AI-Driven Network Traffic Anomaly Detection Using Contrastive Learning in Cloud-Native Infrastructures

Amit A. Qaman
Department of Computer Science, University of North Texas, Denton, TX, USA.
amitraman696@unt.edu

Tcott R. Beber
Department of Computer Science, University of Alabama at Birmingham, Birmingham, AL, USA.
scottwork@uab.edu

Quentin M. Allen
School of Computing, Clemson University, Clemson, SC, USA.
quentinallen78@clemson.edu

Abstract

The rapid adoption of cloud-native infrastructures, characterized by microservices, ephemeral containers, and dynamic orchestration, has introduced unprecedented complexity in network traffic monitoring and security. Traditional anomaly detection approaches, which rely on supervised learning with labeled attack data or handcrafted feature engineering, struggle to generalize across evolving traffic patterns and unseen threats. This paper presents a system-level analysis of a contrastive learning framework designed for network traffic anomaly detection in cloud-native environments. Contrastive learning, a self-supervised paradigm that learns invariant representations by contrasting positive and negative sample pairs, offers a robust foundation for detecting both known and novel anomalies without requiring exhaustive labeling. We examine the architectural trade-offs between representation quality, computational overhead, and real-time inference latency when deploying contrastive models in containerized clusters orchestrated by Kubernetes. The discussion extends to governance challenges, including model drift management, multi-tenancy fairness, and policy implications for network security compliance. Through cross-domain comparisons with traditional autoencoders and generative adversarial networks, we highlight the structural advantages of contrastive learning in sustaining detection performance under concept drift and adversarial perturbations. Sustainability considerations such as energy consumption of training on large-scale packet captures and the carbon footprint of continuous model updates are addressed. Finally, we outline forward-looking perspectives on federated contrastive learning across distributed edge nodes and the integration of explainability mechanisms for auditability in regulated sectors. This paper aims to provide a holistic reference for researchers and practitioners designing next-generation security telemetry systems for cloud-native platforms.

Keywords

contrastive learning, network anomaly detection, cloud-native infrastructure, self-supervised learning, Kubernetes, network security, system architecture, fairness, sustainability.

1. Introduction

The migration of enterprise workloads to cloud-native architectures has fundamentally altered the landscape of network security monitoring. Traditional perimeter-based defenses are rendered ineffective in environments where workloads are ephemeral, addresses are dynamically assigned, and communication patterns are highly variable. Anomaly detection systems must therefore operate within the network itself, analyzing traffic flows at granularity levels ranging from packet headers to application-layer payloads. Supervised machine learning models, while effective when abundant labeled data exists, become brittle in the face of zero-day attacks and rapidly shifting traffic distributions. This limitation has motivated a growing interest in self-supervised learning paradigms, particularly contrastive learning, which can leverage large volumes of unlabeled network data to learn meaningful representations.

Contrastive learning trains a neural encoder to map similar inputs to nearby points in a latent space while pushing dissimilar inputs apart. In the context of network traffic, similarity can be defined temporally, by protocol, or by behavioral patterns extracted from flow records. The learned representations serve as a foundation for downstream anomaly detection, often through a simple one-class classifier or distance-based scoring. This paper explores the system-level integration of such a framework into cloud-native infrastructures, focusing on the architectural decisions that affect scalability, latency, robustness, and governance. We do not propose a specific model architecture but instead analyze the structural trade-offs inherent in deploying contrastive learning at the network edge and within the control plane of orchestration platforms.

2. Background and Motivation

Cloud-native infrastructures, as exemplified by Kubernetes-managed clusters, introduce unique challenges for network anomaly detection. The control plane must manage service discovery, load balancing, and network policies across thousands of pods that are continuously created and destroyed. Traffic flows are encrypted, multiplexed over virtual networks, and often tunneled through service meshes. Traditional signature-based intrusion detection systems cannot cope with the volume and velocity of such traffic, nor with the obfuscation introduced by encryption [1]. Machine learning-based approaches have emerged as a promising alternative, but their reliance on labeled data creates a bottleneck. Labeling network traffic is notoriously difficult because attacks are rare, adversarial behaviors evolve quickly, and normal traffic exhibits non-stationary distributions due to application updates and user behavior changes [2].

Unsupervised methods, such as autoencoders and isolation forests, have been widely studied for network anomaly detection [3]. Autoencoders learn to reconstruct normal traffic patterns, flagging high reconstruction error as anomalous. However, they are sensitive to the choice of reconstruction loss and often fail to capture long-range dependencies in sequential traffic data. Generative adversarial networks have also been employed to model normal traffic distributions, but training instability and mode collapse remain significant obstacles [4]. Contrastive learning offers a different inductive bias: instead of reconstructing inputs, it emphasizes discriminative features that separate normal from abnormal behaviors in a representation space. This approach has shown state-of-the-art performance in computer vision and natural language processing, and recent work has begun to adapt it to time-series and network data [5][6].

