true one. (b) Differences between the boolean and the ACS based solution spaces.

while we also derive

$$\mathcal{A}_j \triangleq \begin{cases} 0, & \text{none-congested for } \mathcal{A}_j^+ = 0; \\ 1, & \text{single-congested for } \mathcal{A}_j^+ = 1; \\ 2, & \text{multiple-congested for } \mathcal{A}_j^+ \geq 2. \end{cases} \quad (3)$$

For the sake of discussion, we specifically mark $\mathcal{A} = (\mathcal{A}_1^+, \mathcal{A}_2^+, \mathcal{A}_3^+, \dots)$ as ACS+ for all observed paths.

Let vectors Y and X represent the path and link states, respectively. Then, the Addictive Congestion Status (\mathcal{A}) and Boolean Congestion Status (\mathcal{B}) are calculated as

$$\mathcal{A}^+ = R \cdot X; \quad (4)$$

$$\mathcal{B} = R \bigcup X, \quad (5)$$

where \cdot denotes the dot product and \bigcup is an element-wise logical OR operation between the products of the corresponding elements of the two vectors, and $X = (x_1, x_2, x_3, \dots)$ collects all unknown links' statuses.

Unlike the BCS information (congested or not congested), ACS utilizes more granular path congestion information to assist the existing network tomography algorithms in diagnosing congested links or inferring link performance. Fig. 3b illustrates the proposed scheme, where the green and red areas represent the solution spaces satisfying the path congestion observation constraints and ACS constraints, respectively. It is evident why the red area is smaller than the green area, as satisfying ACS inherently satisfies BCS, but the reverse is not true. If the solution of a network tomography algorithm does not meet the ACS constraints, i.e., it lies outside the red circle, the algorithm must adjust its solution until it meets ACS. When path congestion observations are accurate, the true solution (the actual congestion status of the link) will be at the bottom of the red circle because it satisfies both ACS and BCS constraints and has the smallest distance to itself, which is zero. For this case, distance is defined by

$$d(x, x_i) = 1 - \frac{\sum_{i=1}^n 1(x_i = x'_i)}{n}, \quad (6)$$

where x_i and x'_i are the true i-th link state and the algorithm's estimated i-th link state, respectively, and n is the number of links in the topology. Through simulations conducted under different topologies and congestion probabilities (see 5.1 for experimental setup). As depicted in Fig. 3a, the average distance between all solutions within the ACS-constrained solution space and the true solution is smaller than the average distance between all solutions that satisfy only the BCS constraint but not the ACS constraint and the true solution. This infers that solutions meeting the ACS constraint are closer to the true solution than those satisfying only the BCS constraint.

Proposition 1. $\forall \mathbf{a}, \exists \mathbf{b}',$ such that

$$d(\mathbf{a}, \mathbf{b}') < d(\mathbf{a}, \mathbf{b}) \implies \mathcal{M}(\mathbf{a}, \mathbf{b}') \geq \mathcal{M}(\mathbf{a}, \mathbf{b})$$

$\mathcal{M}(\mathbf{a}, \mathbf{b})$ represents the performance metrics (such as recall, precision, and F1-score).

Proof: If b' has a smaller distance to a than b : $d(a, b') < d(a, b)$.

This implies: $\sum_{i=1}^n 1(a_i = b'_i) > \sum_{i=1}^n 1(a_i = b_i)$.

In order to understand how this impacts the performance metrics, consider the implications on True Positives (TP), False Positives (FP), and False Negatives (FN): b' has more elements where $a_i = b'_i = 1$; therefore: $TP_{b'} \geq TP_b$.

Similarly, for FP and FN: $FP_{b'} \leq FP_b$, $FN_{b'} \leq FN_b$.

Given these relationships:

Recall increases or remains the same as TP increases and FN decreases. $Recall_{b'} = \frac{TP_{b'}}{TP_{b'} + FN_{b'}} = \frac{1}{1 + \frac{FN_{b'}}{TP_{b'}}} \geq \frac{1}{1 + \frac{FN_b}{TP_b}} = \frac{TP_b}{TP_b + FN_b} = Recall_b$.

Precision increases or remains the same as TP increases and FP decreases. $Precision_{b'} = \frac{TP_{b'}}{TP_{b'} + FP_{b'}} = \frac{1}{1 + \frac{FP_{b'}}{TP_{b'}}} \geq \frac{1}{1 + \frac{FP_b}{TP_b}} = \frac{TP_b}{TP_b + FP_b} = Precision_b$.

F_β -score increases or remains the same since it is the harmonic mean of precision and recall, both of which have increased or remained constant. $F_\beta\text{-score}_{b'} = \frac{(1+\beta^2) \cdot Precision_{b'} \cdot Recall_{b'}}{\beta^2 \cdot Precision_{b'} + Recall_{b'}} = \frac{1}{\frac{\beta^2}{(1+\beta^2) \cdot Recall_{b'}} + \frac{1}{(1+\beta^2) \cdot Precision_{b'}}} \geq \frac{1}{\frac{\beta^2}{(1+\beta^2) \cdot Recall_b} + \frac{1}{(1+\beta^2) \cdot Precision_b}} = \frac{(1+\beta^2) \cdot Precision_b \cdot Recall_b}{\beta^2 \cdot Precision_b + Recall_b} = F_\beta\text{-score}_b$.

The above validation demonstrates that the algorithm performs better when the ACS constraint is included. ■

4.2. Long Short-Term Memory Network

The LSTM network, a specific type of the Recurrent Neural Network (RNNs), operates by passing information from previous time steps to the current time step. LSTM has demonstrated impressive performance in handling spatio-temporal sequence problems. Its cell structure is meticulously designed to overcome the vanishing or exploding gradients problem that RNNs face when processing long sequences.

