EPJ Web of Conferences **337**, 01115 (2025) https://doi.org/10.1051/epjconf/202533701115
CHEP 2024

Learning to Predict Network Paths: A Transformer Model with Confidence-Based Imputation

Petya Vasileva[1,4], *Marian* Babik[2], *Shawn* McKee[1], and *Ilija* Vukotic[3]

[1]University of Michigan Physics, Ann Arbor, MI, USA
[2]European Organization for Nuclear Research (CERN), Geneva, Switzerland
[3]Physics Department, University of Chicago, Chicago, IL, USA
[4]Faculty of Mathematics and Informatics, University of Plovdiv, Bulgaria

Abstract. The research and education community relies on a robust network to access the vast amounts of data generated by scientific experiments. The underlying infrastructure connects hundreds of sites around the world, requiring reliable and efficient transfers of increasingly large datasets. These activities require proactive methods in network management, where potentially severe issues could be predicted and avoided before they can impact data exchanges. Our ongoing research focusses on leveraging deep learning (DL) methodologies, particularly Transformer-based model, to analyse network paths, and explore inter-connectivity across networks with the goal of predicting key performance metrics.

A key challenge in network topology modelling is handling missing or uncertain path segments. To address this, we incorporate confidence-aware learning in our Transformer model. This approach enables a more effective representation of network paths, leading to improved model performance.

In this work, we present our experimental findings, discuss the challenges associated with incomplete network paths, and compare the performance of our model with baseline models. Our results demonstrate the potential of transformer-based models in the refinement of network analysis and pave the way to more robust topology-based anomaly detection methods.

1 Introduction

Large-scale scientific collaborations are based on high-performance networks to transfer vast amounts of data across geographically distributed sites[1]. These networks form a critical backbone for research institutions, and ensuring their efficiency and reliability is crucial. Effective network monitoring can help detect potential disruptions and enhance overall performance. Thus, an international collaboration of experts developed and provided the perfSONAR framework [2] - a set of standardized tools to allow institutions, research centers, and universities to deploy uniform metrics for measuring network performance across varied infrastructures. The purpose of the platform is to facilitate network troubleshooting[3] by continuously measuring different aspects of the network.

Thousands of endpoints are being tested on regular scheduling intervals. Meanwhile, from a monitoring perspective, the volume of data is invaluable for diagnosing local and wide-area performance issues. However, it also raises significant complexity in terms of analyzing,

correlating, and interpreting results across thousands of endpoints. Subtle network issues, such as path-specific bandwidth degradation or loss patterns, can easily be overlooked.

A promising approach to make sense of the massive interwoven measurements is to develop topological models that represent the physical and logical links of the underlying network. However, such models require accurate routing information to map the actual path that packets take through the network. Our observations show more than two thirds of the path measurements (taken from the traceroute tests) may be missing segments or contain non-responsive hops. These gaps in traceroute measurements arise due to firewall policies, private IP address spaces, and transient network behaviors. This complicates efforts to build consistent and accurate topological models, as incomplete paths can cause misalignment or gaps in the network graph, which break the continuity in the network topology. Traditional approaches, such as heuristic-based imputation or simple statistical models, often fail to capture the complexity of network behaviors, particularly when dealing with long-range dependencies.

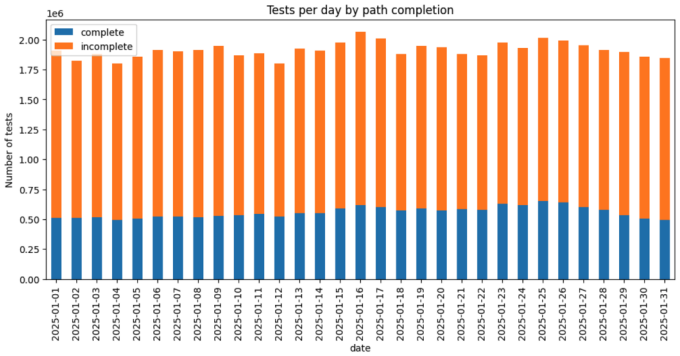

Figure 1: The plot shows the portion of complete/incomplete paths on a daily basis

Figure 1 shows that incomplete paths make up a large portion of our data set. Simply discarding these paths would substantially reduce the number of examples and bias our analysis, particularly since many long-range routes inevitably contain segments with missing information. Consequently, our aim is to retain the informative parts of these paths while minimizing noise, rather than removing them outright.