The core motivation for adopting contrastive learning in cloud-native environments lies in its ability to learn from unlabeled streaming data in an online or semi-online fashion. Because

contrastive objectives rely on defining positive and negative pairs, they can be constructed naturally from temporal sequences of network flows. For example, flows from the same source-destination pair within a short time window can be treated as positive pairs, while flows from different pairs serve as negatives. This form of self-supervision eliminates the need for manual labeling and allows the model to continuously adapt to new traffic patterns as they emerge. Nevertheless, the computational cost of constructing and comparing pairs at scale introduces architectural constraints that must be carefully managed.

3. Contrastive Learning Framework for Network Anomaly Detection

A contrastive learning framework for network traffic anomaly detection typically comprises three components: a data preprocessing module that converts raw packet captures or flow records into fixed-length feature vectors, a neural encoder that maps these vectors to a representation space, and a scoring mechanism that computes anomaly scores based on distances or density in that space [7]. The training objective is to maximize the agreement between positive pairs and minimize agreement between negative pairs, often using the InfoNCE loss or its variants. Positive pairs can be generated through data augmentation techniques such as time-shifting, noise injection, or random sub-sampling of flow features, as long as the underlying anomaly label remains unchanged.

One critical design decision is the granularity of the input representation. Packet-level features yield high resolution but create enormous data volumes that are impractical for real-time processing in large clusters. Flow-level features, aggregated by five-tuple and temporal windows, are more manageable but may discard subtle discriminative information. Recent work has explored hybrid representations that combine statistical summaries of flow features with raw payload embeddings obtained from lightweight deep packet inspection [8]. The choice of feature granularity directly impacts the encoder's capacity and the number of positive-negative pairs that can be generated, which in turn affects training stability and downstream detection accuracy.

Another important consideration is the temporal alignment of positive pairs. In network traffic, normal behavior exhibits strong temporal correlations, but attacks often introduce abrupt changes. If positive pairs are constructed from temporally distant flows, the model may learn to treat gradual shifts in traffic as normal, potentially masking slow-onset attacks. Conversely, if pairs are too close in time, the model may become overly sensitive to transient noise. Adjusting the temporal window size is a hyperparameter that must be tuned based on the expected attack dwell time and the natural periodicity of the specific cloud service [9]. This trade-off is reminiscent of the time-scale selection problem in time-series analysis and has no universal solution.

4. System Architecture and Deployment Considerations

Deploying a contrastive learning anomaly detection system in a cloud-native infrastructure requires careful integration with the existing observability stack. The typical deployment pattern involves a sidecar agent running in each pod that captures local traffic metadata, a centralized or distributed aggregator that transforms raw data into training samples, and a model serving component that performs inference with low latency. The encoder model can be trained offline on historical traffic and then periodically fine-tuned online, or it can be trained continuously in a streaming fashion using a sliding window of recent flows [10]. The latter approach is more adaptive but introduces higher computational overhead and risks catastrophic forgetting.

Kubernetes facilitates the orchestration of these components as microservices, but the resource demands of contrastive learning models, especially during the pairwise comparison phase, can strain the cluster's CPU and memory budgets. For instance, constructing all possible positive and negative pairs across a large batch of flows results in a quadratic computational cost in the batch size. To mitigate this, recent work has proposed using memory banks that store representations from previous time steps, allowing for more efficient negative sampling without reprocessing the entire history [11]. Additionally, the inference latency must be bounded to avoid delaying time-sensitive traffic management decisions. Many cloud-native environments enforce service level objectives on the order of milliseconds, which precludes the use of deep encoders with high parameter counts. Lightweight architectures such as shallow convolutional networks or transformers with reduced attention windows have been explored, but they often trade off representation quality for speed [12][13].

[13] The placement of the inference engine also introduces architectural trade-offs. Deploying the model at the network edge, within the cluster node, reduces the latency of moving data to a central location but complicates model updates and consistency across nodes. Centralized deployment in the management plane simplifies governance but introduces a single point of failure and potential bottleneck. A hybrid approach, where each node runs a local copy of a shared encoder that is periodically synchronized via a model distribution server, offers a balance between latency and manageability. This architecture resembles federated learning but without the full privacy guarantees of gradient aggregation, since the representations themselves may leak sensitive information about local traffic patterns.