The developed scheme considers the number of probing actions completed within the probing time T_P as the number of time steps in the LSTM network and the amount

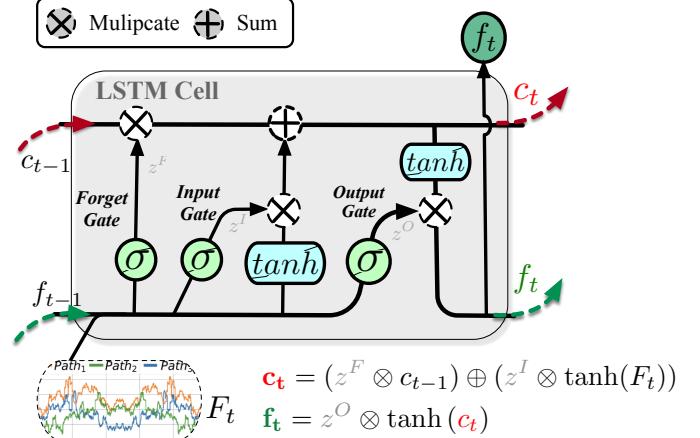


Figure 4: Illustration of the LSTM cell.

of data collected during each probing action as the length of each time step. All observational data are normalized before being input into the model for training.

The LSTM network primarily comprises three stages:

- Forget Gate:** Used for selectively forgetting the output from the previous time step, c_{t-1} . Its function can be interpreted as "forgetting the unimportant". z^F acts as the forget gate control.
- Input Gate:** Selectively remembers the input for the current time step, F_t . Its function can be interpreted as "remembering the important". z^I acts as the memory gate control, participating in the update of the cell state to obtain the new state c_t .
- Output Gate:** Determines the output value f_t based on the cell state at the current time step.

4.3. Adversarial Autoencoder

Adversarial Autoencoders (AAE) [5] combine Autoencoders (AE) [38] and Generative Adversarial Networks (GANs) [39] to learn target data distributions. Autoencoders use an encoder to compress input data into a low-dimensional representation and a decoder to reconstruct the original data, minimizing reconstruction error [40]. GANs consist of a generator producing realistic fake data and a discriminator distinguishing real from fake data [41], improving data quality through adversarial training. AAEs leverage these mechanisms for effective data representation and generation.

Based on the data obtained from the detection flow, \mathbf{F} , the feature equation of the adversarial autoencoder based on LSTM is defined as: $f(\mathbf{F}) = \tilde{\mathbf{F}}$. In the autoencoder, the encoder and decoder are mirrors of each other, i.e., $\tilde{\mathbf{F}} = D(E(\mathbf{F}))$. Considering that the congestion characteristics of each time slot often correlate with the congestion characteristics before and after it, LSTM is employed as a part of the encoder and decoder. This strategy effectively encodes input data with temporal dependencies into the latent space, where \mathcal{F}' and \mathcal{F}'_{cat} are compressed representations of \mathbf{F} from the latent space. We incorporate the

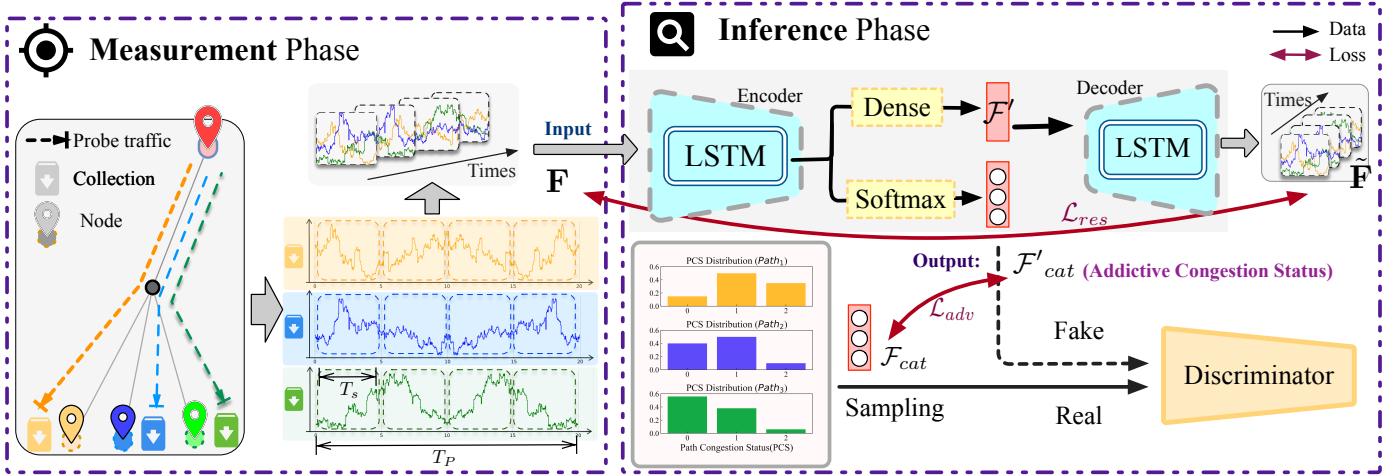


Figure 5: Overview of our proposed scheme. It has two phases: one is illustrated on the left side, for the process of end-to-end measurements and data collection; the other is depicted on the right side, for the ACS path status identification/inference via AAE-LSTM based deep learning. **Left:** The Measurement Phase collects probe traffic data over period T_P with measurement windows T_s . **Right:** The Inference Phase processes this data through an encoder-decoder LSTM structure and a discriminator, outputting both reconstructed probe data (\mathcal{F}) and Addictive Congestion Status (\mathcal{F}_{cat}), guided by loss functions \mathcal{L}_{res} and \mathcal{L}_{adv} .

GAN into the autoencoder to regulate the latent space's data distribution. This ensures that the distribution of the encoder's encoded results conforms to the distribution of Addictive Congestion Status information.

The training process of the proposed method is divided into the following two parts.