In this work, we propose a Transformer-based[4] approach to model network paths, incorporating confidence-aware learning to handle uncertainty. We introduce a beam search[5] strategy for node imputation, which allows us to infer missing hops by exploring multiple candidate sequences. Our model is evaluated against a baseline approach - a Trigram model[1], to assess its effectiveness in reconstructing paths.

Figure 2 shows the distribution of consecutive unknown hop segments in our dataset. Although some long routes exhibit large gaps or only a handful of known hops, the majority of incomplete segments consist of only one or two unknown nodes. Concretely, over 60% of these segments have a single unknown hop, and nearly 20% have only two. This suggests that, in practice, most missing segments are relatively short and should be easier to reconstruct with methods like beam search or trigram-based imputation. In contrast, the small fraction of paths with longer consecutive unknowns (3 or more hops) indicates the more challenging corner cases, typically arising in long-haul routes or heavily firewalled network segments.

[1] A trigram is a group of three consecutive characters in a string

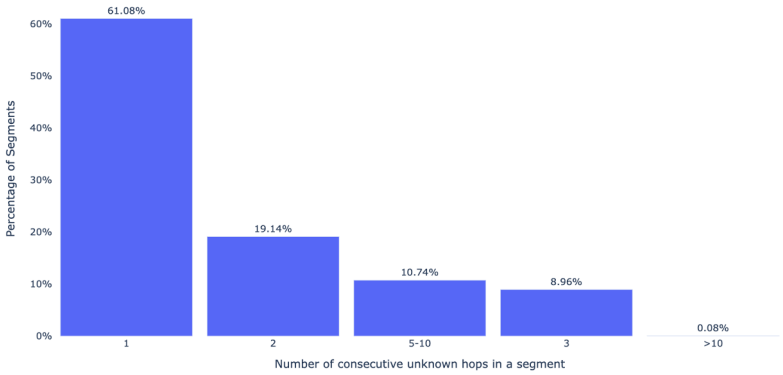

Figure 2: Distribution of missing hop segments as percentage of all paths

Through our experiments, we demonstrate that our method improves path prediction and imputation, providing a more accurate representation of network connectivity. These findings contribute to ongoing efforts in network anomaly detection and predictive network monitoring.

2 Related Work

Traceroute-based measurements have long been a cornerstone of network diagnostics, providing insight into routing paths and performance anomalies. However, modern network environments often produce incomplete traceroute data due to firewall policies, network address translation (NAT), and transient routing behaviors. Traditional heuristic and statistical methods have been used to address missing segments in network traceroute measurements. Spring et al. [6] introduced a heuristic-based approach to mapping internet service provider (ISP) topologies using traceroute data, which includes methods to interpolate missing hops. Similarly, Duffield et al [7] have used statistical network tomography techniques to infer unobserved network characteristics from partial measurements. However, these methods often struggle to capture the long-range dependencies inherent in large-scale networks. In contrast, recent work has taken advantage of learning-based approaches to improve path reconstruction accuracy. For example, Huffaker et al. [8] present an in-depth comparison of Internet topology data that underscores the diversity and complexity of real-world network graphs, a challenge that motivates our Transformer-based approach.

To better understand real-world network topologies, data sets like The Internet Topology Zoo [9] have been compiled, providing a curated collection of diverse network graphs. These data sets reveal the variability in network structures in different domains, which poses a major challenge for accurate topology reconstruction. The incomplete nature of traceroute data makes it difficult to align observed paths with known topologies, underscoring the need for probabilistic or learning-based methods to infer missing segments. Our work builds on these efforts by using a Transformer-based model to reconstruct missing paths while preserving topological consistency.

3 Methodology

3.1 Data Preparation

We use traceroute-based measurements provided by the perfSONAR toolkits, capturing sequences of network hops with corresponding time to live (TTL) values. For the purpose of the current experiment, we work on a subset of the dataset of about 3-4 days of data. The dataset consists of millions of paths, with a significant fraction containing missing or anonymized hops. To ensure consistency, we preprocess the data by aligning TTL values, identifying private IPs, and marking unknown hops.