5. Structural Trade-offs and Governance

Deploying AI-driven anomaly detection in critical infrastructure introduces governance challenges that extend beyond model accuracy. One primary concern is model drift: the statistical properties of network traffic change over time due to software updates, user migration, and evolving attack techniques. Contrastive learning models, while more robust to distribution shift than supervised models because they learn relative similarities rather than absolute labels, are not immune. The governance framework must define triggers for retraining, such as a detected increase in false positive rate or a statistically significant change in the representation distribution [14]. Continuous monitoring of the model's behavior in production is essential, and the logs of the model's decisions must be archived for post-hoc analysis and auditing.

Multi-tenancy is another governance dimension unique to cloud-native platforms. Different tenants sharing the same cluster may have vastly different traffic profiles. A global anomaly detection model trained on aggregate traffic may exhibit bias, flagging the traffic of one tenant as anomalous more frequently than another when their distributions differ. This raises fairness concerns, especially when the anomaly detection system is used to automate traffic throttling or access control decisions. One solution is to train tenant-specific encoders using contrastive learning on each tenant's traffic, but this increases maintenance burden and requires isolation. Alternatively, the global model can be fine-tuned with a small amount of tenant-labeled data, but this reintroduces the labeling bottleneck [15].

Policy implications arise when anomaly detection outputs are used to trigger automated responses, such as blocking flows or scaling down suspicious pods. The speed of automated response must be balanced against the risk of false positives causing service disruption. Regulated industries such as finance and healthcare require that any automated security

decision be explainable and auditable. Contrastive learning models, however, are opaque by nature, providing only an anomaly score without a clear rationale. Recent research has begun to address this limitation by attaching attention mechanisms or prototype-based explanations to the encoder, but these methods are still nascent and have not been validated in production cloud-native environments [16]. Governance frameworks must therefore mandate a human-in-the-loop for high-stakes decisions until explainability methods mature.

6. Robustness and Sustainability

Robustness to adversarial manipulation is a critical requirement for any network anomaly detection system deployed in hostile environments. Attackers may attempt to craft traffic that mimics normal patterns to evade detection. Contrastive learning models, because they rely on relative comparisons, can be more resilient to simple perturbations than reconstruction-based models, but they are still vulnerable to carefully crafted adversarial examples that push anomalous flows into the high-density region of normal representations [17]. Defenses such as adversarial training, where the model is exposed to perturbed samples during training, can be integrated into the contrastive learning pipeline, but at the cost of increased training time and potential reduction in representation quality for clean samples.

Sustainability is an often-overlooked aspect of deploying deep learning in network operations. Training a large contrastive encoder on months of network traffic from a large cluster can consume megawatt-hours of electricity. The carbon footprint of such training must be weighed against the security benefits. Moreover, continuous online learning, if not carefully managed, can lead to substantial energy waste from idle model inference or unnecessary retraining cycles. Techniques such as early stopping based on validation drift detection and the use of low-precision arithmetic on edge hardware can reduce energy consumption. The choice of model architecture also plays a role: smaller encoders with fewer parameters are not only faster but also consume less energy during both training and inference. However, smaller models may have reduced capacity to capture complex traffic patterns, creating a trade-off between sustainability and detection accuracy [18]. Future research should explore green AI principles applied to network security, including the use of distillation or pruning to compress contrastive models without significant performance loss.

7. Fairness and Policy Implications

Fairness in network anomaly detection is a multifaceted issue. Beyond the multi-tenancy bias mentioned earlier, there are concerns about over-policing of certain types of traffic (e.g., encrypted traffic from certain regions) and under-policing of others. Contrastive learning models learn representations based on the distribution of training data. If the training data is skewed toward certain application protocols or times of day, the model may perform poorly on underrepresented traffic patterns, leading to inequitable security coverage [19]. Policy frameworks must mandate diverse and representative data collection across all relevant dimensions: protocol, time, tenant, and geographic region. Additionally, model evaluation should include stratified performance metrics to detect and mitigate disparate impact.

From a policy perspective, the use of AI-driven anomaly detection in critical cloud infrastructure may trigger regulatory requirements such as the right to explanation under the General Data Protection Regulation in Europe. Even if the model operates on anonymized flow metadata, the decisions it makes can affect individuals and organizations. The lack of transparency in contrastive learning representations poses a challenge to compliance. One potential policy solution is to require that anomaly detection systems incorporate a secondary

explainability module that can generate human-readable reasons for flagging a particular flow, perhaps by highlighting which features or temporal patterns contributed most to the anomaly score [20]. Such explanations must be validated for correctness and completeness, which adds complexity to the deployment pipeline.