Reconstruction Process: The autoencoder is trained by minimizing the difference between the input and the reconstructed output (e.g., using reconstruction error). Multiple LSTM layers are stacked in the encoder portion, where LSTM learns the temporal dependencies in the sequence through its internal state (hidden state and cell state), thereby converting the input sequence data into a fixed-size latent representation [42]. Then, the decoder decodes this latent representation back into the original data space. Besides, the compressed vector from the encoder passes through the dense and softmax layers, generating continuous vectors \mathcal{F}' and probability distribution vectors \mathcal{F}'_{cat} , respectively. Although both vectors possess the characteristics of observational data, \mathcal{F}' is used for subsequent data reconstruction, while \mathcal{F}'_{cat} is utilized for classification and specifically for ACS recognition.

Adversarial Process: The latent representation is input to the adversarial network, which aims to differentiate between the distribution of the latent representation and the target distribution. The encoder in the autoencoder acts as the generator, attempting to produce latent representations that match the target distribution [43]. At the same time, the adversarial network tries to distinguish whether these representations are from the target distribution. The adversarial process aims to make the distribution of \mathcal{F}'_{cat} approach the true distribution of Addictive Congestion Status. Both the reconstruction loss and adversarial loss are defined as follows:

$$\begin{cases} \mathcal{L}_{D^1}^{adversarial} = \mathcal{L}_{\mathcal{F}_{cat} \sim \mathcal{N}}^{bce}(\mathcal{D}^1(\mathcal{F}_{cat}), 1)) + \\ \quad \mathcal{L}_{\mathcal{F}'_{cat} \sim q(\mathcal{F}'_{cat})}^{bce}(\mathcal{D}^1(\mathcal{F}'_{cat}), 0)); \\ \mathcal{L}_{LSTM-AAE}^{reconstruction} = \mathcal{L}_{\mathcal{F} \sim q(\mathcal{F})}^{bce}(\mathcal{D}^A(\mathcal{F}), 1)) + \mathcal{L}_{mse}(\tilde{\mathbf{F}}, \mathbf{F}). \end{cases}$$

These losses will be used to guide the training of our proposed AAE-LSTM neural network as depicted in Fig. 5, as similar to the parameter estimation of conventional regression problems like curve fitting.

4.4. Overview of ACS-based Network Tomography

Figure 5 illustrates our proposed two-phase framework for network congestion identification. In the measurement phase, probe traffic is sent through different network paths to collect path performance metrics over time T_P , with each probing action taking time T_s . The inference phase then processes this data through our deep learning architecture. Specifically, the collected probe information \mathbf{F} is first fed into an LSTM encoder to capture temporal dependencies. The encoded features then pass through two parallel branches: one branch uses a dense layer followed by a decoder to reconstruct the input data for training stability, while the other branch employs a softmax layer to generate the Addictive Congestion Status classification. In the adversarial training process, we sample from the true link congestion probability distribution to obtain real ACS labels, and train the discriminator to distinguish between these real samples and the predicted ACS from our model. This encourages the model to generate ACS predictions that match the underlying network congestion probability distribution, improving the accuracy of congestion status identification.

Algorithm 1: Adversarial LSTM-based Addictive Congestion Status (ALACS)

```

Input : routing matrix  $\mathbf{R}$ , path set  $\mathbf{P}$ 
        number of paths  $P$ , probing times  $\mathcal{N}$ 
/* Monitor Paths' Status */
1 Infer the sets of Paths' Probing Information  $\mathbb{F} \leftarrow \emptyset$ ;
2 for  $t=1$  to  $\mathcal{N}$  do
3   foreach  $p_j \in \mathcal{P}$  do
4      $F_j(t) \leftarrow$  Probing and collecting the
      Information ;
5    $\mathbb{F} \leftarrow [\mathbb{F}, \{F_j(t)|j = 1, \dots, P\}]$ ;
/* Calculate Paths' Congestion Status */
6 Initialize the sets of Paths' Congesiton Status  $\mathbb{S} \leftarrow \emptyset$ ;
7 foreach  $p_j \in \mathcal{P}$  do
8    $A_j \leftarrow$  Obtain the Addictive Congestion Status
      with model by  $\{F_j(t)|t = 1, \dots, \mathcal{N}\}$ ;
9    $\mathbb{A} \leftarrow [\mathbb{A}, A_j]$ ;
/* Infer the links' Performance */
10 Initialize the sets of links' Performances by  $\mathbb{L} \leftarrow \emptyset$ ;
11  $\mathbb{L} \leftarrow$  Network Tomography( $\mathbb{F}, \mathbf{R}, \mathbb{A}$ ) ;
Output:  $\mathbb{L}$ 

```

5. Numerical Evaluations

This section demonstrates the impact of different detection flow settings on the experimental results and then evaluates the algorithm's ACS identification from both qualitative and quantitative perspectives. Finally, model ablation experiments prove the indispensable roles of LSTM and Adversarial Autoencoders.

5.1. Network Setups

All experiments were conducted with real network topologies obtained from TopologyZoo [44], aiming to assess the accuracy and reliability of identifying ACS. Each network topology had a distinct structure, such as varying path lengths and numbers of paths 1. The network event simulations were conducted with NS-3 (version 3.27) on a server equipped with Intel Platinum 8383C Official Edition CPU 40-Cores running Ubuntu 20.04 LTS. The duration of congestion in each simulation instance was set to 5 minutes. All experimental data and related code are available online¹.

During the experiments, it was assumed that each link in the topology has a certain congestion probability, which is independent. Less than 20% of the available bandwidth was used to determine whether a link is congested, i.e., when the available bandwidth of a link is between 100% and 80%, the link's status is considered normal. The link is considered congested when the available bandwidth

Table 1: Characteristics of four real networks used for numerical simulations.

