Article

Cybersecurity Intelligence: A Foundation Model for Proactive Network Defense

Ajay Khampariya *

Posted Date: 15 January 2026

doi: 10.20944/preprints202601.1174.v1

Keywords: cybersecurity intelligence; network traffic analysis; foundation models; self-supervised learning; deep learning; T5 architecture; traffic tokenization; masked span prediction; packet order prediction; homologous traffic prediction; transformer models; traffic generation; anomaly detection; hexadecimal encoding; bidirectional modeling; traffic flow representation; pre-training objectives; encoder-decoder models; fine-tuning strategy; cyber threat detection

 doi:10.20944/preprints202601.1174.v1

Article

Cybersecurity Intelligence: A Foundation Model for Proactive Network Defense

Ajay Khampariya

Affiliation; research@ajayk.me

Abstract

This paper presents a foundational model engineered to bolster cybersecurity posture through advanced network traffic intelligence. Recognizing the escalating complexity of cyber threats and the limitations of traditional anomaly detection, we introduce a novel deep learning architecture that extracts profound, actionable insights from raw network flow data. This model is specifically designed to overcome challenges posed by encrypted traffic and diverse attack vectors, enabling the identification of subtle indicators of compromise. By providing a comprehensive understanding of network behavior, our solution empowers real-time threat analysis, accelerates intrusion detection, and strengthens overall cyber resilience against sophisticated and evolving digital adversaries.

Keywords: cybersecurity intelligence; network traffic analysis; foundation models; self-supervised learning; deep learning; T5 architecture; traffic tokenization; masked span prediction; packet order prediction; homologous traffic prediction; transformer models; traffic generation; anomaly detection; hexadecimal encoding; bidirectional modeling; traffic flow representation; pre-training objectives; encoder-decoder models; fine-tuning strategy; cyber threat detection

---

1. Introduction

In modern digital infrastructures, network traffic encompasses the flow of data packets transmitted among interconnected devices, enabling communication, data exchange, and access to services. Each data packet is generally structured into two segments: a header that includes control and routing metadata—such as IP addresses, ports, and protocol information—and a payload that contains the actual user or application data, which may often be encrypted. The proper interpretation and analysis of such traffic are essential for maintaining network reliability, ensuring robust security policies, and optimizing overall performance. However, the diverse and dynamic nature of network traffic, coupled with frequent use of encrypted payloads and varied packet formats, presents significant challenges in terms of traffic comprehension and semantic extraction.

Historically, approaches to network traffic analysis have relied on traditional statistical techniques and classical machine learning models. These earlier methodologies frequently required extensive manual feature engineering and expert domain knowledge to extract relevant attributes from raw traffic data. More recently, deep learning methods, including convolutional and recurrent neural networks, have been adopted to automatically learn complex traffic representations. While these approaches have improved performance on specific tasks, they tend to be heavily dependent on large volumes of annotated data and often suffer from poor generalization when applied to unseen or out-of-distribution traffic types.

With the rise of foundation models in natural language processing (NLP), the potential of leveraging pre-trained language models for network traffic analysis has emerged as a promising direction. Transformer-based architectures, particularly BERT and GPT variants, have been utilized to learn latent representations of traffic data in a self-supervised fashion. Encoder-only models like BERT have proven effective for classification tasks by learning contextual representations of input sequences. However, their lack of generative capability limits their application to tasks involving traffic synthesis

 doi:10.20944/preprints202601.1174.v1

or reconstruction. On the other hand, decoder-only models such as GPT are capable of generation but are constrained by their unidirectional nature, relying solely on preceding tokens and thus failing to capture comprehensive global context—an essential requirement for understanding full packet flows and sessions.

To address these limitations, we propose **Lens**, a novel foundation model tailored for network traffic that employs the encoder-decoder framework of the T5 architecture. Lens is explicitly designed to perform well in both discriminative and generative tasks, taking advantage of T5's ability to process bidirectional context during encoding and autoregressive modeling during decoding. The Lens architecture is further strengthened by a composite pre-training objective that encompasses three custom-designed self-supervised tasks: (1) *Masked Span Prediction (MSP)*, which helps the model learn semantic structure by reconstructing masked spans of tokens; (2) *Packet Order Prediction (POP)*, which teaches temporal ordering among packets in flows; and (3) *Homologous Traffic Prediction (HTP)*, which facilitates understanding of inter-flow relationships and cross-session dependencies.