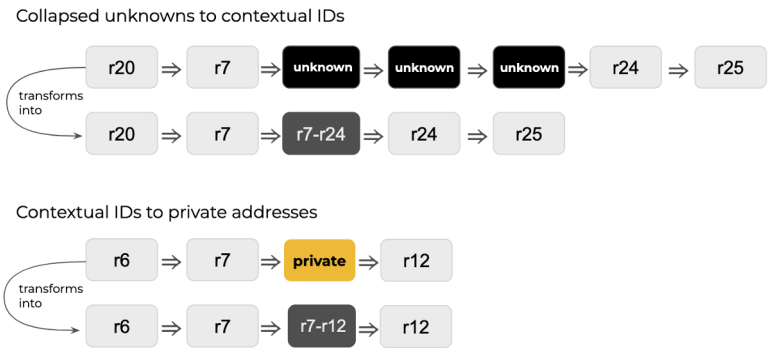

Figure 3: We propose path transformations where one or many consecutive unknown hops become one merged token, and private nodes become a single contextual placeholder.

As described in the figure 3, in case of missing hops or internal IP addresses, we define a placeholder node by taking the previous and next measured known nodes. This approach reduces the noise in our dataset by discarding the meaningless (repeated) unknown hops and simplifies the sequences of hops which will be provided for training the Transformer model. Moreover, considering that private IP addresses are not globally unique, we anchor each internal address to the local portion of the path where it appears. This prevents it from being treated as the same router across the entire global network.

3.2 Transformer Model for Path Prediction

Our model is based on a Transformer architecture, which enables efficient sequence modeling with self-attention mechanisms. Each hop in a network path is treated as a token, and its position in the sequence is encoded along with additional confidence scores. The model is trained to predict the next hop in a sequence, even in the presence of missing segments.

3.3 Confidence-Aware Learning

To account for uncertainty in network paths, we adopt a confidence-aware approach in which every hop in the sequence receives a numerical confidence score. Known public IPs and identifiable private addresses are assigned full confidence (that is, a score of 1.0), while partially anonymised or unknown hops receive lower confidence values. This distinction helps the model focus on reliable nodes while still preserving information about segments that pass

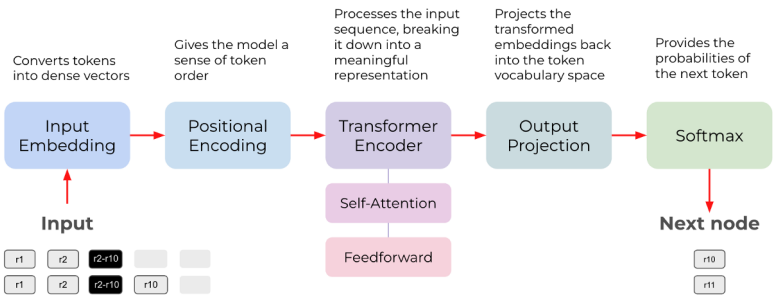

Figure 4: Transformer model architecture

through local networks or firewall-protected devices. As illustrated in Figure 5, certain positions (e.g., hops 10–12) consistently appear in the path over multiple days—indicating real nodes, but remain unresolvable at the IP level. Rather than discarding these uncertain tokens, our framework lowers their relative weight during training yet continues to "pay attention" to them during inference, ensuring the model captures the stable presence of these unknown hops. This balanced treatment prevents overfitting to unreliable data while retaining valuable structural clues about the path.