Metrics \ Network	CHINANET	AGIS	GEANT	ERNET
#Paths	17	14	15	12
#Links	21	18	17	13
Average.Hops	3.9	3.6	3.6	3.25
Average.Weights	4.3	2.8	3.1	3

is less than 20%. The experiments were conducted on two topology types: *Homogeneous Topology*, where all links in the network have the same propagation delay and bandwidth, both set at 20ms and 20Mbps, respectively and *Heterogeneous Topology*, where the properties of links in the network vary. The propagation delay and bandwidth values follow a uniform distribution between 20 and 25. Unless specified otherwise, each scenario in the experiments, comprising the detection flow settings and the network topology, is repeated 40 times.

The dataset used in this research consists of end-to-end network measurements collected across multiple TopologyZoo network topologies (CHINANET, AGIS, GEANT, and ERNET). We systematically varied the congestion probability parameter (0.1-0.9) to create diverse network conditions, collecting path-level metrics including delay distributions, packet loss rates, and throughput measurements. The dataset was split into training (80%) and testing (20%) sets, with a validation subset used for hyperparameter tuning. To address class imbalance, we employed balanced sampling during model training. The LSTM-AAE model was trained using Adam optimizer (learning rate 0.001) with dropout regularization (0.5) and early stopping to prevent overfitting. Each scenario was repeated 40 times under different network conditions, resulting in approximately 3,000 samples per topology.

5.2. Evaluations of ACS

5.2.1. Optimizing Probing Flow

Prolonged monitoring and excessive probing rates can negatively impact network performance, while short-duration and low-rate probes may not effectively capture relevant network characteristics. We configured the probing flows regarding *probing duration ratio*, *Probe Traffic Bandwidth*, and *windows number*, aiming to find an optimal set of probing settings to balance accuracy and intrusiveness.

Fig. 6 reveals the impact of the *Probe Traffic Bandwidth Ratio* (obtained by comparing the probing bandwidth to the end-to-end path bandwidth) and windows number on the accuracy and stability of Addictive Congestion Status identification. The area above indicates high accuracy (the darker the color, the higher the accuracy), and the area below represents the coefficient of accuracy variation (the lighter the color, the better the stability). It is evident that intense probing flows negatively affect background traffic by causing congestion, while too weak probes fail to capture congestion characteristics. Additionally, an excessive

¹<https://github.com/Monickar/ACS>

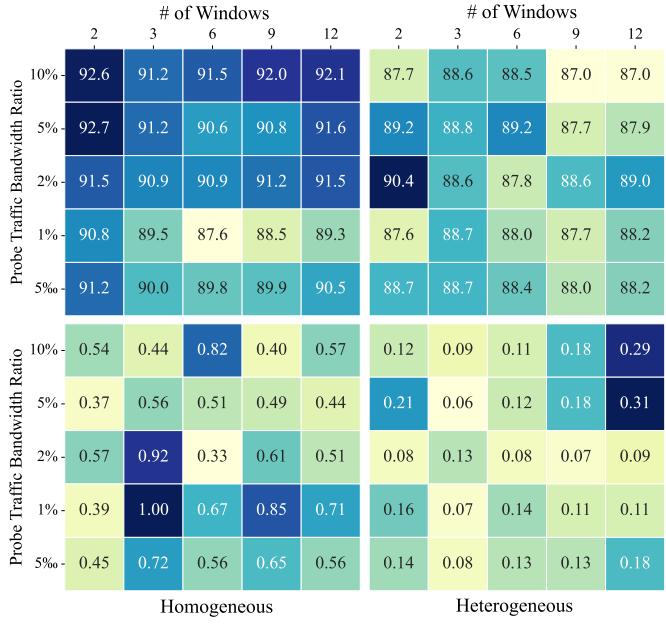


Figure 6: Accuracy of ACS identification under different probing conditions. Experiments were conducted in both homogeneous and heterogeneous network setups.

probe step size causes issues like data loss and gradient explosion in the LSTM model, while too small step sizes limit the LSTM’s learning capability due to insufficient information, leading to poor training outcomes, low accuracy, and stability. Notably, compared to homogeneous topology, heterogeneous topology is more challenging in identifying Addictive Congestion Status.

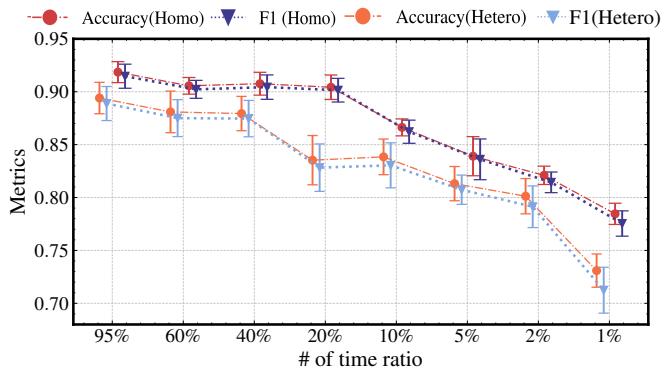


Figure 7: Accuracy and F1 scores of ACS-based congestion diagnosis under varying probing duration ratios. The experiments are conducted in both homogeneous (Homo) and heterogeneous (Hetero) network environments, showing the impact of observation time on model performance.

Fig. 7 illustrates the effect of probing duration on the accuracy and stability of ACS using the accuracy and F1 score metrics to minimize the impact of sample imbalance on the results. Spherical markers represent accuracy, and triangular markers represent the F1 scores. The results reveal that as the probing duration decreases, accuracy and balance indices drop significantly when the observation

time is less than 40%. However, this is understandable, as longer probing times capture more congestion feature information. Regarding the balance between intrusiveness, cost, and accuracy, the probing flow settings were 40% probing duration, 1% of average bandwidth for probe intensity, and a step size of 6.