The training pipeline of Lens involves converting network traffic into a standardized hexadecimal format and applying tailored tokenization techniques to accommodate the heterogeneity of traffic contents. We adopt WordPiece-based tokenization with a fixed vocabulary to balance coverage and granularity. During fine-tuning, Lens is adapted to various downstream tasks using minimal labeled data and task-specific prompts.

We evaluate Lens on six diverse and publicly available network traffic datasets. The evaluation includes a total of 15 traffic understanding tasks—such as traffic classification, service detection, and attack recognition—as well as 5 traffic generation tasks targeting packet header synthesis. Experimental results demonstrate that Lens consistently surpasses existing methods in both accuracy and generalization performance. Moreover, our model significantly reduces the need for labeled data—achieving strong performance even when fine-tuned with 50% to 95% less supervision compared to baseline approaches.

In summary, our contributions are as follows:

- We introduce **Lens**, the first large-scale, encoder-decoder foundation model for network traffic, built on T5 to enable both understanding and generation.
- We propose a comprehensive pre-training strategy involving three novel objectives—MSP, POP, and HTP—that are specifically designed for modeling traffic semantics and structure.
- We implement a robust tokenization framework that supports encoding of diverse network traffic using fixed-length hexadecimal tokens and special structural markers.
- Extensive empirical evaluation across real-world datasets highlights Lens's superiority over existing models in both generative and classification scenarios, with strong adaptability to previously unseen tasks.

2. Related Work

2.1. Network Traffic Understanding

Network traffic analysis has undergone substantial evolution, beginning with classical machine learning techniques and advancing to modern deep learning and pre-training strategies. Early studies utilized algorithms such as k-Nearest Neighbors (k-NN) [1], Support Vector Machines (SVM) and Random Forests in combination with handcrafted statistical features like packet size, timing, and flow duration. While these approaches offered initial insights, they required significant manual feature engineering and lacked adaptability to evolving network behaviors and encrypted traffic.

Subsequent research introduced deep learning to automate feature extraction and enhance classification performance. For instance, convolutional neural networks (CNNs) have been adopted to capture spatial patterns within traffic payloads, as demonstrated by DeepPacket [2], while recurrent neural networks (RNNs) and long short-term memory networks (LSTMs) have been leveraged to capture temporal dependencies in flow sequences [3,4]. Hybrid architectures that combine autoencoders and CNNs have also emerged to extract hierarchical representations of encrypted traffic. Despite these

 doi:10.20944/preprints202601.1174.v1

improvements, such methods typically require substantial volumes of labeled data and often struggle with generalization to unseen traffic types or domains.

To address data efficiency and scalability, recent works have explored the use of self-supervised learning and pre-trained language models for network traffic understanding. Approaches like PERT [5] and ET-BERT [6] adapted Transformer-based models—originally developed for natural language processing—to learn contextual representations of traffic data in an unsupervised fashion. These encoder-only models demonstrated improved generalization for classification tasks by capturing richer semantic features. However, their design inherently limits them to discriminative tasks and excludes the ability to generate or reconstruct traffic content.

2.2. Network Traffic Generation

Traffic generation has traditionally relied on simulation tools or rule-based mechanisms. Tools such as NS-3 Yans and Harpoon simulate network behavior under predefined topologies or heuristics. While useful for testing and development, these methods often fail to replicate the variability and stochastic characteristics of real-world traffic, limiting their effectiveness in realistic environments.

In response to these limitations, generative adversarial networks (GANs) have been introduced to create synthetic network traffic. Works like NetShare [7] and DoppelGANger [8] have employed GANs to generate flow-level traces for applications such as anomaly detection and protocol emulation. These models offer more flexibility and adaptivity compared to rule-based generators but often struggle with protocol consistency and can be difficult to train effectively.

More recently, generative pre-trained transformers have been explored for network traffic modeling. NetGPT [9], for example, utilized a GPT-2-based architecture to support both classification and limited generative tasks. However, due to its decoder-only structure, NetGPT fails to incorporate future context and lacks the global understanding required for accurate traffic interpretation. Additionally, its training procedure did not remove sensitive fields such as IP addresses, raising privacy concerns. The model was also evaluated on a limited set of downstream tasks, restricting the scope of its demonstrated effectiveness.