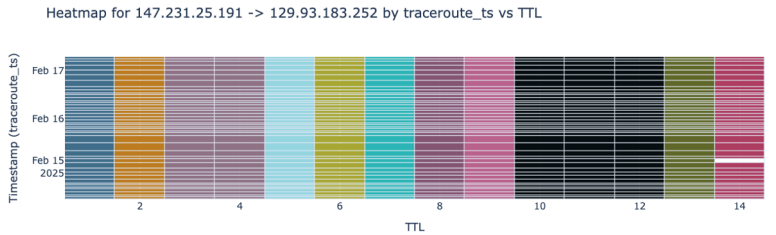

Figure 5: Heatmap of traceroutes from 147.231.25.191 to 129.93.183.252 over multiple days, with each hop color-coded by IP address and unknown hops shown in black. The x-axis represents the TTL (hop position), while the y-axis shows successive traceroute measurements by timestamp. Notably, hops at positions 10–12 remain consistently unknown across all measurements, indicating stable but unresolvable local or firewall-protected segments.

3.4 Confidence-Weighted Loss

To handle missing or uncertain segments more effectively, we introduce a *confidence-weighted* loss function. Each hop in a sequence is assigned a confidence score α_i, where $\alpha_i = 1$ for reliably known nodes (including private IP addresses with context) and $0 < \alpha_i < 1$ for uncertain or anonymized hops. The Transformer is then trained to minimize a modified cross-entropy loss:

$$\mathcal{L}_{\text{weighted}} = -\sum_{i=1}^{N} \alpha_i \cdot \log\Big(P(\text{hop}_i \mid \text{context}_i)\Big),$$

where $P(\text{hop}_i \mid \text{context}_i)$ is the predicted probability of the correct insertion at position i, and N is the length of the sequence. By multiplying the logarithmic probability term by α_i, we reduce the impact of partially or uncertain tokens during training. This encourages the model to focus more on hops where we have higher certainty, while still allowing it to learn from partially known segments.

3.5 Beam Search for Node Imputation

To reconstruct missing segments, we employ a beam search strategy [10], generating multiple candidate sequences, and selecting the most probable path. Given a known start and end node in a path, the model iteratively predicts intermediate hops while maintaining a limited set of best candidates. This approach balances efficiency with accuracy, ensuring that the predicted sequence aligns with the observed network patterns.

Currently, we are in the process of implementing the optimal beam search method, which can work on true incomplete paths. Further analysis of the results will help us decide whether to use both models in a hybrid implementation or prefer one over the other.

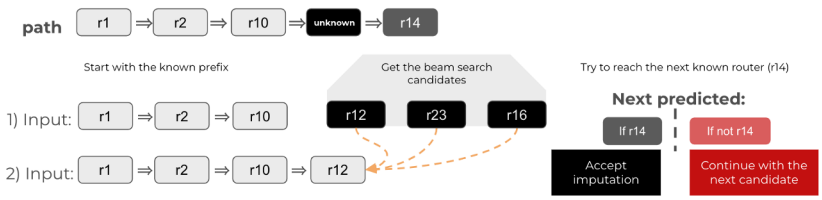

Figure 6: Illustration of beam search–based node imputation. We begin with a known prefix (r1 → r2 → r10) and aim to reach the next known router (r14). The model proposes multiple candidate hops (r12, r23, r16), each representing a possible missing node. If a candidate successfully matches the known anchor (r14), we accept the imputation; otherwise, we continue exploring other beam candidates or subsequent steps.

4 Experiments and Results

We evaluated our approach using a dataset of real-world network paths, partitioned into training, validation, and test sets. The Transformer model is trained using standard sequence prediction techniques, with cross-entropy loss and confidence-based weighting. For comparison, we implement a baseline model - trigram-based predictions.

4.1 Comparison of Transformer and Trigram Baseline

After training both our Transformer model (with confidence-weighted loss) and a simpler Trigram baseline on the same dataset, we evaluated them on the test set, measuring average loss and accuracy for the final prediction step in each sequence. Figure 7 presents the results:

From the left plot in Figure 7, we observe that the Transformer achieves a somewhat lower average loss than the Trigram model. While this gap is not dramatic, it may reflect the effect of confidence-weighted training, which allows the Transformer to down-weight uncertain tokens. On the right, the accuracy plot shows that both methods perform similarly, with the Transformer having a marginal edge on more challenging segments.