5.2.2. Categorized evaluation

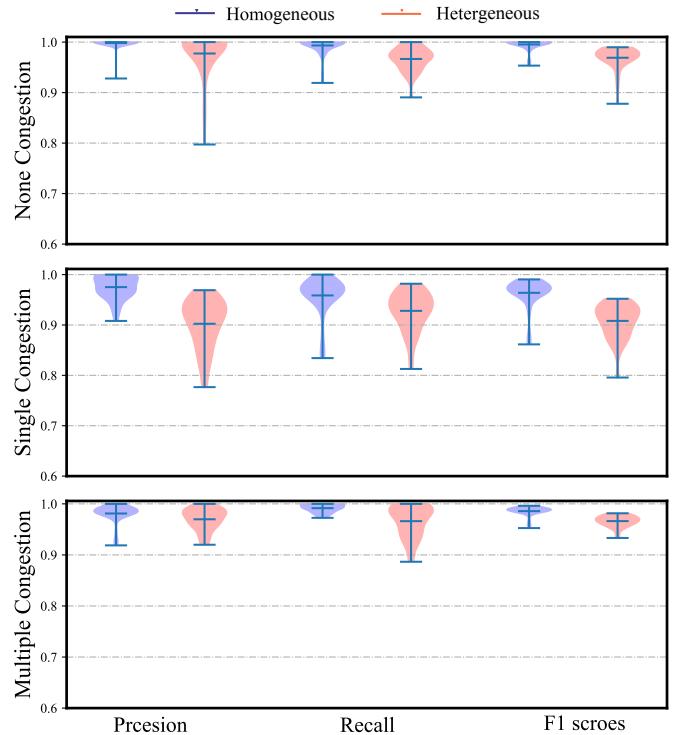


Figure 8: Distributions of precision, recall, and F1 scores across three categories of congestion: None, Single, and Multiple. The experiments are conducted in both homogeneous and heterogeneous network topologies.

Congestion detection was considered a three-class problem, and we employed balanced sampling techniques for model training. The proposed model was qualitatively evaluated in both homogeneous and heterogeneous network scenarios. The corresponding results are presented in Fig. 8, highlighting that our model exhibits high classification performance, with precision, recall, and average F1 scores all reaching or exceeding 0.95. However, compared to homogeneous topologies, heterogeneous topologies are more challenging for the model due to the diversity of link properties, resulting in lower performance across all metrics. It should be noted that the model demonstrates higher accuracy in identifying none-congested situations. However, distinguishing congestion further presents challenges for the model.

5.2.3. Quantitative evaluation

As the path lengths and the complexity of heterogeneous topologies increase, accurately identifying the num-

ber of congestions in a path becomes challenging. For instance, an algorithm might identify five congested links in a path where there are actually six. Hence, to demonstrate identification effectiveness, we use Relative Accuracy, with Absolute Accuracy and Relative Accuracy denoted as

$$\frac{\text{Number of Correct Classifications}}{N} \quad \text{and} \quad \frac{\sum_{i=1}^N \left(1 - \frac{|y_i - \hat{y}_i|}{\max(Y) - \min(Y)}\right)}{N},$$

respectively.

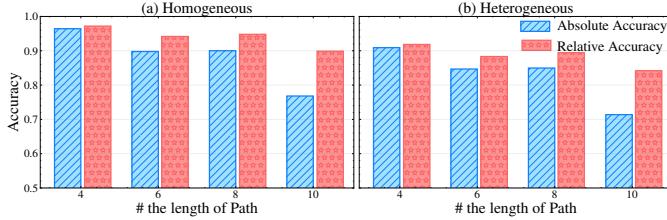


Figure 9: Estimation accuracy comparison between absolute path lengths (counting exact number of congested links) and relative path lengths (normalized distance from true value) from both homogeneous (a) and heterogeneous (b) network setups. The x-axis represents different path lengths ranging from 4 to 10 hops.

Fig. 9 illustrates the relationship between the Absolute and Relative accuracy and path length in homogeneous and heterogeneous topologies. As path length increases, the model’s absolute accuracy decreases, owing to the increased classification difficulty linked to the growing number of categories. However, the relative accuracy experiences a smaller decline, indicating that the model maintains a certain level of stability even as the data patterns become more complex.

5.2.4. Ablation Studies

To validate the effectiveness of our proposed framework, we conduct comprehensive ablation experiments comparing three model variants: LSTM-only, AAE-only (replacing LSTM with fully connected layers), and our combined LSTM-AAE approach. Table 2 presents the comparison results across multiple metrics.

In homogeneous networks, while LSTM-only achieves 89.0% accuracy by capturing temporal patterns and AAE-only reaches 80.2% accuracy through distribution learning, our combined approach significantly improves the performance to 96.3%. This enhancement demonstrates how LSTM’s temporal modeling complements AAE’s distribution learning. Specifically, LSTM helps track congestion patterns over time, while AAE ensures the predicted congestion status matches the underlying network probability distribution.

The advantage of our combined approach becomes even more prominent in heterogeneous networks, where single-model approaches show notable performance degradation (LSTM-only: 77.8%, AAE-only: 67.9%). In contrast, our LSTM-AAE framework maintains robust performance with 91.4% accuracy. This demonstrates that the temporal dependencies captured by LSTM and the distribution learning from AAE work synergistically to handle complex network

conditions, addressing limitations of using either component alone.

Table 2: Ablation results from both homogeneous and heterogeneous setups of link prior congestion probability.

	AAE	LSTM	Precision	Recall	F1-score	ACC(%)	R-Acc(%)
Homo.	✓		0.939 \pm 0.03	0.946 \pm 0.03	0.936 \pm 0.04	89.0 \pm 5.3	93.0 \pm 3.1
	✓		0.927 \pm 0.09	0.935 \pm 0.07	0.926 \pm 0.08	80.2 \pm 9.9	89.4 \pm 6.9
	✓	✓	0.984\pm0.01	0.981\pm0.02	0.981\pm0.01	96.3\pm2.4	97.8\pm1.4
Hetero.	✓		0.829 \pm 0.03	0.833 \pm 0.02	0.816 \pm 0.03	77.8 \pm 3.4	83.5 \pm 3.2
	✓		0.736 \pm 0.11	0.766 \pm 0.07	0.712 \pm 0.13	67.9 \pm 8.9	78.5 \pm 9.0
	✓	✓	0.949\pm0.02	0.953\pm0.01	0.947\pm0.02	91.4\pm3.0	94.1\pm2.3

5.3. Performance Improvement

We verified the performance improvement of various algorithms constrained by the solution space using ACS in both congested link diagnosis and link performance inference (packet loss rate). Among the competitor algorithms, CLINK [7] only has the function of congested link diagnosis. Sum-Tomo [4] and Netscope [6] first infer the link performance based on the observation information of end-to-end paths and then diagnose congestion based on the inferred link performance. For example, if a link’s inferred packet loss rate exceeds 1%, the link is considered congested.