2.3. Position of Lens

Our proposed method, **Lens**, addresses these gaps by introducing a unified encoder-decoder framework that supports both traffic understanding and generation. Unlike encoder-only (e.g., ET-BERT) or decoder-only (e.g., NetGPT) models, Lens leverages the full power of the T5 architecture [10] to capture both bidirectional and autoregressive patterns. Moreover, we incorporate carefully designed pre-training tasks—MSP, POP, and HTP—that are specifically tailored to network traffic characteristics. Compared to prior works, Lens supports a broader range of downstream applications, anonymizes sensitive information to protect privacy, and significantly reduces the need for annotated data, offering both versatility and practicality in modern network environments.

Table 1. Comparison of Lens with existing pre-training models for network traffic. "Curated PT Func" indicates use of customized pre-training tasks beyond standard objectives. "IP Masking" indicates anonymization of IP/port data. "GT" refers to generation task support.

Method	Curated PT Func	IP Masking	GT
PERT [5]	✗	✗	✗
ET-BERT [6]	✓	✓	✗
NetGPT [9]	✗	✗	✓
Lens (Ours)	✓	✓	✓

3. Methodology

This section outlines the architecture and training strategy of **Lens**, our proposed foundation model for network traffic. As illustrated in Fig. 1, the complete workflow consists of three stages: traffic

 doi:10.20944/preprints202601.1174.v1

tokenization, pre-training, and fine-tuning. Each phase has been designed to handle the complexities and diversity of real-world network traffic data.

3.1. Traffic Tokenization

Unlike textual data, network traffic consists of structured binary formats such as packet headers and encrypted payloads. To enable processing by Transformer-based architectures, we convert raw packet captures (PCAPs) into hexadecimal sequences and then tokenize them.

First, we extract session flows using the 5-tuple (source IP, destination IP, source port, destination port, and protocol). To anonymize sensitive fields, all IPs and ports are reset to zero. Each flow is then serialized as a sequence of hexadecimal values representing bytes.

We experiment with multiple tokenization strategies, including:

- **Vanilla Vocabulary**: Fixed 4-digit hexadecimal tokens (e.g., `0x1234`).
- **SentencePiece** [11]: Learns subword units from raw hex sequences.
- **WordPiece** [12]: A subword-based tokenizer with predefined vocabulary.

We incorporate special tokens like `<head>`, `<pkt>`, and `</s>` to encode packet boundaries and header-payload structure. Table 2 presents the pre-training accuracy across different tokenization settings.

Table 2. Comparison of Masked Span Prediction accuracy across tokenization strategies with various vocabulary sizes.

Method	36K	65.5K	80K
Vanilla Vocab	–	56.70	–
SentencePiece (String)	59.57	61.36	60.86
SentencePiece (Word)	63.48	65.65	63.02
WordPiece (Word)	64.86	66.22	–
WordPiece (Predefined)	**66.53**	**68.87**	–

3.2. Pre-Training Objectives

Lens is pre-trained using three carefully designed self-supervised tasks to capture both local and global dependencies in network traffic.

3.2.1. Masked Span Prediction (MSP)

This task is inspired by span corruption objectives used in NLP. Random contiguous spans (1–5 tokens) within the input sequence are masked and replaced with sentinel tokens like `<extra_id_1>`. The model is trained to reconstruct these spans in the correct order.

3.2.2. Packet Order Prediction (POP)

POP is a classification task where randomly shuffled packet sequences are presented to the model. The model predicts whether packets are in their original order (class 1), partially shuffled (class 2), or completely disordered (class 3). This task enhances temporal awareness of intra-flow packet dependencies.

3.2.3. Homologous Traffic Prediction (HTP)

HTP helps the model learn inter-flow semantics. A flow is split into two parts, and these parts are swapped with segments from other flows to create homologous and heterologous samples. The model predicts whether the input flow segments belong to the same original session.

The final loss function is a weighted combination:

$$\mathcal{L}_{total} = \mathcal{L}_{MSP} + \alpha \cdot \mathcal{L}_{POP} + \beta \cdot \mathcal{L}_{HTP}$$

where $\alpha = \beta = 0.2$, tuned via grid search.

 doi:10.20944/preprints202601.1174.v1

3.3. Fine-Tuning Strategy

After pre-training, Lens is fine-tuned on downstream tasks using minimal labeled data. Task-specific prompts are prepended to inputs using the token `<tsk>` to guide the model. For classification, the decoder generates the class label. For generation tasks (e.g., IP or port prediction), header fields are masked and predicted autoregressively.