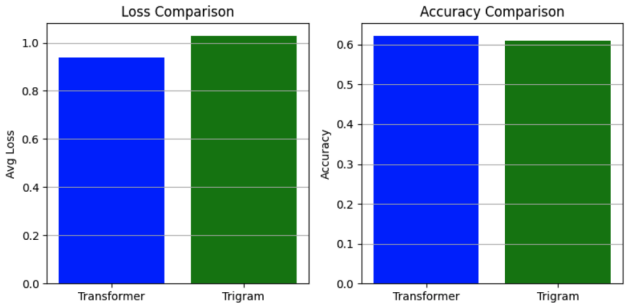

Figure 7: Comparison of average loss (left) and accuracy (right) for the Transformer-based model vs. the Trigram baseline. Both methods achieve similar performance, though the Transformer shows a slightly lower loss, likely aided by the confidence-weighted training.

5 Conclusion and Next Steps

In this work, we presented an initial attempt to impute missing segments in network paths using a Transformer-based model with confidence-aware learning, alongside a complementary Trigram baseline. While our approach demonstrates the potential of deep learning for reconstructing network paths, particularly through the use of dynamic beam search and contextual placeholders, our current implementation achieves only about 60% accuracy on incomplete segments. This performance suggests that there is significant room for improvement. Future work will focus on fine-tuning the Transformer model, optimizing hyperparameters, and exploring hybrid approaches — such as using the Trigram model for shorter gaps and the Transformer for longer ones. The reason for considering this is that preliminary checks demonstrate more plausible predictions done by the Transformer model on longer paths compared to Trigram-based imputations on shorter paths.

In addition, we intend to incorporate the time dimension into our analysis. Network paths evolve over time due to changes in routing policies, maintenance activities, and varying network conditions. By integrating temporal information—using techniques such as time embeddings or recurrent layers—into our models, we expect to capture these dynamic changes more accurately. This enhancement would allow the system not only to reconstruct missing segments based on static contextual cues but also to adapt to temporal variations, ultimately enabling more proactive network management and anomaly detection.

Together, these refinements promise a more robust and adaptive framework for network path reconstruction.

6 Acknowledgements

We acknowledge our collaborations with the CERN IT, WLCG and LHCONE/LHCOPN communities, who participated in this effort. We want to explicitly acknowledge the support of the National Science Foundation which provided long-term support for this work via:

OSG: NSF MPS-1148698, NSF SAND Project: NSF OAC-1827116, IRIS-HEP: NSF OAC-1836650.

References

[1] M. Babik, S. McKee, P. Andrade, B.P. Bockelman, R.W. Gardner, E.M.F. Hernandez, E. Martelli, I. Vukotic, D. Weitzel, M. Zvada, CoRR **abs/2007.00598** (2020), 2007.00598

[2] A. Hanemann, J.W. Boote, E.L. Boyd, J. Durand, L. Kudarimoti, R. Łapacz, D.M. Swany, S. Trocha, J. Zurawski, *PerfSONAR: A Service Oriented Architecture for Multi-domain Network Monitoring*, in *Service-Oriented Computing - ICSOC 2005*, edited by B. Benatallah, F. Casati, P. Traverso (Springer Berlin Heidelberg, Berlin, Heidelberg, 2005), pp. 241–254, ISBN 978-3-540-32294-8

[3] Vasileva, Petya, Babik, Marian, McKee, Shawn, Vukotic, Ilija, EPJ Web of Conf. **295**, 07003 (2024)

[4] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, L. Kaiser, I. Polosukhin, CoRR **abs/1706.03762** (2017), 1706.03762

[5] S. Wiseman, A.M. Rush, CoRR **abs/1606.02960** (2016), 1606.02960

[6] N. Spring, R. Mahajan, D. Wetherall, IEEE/ACM Transactions on Networking **12**, 2 (2004)

[7] N. Duffield, *Network Tomography*, in *Proceedings of the 2004 ACM SIGCOMM Workshop on Internet Measurement* (ACM, 2004), pp. 1–8

[8] B. Huffaker, M. Fomenko, K. Claffy, *Internet topology data comparison*, Technical Report, CAIDA (2024), accessed: 2025

[9] L. Doyle et al., *The internet topology zoo*, http://www.topology-zoo.org (2011), accessed: 2025-02-26

[10] S. Wiseman, A.M. Rush, CoRR **abs/1606.02960** (2016), 1606.02960