In congested link diagnosis, we measure the diagnostic performance of the algorithms using recall, precision, and F1 score. The x-axis represents different scenarios of link congestion probabilities. The green, red, and blue plots represent the CLINK, Sum-Tomo, and Netscope algorithms, respectively. The dashed lines are the performance results without using ACS for the solution space constraints. The smooth solid lines represent the performance results of algorithms using qualitative ACS, and the solid lines with cross markers represent the performance results of algorithms using quantitative ACS.

Fig. 10 highlights that the performance of algorithms without using ACS for solution space constraint significantly declines as the congestion probability increases. Both the Sum-Tomo and Netscope aim to minimize the number of congested links, leading to potential miss-detections in high-congestion scenarios and consequently decreasing recall, resulting in a final decrease in the F1 score. In contrast, CLINK, as a greedy strategy based on the Maximum A Posteriori (MAP) algorithm that aims to find the most probable set of congested links according to the prior congestion probabilities of links, increases slightly in performance when the average congestion probability in the scenario exceeds 0.5. This is because its expected number of congested links increases in the congestion probability interval of [0.5-0.9] to satisfy the MAP criterion, thereby reducing miss-detections in high congestion scenarios and increasing recall.

Furthermore, as the link congestion probability increases, the performance improvement due to ACS also increases. This is because ACS helps algorithms address the pain points of uncertainty in network tomography by trimming

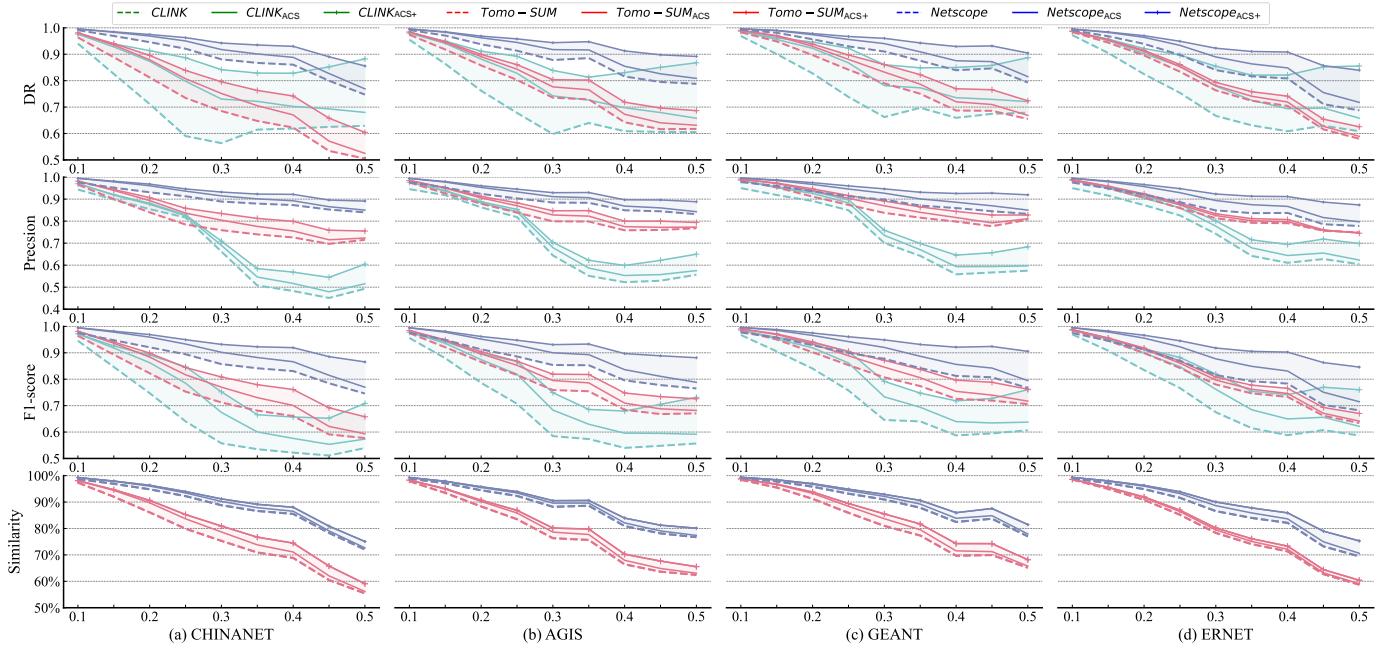


Figure 10: Performance comparisons among different network tomography algorithms under various link congestion probabilities. The subscript "ACS" and "ACS+" indicate qualitative ($A_j \in \{0, 1, 2\}$) and quantitative ($A_j^+ \in \{0, 1, 2, 3, \dots\}$) ACS inputs respectively. Results are shown across four real network topologies: (a) CHINANET, (b) AGIS, (c) GEANT, and (d) ERNET.

the solution space, allowing the algorithms to avoid selecting sets with fewer congested links in scenarios with high congestion probabilities. Thus, DR increases with increasing link congestion probability. Notably, the precision of the algorithms (diagnosing congested links that are indeed congested) also increases because the solution space reduces, ensuring the algorithms reduce their miss-detections while maintaining precision, thereby significantly enhancing the overall F1 score.