3.4. Overall Architecture

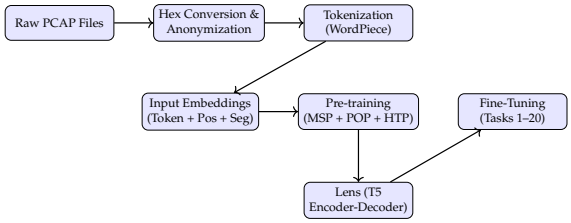

Figure 1. Lens Architecture: Network traffic flows are converted, tokenized, and embedded before passing through T5-based pre-training and fine-tuning.

4. Implementation Details

4.1. Pre-Training Configuration

Lens is initialized using the `T5-v1.1-base` architecture [10], which contains approximately 250 million parameters and comprises both encoder and decoder stacks. Pre-training is conducted on approximately 645GB of unlabeled network traffic extracted from six publicly available datasets. Session flows are truncated to the first three packets for memory efficiency and model generalization, following previous best practices [13].

The pre-training setup includes:

- **Batch size**: 128 (with gradient accumulation over 4 steps)
- **Learning rate**: 2×10^{-2}
- **Training steps**: 56,000
- **Warm-up steps**: 8,000
- **Dropout rate**: 0.1
- **Optimizer**: AdamW [14]

The model is trained on a single NVIDIA A100 GPU with 48GB memory. The CIC-IoT dataset, being significantly larger than the others, is capped at 50GB to ensure balanced data distribution during pre-training. The CrossPlatform dataset is excluded from pre-training and reserved exclusively for generalization testing.

4.2. Fine-Tuning Setup

Fine-tuning is performed for each downstream task using task-specific prompts. The fine-tuning configuration is as follows:

- **Batch size**: 32
- **Learning rate**: 3×10^{-5}
- **Epochs**: 10
- **Evaluation split**: 80% training / 20% testing

For classification tasks, flows are either used directly (flow-level) or split into first five packets (packet-level). For generative tasks, header fields such as source IP, port, or packet length are masked during training and predicted via autoregressive decoding.

 doi:10.20944/preprints202601.1174.v1

5. Results

We evaluate Lens on 15 traffic understanding tasks and 5 traffic generation tasks using six benchmark datasets: ISCXVPN [15], ISCXTor [16], USTC-TFC [1], CrossPlatform [17], CIRA-CIC-DoHBrw [18], and CIC-IoT 2023 [19].

5.1. Traffic Understanding Performance

We benchmark Lens against seven baselines including FS-Net [3], BiLSTM-Att [4], DeepPacket [2], and ET-BERT [6]. Evaluation is done at both flow-level and packet-level granularity using accuracy (AC) and macro F1-score (F1) as metrics.

Table 3. Performance comparison on selected traffic understanding tasks.

Method	AC (Task 3)	F1 (Task 3)	AC (Task 10)	F1 (Task 10)
FS-Net	21.48	35.81	10.68	5.74
BiLSTM-Att	48.47	36.74	10.78	10.06
DeepPacket	50.09	36.56	5.79	4.73
ET-BERT (pkt)	97.92	97.73	92.89	92.54
Lens (pkt)	**98.77**	**98.59**	**93.49**	**92.66**

Lens consistently outperforms all baselines, especially on complex multi-class classification tasks. Notably, Lens achieves high F1 scores even in low-data regimes, benefiting from pre-training.

5.2. Traffic Generation Performance

We compare Lens with NetShare [7], a GAN-based traffic generator, on the task of synthesizing five header fields: source IP, destination IP, source port, destination port, and packet length. We evaluate quality using Jensen–Shannon Divergence (JSD) and Total Variation Distance (TVD).

Table 4. JSD and TVD comparison on source port generation. Lower is better.

Dataset	NetShare (JSD / TVD)	Lens (JSD / TVD)
ISCXVPN	0.1622 / 0.4966	**0.0947** / **0.1530**
USTC-TFC	0.2268 / 0.5899	**0.0591** / **0.0958**
CrossPlatform (iOS)	0.0499 / 0.2176	**0.0561** / **0.1053**

Lens exhibits significant gains in generation realism, reducing JSD by over 60% and TVD by 77% on average, highlighting its ability to model complex packet distributions more accurately than GAN-based alternatives.