Regarding link performance inference, this paper represents the estimation error by using the normalized root mean square error (NRMSE), which measures the similarity between two signals. For the true link performance $y[m]$ and inferred link performance $\hat{y}[m]$, their NRMSE is defined as follows:

$$\text{NRMSE} = \sqrt{\frac{\sum_{m=0}^{M-1} |y[m] - \hat{y}[m]|^2}{\sum_{m=0}^{M-1} |y[m]|^2}}$$

Netscope and Sum-Tomo present smaller estimation errors and a specific link performance in Fig. 10. However, when ACS is inaccurate, its performance enhancement effect is understandably reduced, as incorrect ACS information introduces biases in trimming the solution space, potentially preventing the algorithms from finding solutions closer to the actual situation.

6. CONCLUSION & FUTURE WORK

This paper has addressed the inherent challenges in network tomography, particularly the accurate identification of

Additive Congestion Status, by combining adversarial autoencoders with LSTM networks. The proposed framework leverages the spatio-temporal characteristics of network traffic, offering a robust solution for classifying and quantifying the Additive Congestion Status. Our approach significantly enhances the network tomography's precision, mitigating the impact of anomalous links on performance assessments while ensuring minimal invasiveness.

For future work, we plan to explore four main directions: (1) extending our framework to handle dynamic network topologies where routing paths change over time, (2) developing a lightweight version of our model to enable real-time congestion detection in resource-constrained environments, and (3) incorporating transfer learning techniques to adapt our pre-trained models to new network architectures with minimal retraining, and (4) conducting comprehensive validation using real-world network datasets to rigorously evaluate model accuracy across diverse operational environments.

Experimental results demonstrate the efficacy of our method in reducing false and missed congestion detections, thereby validating its contribution to improving network performance evaluations. Specifically, our approach shows significant improvements over traditional network tomography techniques including CLINK (Boolean tomography), Netscope (Analog tomography), and Sum-Tomo (Range tomography), achieving better precision in congested link localization and more accurate link performance inference. By capturing the dynamic nature of network traffic, our AAE-LSTM framework provides a more robust solution for network congestion detection and quantification compared

to these conventional approaches.

Acknowledgments

We thank our colleagues and mentors for their valuable guidance during this research. We also appreciate the support from the High-performance Computing Platform of BUPT for providing the resources necessary to complete this work.