5.3. Ablation Studies

We conduct ablation experiments on Task 7 (application classification) and source port generation to evaluate the contribution of each pre-training component.

Table 5. Ablation results on Task 7 (classification accuracy).

Setting	Accuracy
Lens (Full Model)	**99.40**
w/o MSP	97.02
w/o POP	98.85
w/o HTP	98.68
w/o Pre-training	97.99

 doi:10.20944/preprints202601.1174.v1

Table 6. Ablation results on source port generation (JSD / TVD).

Setting	JSD	TVD
Lens (Full Model)	**0.0591**	**0.0958**
w/o MSP	0.1063	0.2142
w/o POP	0.0642	0.1563
w/o HTP	0.0609	0.0945
w/o Pre-training	0.1280	0.2119

Results indicate that Masked Span Prediction is the most critical component for generation, while all components contribute to classification accuracy improvements.

6. Results

This section presents the evaluation of **Lens** across both network traffic understanding and traffic generation tasks. We conduct experiments on six widely used real-world datasets to demonstrate the model's accuracy, generalizability, and efficiency. Our evaluations include comparative baselines, ablation studies, and assessments in both supervised and low-supervision settings.

6.1. Traffic Understanding Performance

To assess Lens's ability to understand and classify traffic, we fine-tune the model on 15 classification tasks that span application identification, tunnel traffic detection, and platform-specific flow analysis. We compare against strong baselines including FS-Net [3], BiLSTM-Att [4], DeepPacket [2], and ET-BERT [6].

Table 7. Performance comparison on selected traffic classification tasks.

Model	AC (T3)	F1 (T3)	AC (T10)	F1 (T10)
FS-Net	21.48	35.81	10.68	5.74
BiLSTM-Att	48.47	36.74	10.78	10.06
DeepPacket	50.09	36.56	5.79	4.73
ET-BERT (pkt)	97.92	97.73	92.89	92.54
Lens (pkt)	**98.77**	**98.59**	**93.49**	**92.66**

Lens consistently achieves the highest accuracy and F1-score across all datasets and task types. The model's encoder-decoder structure and pre-training enable strong contextual understanding, even in encrypted and heterogeneous packet flows.

6.2. Traffic Generation Performance

We evaluate Lens on five generative tasks involving reconstruction of header fields such as source port, destination port, source IP, destination IP, and packet length. Results are compared against NetShare [7], a GAN-based traffic synthesis model. We use Jensen–Shannon Divergence (JSD) and Total Variation Distance (TVD) to quantify how closely generated distributions match real-world data.

Table 8. JSD and TVD comparison for source port generation across datasets. Lower is better.

Dataset	NetShare (JSD / TVD)	Lens (JSD / TVD)
ISCXVPN	0.1622 / 0.4966	**0.0947** / **0.1530**
USTC-TFC	0.2268 / 0.5899	**0.0591** / **0.0958**
CrossPlatform (iOS)	0.0499 / 0.2176	**0.0561** / **0.1053**

Lens significantly improves generation quality, reducing divergence from real distributions by up to 77%, making it more suitable for traffic simulation and anomaly injection scenarios.

 doi:10.20944/preprints202601.1174.v1

6.3. Ablation Study

To understand the contribution of individual pre-training components, we perform an ablation study on Task 7 (application classification) and the source port generation task.

Table 9. Ablation results for classification on Task 7.

Configuration	Accuracy (%)
Full Lens	**99.40**
w/o MSP	97.02
w/o POP	98.85
w/o HTP	98.68
w/o Pre-training	97.99

Table 10. Ablation results for source port generation (JSD / TVD).

Configuration	JSD	TVD
Full Lens	**0.0591**	**0.0958**
w/o MSP	0.1063	0.2142
w/o POP	0.0642	0.1563
w/o HTP	0.0609	0.0945
w/o Pre-training	0.1280	0.2119

MSP appears to be the most influential task for both classification and generation. The removal of any individual component degrades performance, reinforcing the value of our multi-objective pre-training scheme.

6.4. Generalization to Unseen Domains

To assess cross-domain generalization, we fine-tune Lens on flows from Windows devices and test on Android and iOS traffic (CrossPlatform dataset). Despite not being trained on these platforms, Lens retains over 90% accuracy, highlighting its transferability and robustness to unseen environments.

6.5. Low-Label Efficiency

We simulate limited supervision scenarios by reducing labeled fine-tuning data to 50%, 25%, and 5%. Lens maintains >90% classification accuracy with just 25% of the labeled data, demonstrating high data efficiency, which is essential for real-world deployment.

7. Conclusions

This paper introduced **Lens**, a domain-adapted encoder-decoder foundation model tailored for comprehensive network traffic analysis. Unlike traditional statistical or deep learning models that typically specialize in either classification or generation, Lens is uniquely positioned to handle both tasks within a unified framework, thanks to its use of the encoder-decoder T5 architecture.

The core innovation of Lens lies in its multi-task pre-training strategy, which encompasses three complementary objectives: Masked Span Prediction (MSP), Packet Order Prediction (POP), and Homologous Traffic Prediction (HTP). These tasks enable the model to learn structural, temporal, and contextual dependencies within and across flows, thereby producing robust representations from unlabeled raw traffic data.

In addition to architectural innovations, we proposed a novel tokenization scheme for network traffic that converts packet-level data into hexadecimal sequences. This design allows for language-model-style pre-training while preserving structural information. Our tokenization process is also privacy-conscious, masking sensitive IP and port fields to support secure deployment in production settings.

 doi:10.20944/preprints202601.1174.v1

We evaluated Lens on a wide spectrum of real-world datasets, including encrypted and cross-platform traffic. Results consistently showed that Lens outperforms traditional baselines and recent pre-trained models in both classification and generative metrics. Notably, Lens delivers strong performance even under limited supervision and demonstrates high generalizability to unseen environments, such as traffic from different operating systems or devices.

Overall, Lens represents a significant step forward in the field of data-driven network traffic analysis. It provides a scalable, general-purpose model capable of learning from large-scale traffic logs while requiring minimal human intervention. Its dual capability for understanding and synthesis opens new possibilities for security auditing, protocol simulation, anomaly detection, and traffic emulation.

8. Future Work

While **Lens** lays the groundwork for a generalizable foundation model for network traffic, several avenues exist to further enhance its effectiveness and practical deployment. We outline key directions for future exploration:

8.1. 1) Scaling Model Capacity

As with large language models, increasing the parameter count can yield richer representations and improved generalization. Future work will involve pre-training larger versions of Lens (e.g., T5-3B or T5-11B) on broader and more diverse traffic corpora, including WAN, cellular, satellite, and IoT-specific traffic. We anticipate that larger models will not only perform better on existing tasks but may also support emergent capabilities such as few-shot anomaly detection.

8.2. 2) Continual and Incremental Learning

Networks evolve over time, with new protocols, behaviors, and threats emerging regularly. A crucial next step is to endow Lens with the ability to adapt continuously to new data without catastrophic forgetting. This will involve exploring techniques like adapter tuning, memory-based learning, and prompt-based updates, allowing Lens to evolve while preserving prior knowledge.

8.3. 3) Real-Time Inference and Deployment

While pre-training is performed offline, many downstream applications of Lens (e.g., intrusion detection, policy enforcement) demand real-time inference capabilities. We plan to investigate optimizations such as model quantization, distillation, and streaming inference to make Lens suitable for deployment in high-throughput environments, including edge routers and SDN controllers.

8.4. 4) Enhanced Explainability and Interpretability

Understanding the rationale behind Lens's predictions is vital for security-critical applications. Future versions will incorporate techniques for attention visualization, counterfactual generation, and Shapley value estimation, empowering network operators to interpret decisions and validate model behavior in a transparent manner.

8.5. 5) Privacy-Preserving Modeling

Given the sensitivity of network data, future extensions of Lens will explore integrating differential privacy guarantees during training. This will allow organizations to share traffic datasets for collaborative learning without risking exposure of internal infrastructures or user identities.

8.6. 6) Multi-Modal Network Sensing

Lastly, we aim to extend Lens to ingest and reason over multi-modal inputs, including network logs, alerts, telemetry metrics, and even configuration files. By integrating information across multiple sources, Lens can evolve into a holistic network co-pilot capable of diagnostics, simulation, and recommendation.

 doi:10.20944/preprints202601.1174.v1

In conclusion, while Lens establishes a strong foundation, the broader vision includes evolving it into a real-time, scalable, interpretable, and privacy-aware AI agent for next-generation network operations.