References

- [1] Muyi Sun, Bingyu He, Ran Li, Jinhua Li, and Xinchang Zhang. A survey: Network feature measurement based on machine learning. *Applied Sciences*, page 2551, Feb 2023.
- [2] Rui Castro, Mark Coates, Gang Liang, Robert Nowak, and Bin Yu. Network tomography: Recent developments. *Statistical Science*, Aug 2004.
- [3] Nick Duffield. Network tomography of binary network performance characteristics. *IEEE Transactions on Information Theory*, page 5373–5388, Dec 2006.
- [4] Sajjad Zarifzadeh, Madhwaraaj Gowdagere, and Constantine Dovrolis. Range tomography: combining the practicality of boolean tomography with the resolution of analog tomography. In *Proceedings of the 2012 Internet Measurement Conference*, pages 385–398, 2012.
- [5] Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, Ian Goodfellow, and Brendan Frey. Adversarial autoencoders. *arXiv preprint arXiv:1511.05644*, 2015.
- [6] Denisa Ghita, Hung Nguyen, Maciej Kurant, Katerina Argyraki, and Patrick Thiran. Netscope: Practical network loss tomography. In *2010 Proceedings IEEE INFOCOM*, Mar 2010.
- [7] H. X. Nguyen and P. Thiran. The boolean solution to the congested ip link location problem: Theory and practice. In *IEEE INFOCOM 2007 - 26th IEEE International Conference on Computer Communications*, Jan 2007.
- [8] R. Caceres, N.G. Duffield, S.B. Moon, and D. Towsley. Inference of internal loss rates in the mbone. In *Seamless Interconnection for Universal Services. Global Telecommunications Conference. GLOBECOM'99. (Cat. No.99CH37042)*, Jan 2003.
- [9] A. Chen, J. Cao, and T. Bu. Network tomography: Identifiability and fourier domain estimation. In *IEEE INFOCOM 2007 - 26th IEEE International Conference on Computer Communications*, Jan 2007.
- [10] Peter Sossalla, Justus Rischke, and Frank H. P. Fitzek. Enhanced one-way delay monitoring with openflow. In *2022 International Conference on Information Networking (ICOIN)*, pages 171–176, 2022.
- [11] Yan Qiao, Jun Jiao, Xinhong Cui, and Yuan Rao. Robust loss inference in the presence of noisy measurements and hidden fault diagnosis. *IEEE/ACM Trans. Netw.*, 28(1):43–56, feb 2020.
- [12] Jakub Kolar, Jan Sykora, Umberto Spagnolini, et al. Distributed network tomography applied to stochastic delay profile estimation. *Radioengineering*, 29(1):189–196, 2020.
- [13] Nick Duffield. Simple network performance tomography. In *Proceedings of the 2003 ACM SIGCOMM conference on Internet measurement - IMC '03*, Jan 2003.
- [14] Nagao Ogino, Takeshi Kitahara, Shin'ichi Arakawa, Go Hasegawa, and Masayuki Murata. Lightweight boolean network tomography based on partition of managed networks. *Journal of Network and Systems Management*, 26, 04 2018.
- [15] Novella Bartolini, Ting He, Viviana Arrigoni, Annalisa Massini, Federico Trombetti, and Hana Khamfroush. On fundamental bounds on failure identifiability by boolean network tomography. *IEEE/ACM Transactions on Networking*, 28(2):588–601, 2020.
- [16] Hung X. Nguyen and Patrick Thiran. *Binary Versus Analogue Path Monitoring in IP Networks*, page 97–107. Springer Berlin Heidelberg, Jan 2005.
- [17] Cho-Chun Chiu and Ting He. Stealthy dgos attack: Degrading of service under the watch of network tomography. In *IEEE INFOCOM 2020 - IEEE Conference on Computer Communications*, pages 367–376, 2020.
- [18] Cho-Chun Chiu and Ting He. Stealthy dgos attack against network tomography: The role of active measurements. *IEEE Transactions on Network Science and Engineering*, 8(2):1745–1758, 2021.
- [19] Yuntong Hu and Liang Zhao. Deepnt: Path-centric graph neural networks for network tomography. 2024.
- [20] Ippokratis Sartzetakis and Emmanouel Varvarigos. Machine learning network tomography with partial topology knowledge and dynamic routing. In *GLOBECOM 2022-2022 IEEE Global Communications Conference*, pages 4922–4927. IEEE, 2022.
- [21] Xu Tao and Simone Silvestri. Network tomography and reinforcement learning for efficient routing. In *2023 IEEE 20th International Conference on Mobile Ad Hoc and Smart Systems (MASS)*, pages 384–389. IEEE, 2023.
- [22] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.
- [23] Fazle Karim, Somshubra Majumdar, and Houshang Darabi. Insights into lstm fully convolutional networks for time series classification. *IEEE Access*, 7:67718–67725, 2019.
- [24] Ganji Tejasree and L Agilandeswari. Land use/land cover (lulc) classification using deep-lstm for hyperspectral images. *The Egyptian Journal of Remote Sensing and Space Sciences*, 27(1):52–68, 2024.
- [25] Sudhanshu Saurabh and PK Gupta. Deep learning-based modified bidirectional lstm network for classification of adhd disorder. *Arabian Journal for Science and Engineering*, 49(3):3009–3026, 2024.
- [26] Gang Dou, Kaixuan Zhao, MEI Guo, and JUN Mou. Memristor-based lstm network for text classification. *Fractals*, 31(06):2340040, 2023.
- [27] Mahati Munikoti Srikantamurthy, VP Subramanyam Rallabandi, Dawood Babu Dudekula, Sathishkumar Natarajan, and Junhyung Park. Classification of benign and malignant subtypes of breast cancer histopathology imaging using hybrid cnn-lstm based transfer learning. *BMC Medical Imaging*, 23(1):19, 2023.
- [28] Wenjian Sun, Linying Pan, Jingyu Xu, Weixiang Wan, and Yong Wang. Automatic driving lane change safety prediction model based on lstm. *arXiv preprint arXiv:2403.06993*, 2024.
- [29] Baozhu Li, Xiaotian Lv, and Jiaxin Chen. Demand and supply gap analysis of chinese new energy vehicle charging infrastructure: Based on cnn-lstm prediction model. *Renewable Energy*, 220:119618, 2024.
- [30] Parul Arora, Himanshu Kumar, and Bijaya Ketan Panigrahi. Prediction and analysis of covid-19 positive cases using deep learning models: A descriptive case study of india. *Chaos, Solitons ; Fractals*, page 110017, Oct 2020.
- [31] Farah Shahid, Aneela Zameer, and Muhammad Muneeb. Predictions for covid-19 with deep learning models of lstm, gru and bi-lstm. *Chaos, Solitons; Fractals*, page 110212, Nov 2020.
- [32] Vinay Kumar Reddy Chimmula and Lei Zhang. Time series forecasting of covid-19 transmission in canada using lstm networks. *Chaos, Solitons ; Fractals*, page 109864, Jun 2020.
- [33] Cornelius Fritz, Emilio Dorigatti, and David Rügamer. Combining graph neural networks and spatio-temporal disease models to predict covid-19 cases in germany. *arXiv: Learning,arXiv: Learning*, Jan 2021.
- [34] Jinyin Chen, Xuanheng Xu, Yangyang Wu, and Haibin Zheng. Gc-lstm: Graph convolution embedded lstm for dynamic link prediction. *arXiv: Social and Information Networks,arXiv: Social and Information Networks*, Dec 2018.
- [35] César Quilodrán Casas, Rossella Arcucci, Laetitia Mottet, Yike Guo, and ChristopherC. Pain. Adversarial autoencoders and adversarial lstm for improved forecasts of urban air pollution simulations. *Cornell University - arXiv,Cornell University - arXiv*, Apr 2021.
- [36] SukhpreetKaur Khangura and Sami Akin. Measurement-based online available bandwidth estimation employing reinforcement

- learning. *International Test Conference, International Test Conference*, Jan 2019.
- [37] Qiang Wang and Liang Cheng. Feat: Improving accuracy in end-to-end available bandwidth measurement. *Global Communications Conference, Global Communications Conference*, Jan 2006.
- [38] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, Pierre-Antoine Manzagol, and Léon Bottou. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of machine learning research*, 11(12), 2010.
- [39] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.
- [40] Pengzhi Li, Yan Pei, and Jianqiang Li. A comprehensive survey on design and application of autoencoder in deep learning. *Applied Soft Computing*, 138:110176, 2023.
- [41] Zhengcong Fei, Mingyuan Fan, Li Zhu, Junshi Huang, Xiaoming Wei, and Xiaolin Wei. Masked auto-encoders meet generative adversarial networks and beyond. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24449–24459, 2023.
- [42] Jingzhou Xin, Chaoying Zhou, Yan Jiang, Qizhi Tang, Xianyi Yang, and Jianting Zhou. A signal recovery method for bridge monitoring system using tvfemd and encoder-decoder aided lstm. *Measurement*, page 112797, 2023.
- [43] Dongyu Liu, Jinglin Zhou, and Huixia Jiang. Adversarial autoencoder concurrent projection to latent structure and its application. *The Canadian Journal of Chemical Engineering*, 102(1):274–290, 2024.
- [44] Simon Knight, Hung X. Nguyen, Nickolas Falkner, Rhys Bowden, and Matthew Roughan. The internet topology zoo. *IEEE Journal on Selected Areas in Communications*, page 1765–1775, Oct 2011.