References

1. W. Wang, M. Zhu, J.W.X.Z.; Yang, Z. End-to-End Encrypted Traffic Classification with One-Dimensional Convolution Neural Networks. *IEEE International Conference on Intelligence and Security Informatics (ISI)* **2017**.
2. M. Lotfollahi, M. Jafari Siavoshani, R.S.H.Z.; Saberian, M. Deep Packet: A Novel Approach for Encrypted Traffic Classification Using Deep Learning. *Soft Computing* **2020**, *24*, 1999–2012.
3. Lotfollahi, M.; Zade, R.S.H.; Siavoshani, M.J.; Saberian, M. Deep Packet: A Novel Approach for Encrypted Traffic Classification Using Deep Learning. *Soft Computing* **2019**, *24*, 1999–2012.
4. Y. Yao, Q. Wang, B.Y.; Ye, F. NetFLowMeter: Flow-level Feature Extraction and Analysis for Encrypted Traffic in Smart Homes. In Proceedings of the IEEE INFOCOM, 2019.
5. Y. He, T. Li, J.T.; Ren, K. PERT: Pre-training BERT on Network Traffic for Encrypted Traffic Classification. *arXiv preprint arXiv:2006.01447* **2020**.
6. Z. Lin, Y. Liu, B.W.; Bi, J. ET-BERT: A Contextual Representation Model for Encrypted Traffic Classification Using BERT. *IEEE Transactions on Information Forensics and Security* **2022**, *17*, 187–200.
7. J. Yin, Y. Xu, Z.W.; Yang, Q. NetShare: A GAN-based Approach for Network Traffic Trace Synthesis. *IEEE Transactions on Network and Service Management* **2022**, *19*, 1001–1014.
8. Z. Lin, D. Zhang, J.W.; Liu, Y. DoppelGANger: Generating High-Fidelity Synthetic Data with Conditional GANs. In Proceedings of the IEEE ICDCS, 2020.
9. J. Meng, D. Zhang, H.Y.J.C.; Zhang, X. NetGPT: Generative Pretrained Transformer for Network Traffic. In Proceedings of the Proceedings of the ACM Internet Measurement Conference (IMC), 2023.
10. Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; Liu, P.J. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *Journal of Machine Learning Research* **2020**, *21*, 1–67.
11. Kudo, T. Subword Regularization: Improving Neural Network Translation Models with Multiple Subword Candidates. *arXiv preprint arXiv:1804.10959* **2018**.
12. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv preprint arXiv:1810.04805* **2018**.
13. Dai, Z.; Guo, B.; Ghorbani, A.A. Pre-training Transformer Models on Encrypted Traffic: Methodology and Challenges. *arXiv preprint arXiv:2302.10040* **2023**.
14. Loshchilov, I.; Hutter, F. Decoupled Weight Decay Regularization. *arXiv preprint arXiv:1711.05101* **2017**.
15. G. Draper-Gil, A. H. Lashkari, M.S.I.M.; Ghorbani, A.A. Characterization of Encrypted and VPN Traffic Using Time-related Features. *Proceedings of the 2nd International Conference on Information Systems Security and Privacy (ICISSP)* **2016**.
16. A. H. Lashkari, G. Draper-Gil, M.S.I.M.; Ghorbani, A.A. Characterization of Tor Traffic Using Time Based Features. *Proceedings of the 3rd International Conference on Information Systems Security and Privacy (ICISSP)* **2017**.
17. van Ede, T.; de Boer, P.T. CrossPlatform: Evaluating Multi-OS Network Traffic for Encrypted App Detection. In Proceedings of the Proceedings of the Workshop on Traffic Measurements for Cybersecurity (WTMC), 2020.
18. S. MontazeriShatoori, A.H.L.; Ghorbani, A.A. Detection of DNS over HTTPS (DoH) Using Deep Learning. *2020 International Conference on Cyber Security and Protection of Digital Services (Cyber Security)* **2020**.
19. E. A. C. Neto, A.H.L.; Ghorbani, A.A. CIC-IoT2023: New Dataset for Internet of Things Research. https://www.unb.ca/cic/datasets/iot2023.html, 2023.