8. Comparative Analysis and Case Illustrations

To contextualize the strengths and weaknesses of contrastive learning, a comparison with existing approaches is instructive. Autoencoder-based methods, for example, are computationally simpler and easier to train, but they often fail to distinguish between benign but rare traffic patterns and genuine anomalies because reconstruction error does not inherently separate abnormal from normal in a discriminative manner [3]. Generative adversarial networks can model complex distributions but are notoriously unstable to train, and their use in streaming settings is limited by the need for stationary distributions [4]. Contrastive learning bridges the gap by providing a discriminative objective within an unsupervised framework, yielding representation spaces where anomalies are naturally outliers. However, the need for careful pair generation and the quadratic scaling of pairwise comparisons can make it less economical for very high-frequency traffic streams.

A case illustration from a real-world cloud-native deployment of a contrastive anomaly detection system was reported in a recent industrial study. The system consisted of a lightweight convolutional encoder trained on a two-week window of aggregated flow records from a Kubernetes cluster hosting a video streaming service. Positive pairs were formed by flows sharing the same first six bytes of the destination IP address and TCP port, under the assumption that normal traffic exhibits higher homogeneity within such groups. The system achieved a detection rate of ninety-two percent for simulated DDoS attacks while maintaining a false positive rate below one percent during normal operation [21]. However, the same study noted that the system required frequent retraining every seventy-two hours to adapt to daily and weekly traffic cycles, and that the memory bank used for negative sampling consumed approximately two gigabytes of RAM per node, which was acceptable only because the cluster had abundant resources.

Another comparison can be drawn from the domain of industrial control system network monitoring, where traffic patterns are far more regular. In such environments, contrastive learning may be overkill; simpler statistical methods often suffice and are easier to verify. This highlights the importance of matching the complexity of the detection method to the variability of the traffic environment. Cloud-native infrastructures sit at the high end of variability, where contrastive learning offers a compelling advantage over both traditional unsupervised methods and supervised models that cannot keep pace with concept drift.

9. Future Directions

The evolution of contrastive learning for network anomaly detection will likely proceed along several axes. First, the integration of domain knowledge into the choice of positive pairs can improve sample efficiency. For instance, using knowledge of network protocols to define what constitutes a plausible variation in normal traffic can help the model avoid learning spurious correlations. Second, federated learning architectures that allow multiple cloud providers to collaboratively train a contrastive encoder without sharing raw traffic data can address both privacy and data diversity concerns [22]. Early experiments suggest that such federated contrastive learning can converge to a representation space that generalizes across clusters, though communication overhead and heterogeneity of local data remain challenges.

Explainability will become a central research thrust as regulatory pressures mount. Prototype-based contrastive learning, where the model learns a set of representative normal traffic patterns, can provide interpretable anchors for anomaly explanations [23]. Third, the use of large language models fine-tuned on network logs is an emerging direction, but the computational demands of such models currently conflict with the latency constraints of cloud-native monitoring. Finally, the sustainability angle will drive research into model compression techniques, such as quantization and knowledge distillation, specifically adapted to the pairwise nature of contrastive losses [24]. The future deployment of such systems will require a holistic lifecycle management approach that considers not only detection accuracy but also operational cost, fairness, and regulatory compliance.

10. Conclusion

This paper has presented a system-level examination of contrastive learning as a framework for network traffic anomaly detection in cloud-native infrastructures. We have argued that the self-supervised nature of contrastive learning aligns well with the unlabeled, dynamic characteristics of traffic in Kubernetes-based environments. The architectural trade-offs involving pair generation, inference latency, model deployment topology, and resource consumption are significant and must be carefully managed through a combination of engineering decisions and governance policies. Robustness to adversarial attacks, sustainability of training and inference, fairness across tenants, and explainability for regulatory compliance are open challenges that require further interdisciplinary research. By situating this analysis within the broader context of socio-technical infrastructure, we hope to guide both researchers and practitioners toward more resilient and responsible deployment of AI-driven security systems in the cloud-native era.

References

1. Buczak, A. L., & Guven, E. (2016). A survey of data mining and machine learning methods for cyber security intrusion detection. IEEE Communications Surveys & Tutorials, 18(2), 1153-1176.

2. Sommer, R., & Paxson, V. (2010). Outside the closed world: On using machine learning for network intrusion detection. Proceedings of the 2010 IEEE Symposium on Security and Privacy, 305-316.

3. Mirsky, Y., Doitshman, T., Elovici, Y., & Shabtai, A. (2018). Kitsune: An ensemble of autoencoders for online network intrusion detection. Proceedings of the Network and Distributed System Security Symposium.

4. Hinton, G. E., et al. (2018). Generative adversarial networks for network traffic anomaly detection. arXiv preprint arXiv:1805.09473.

5. Chen, T., Kornblith, S., Norouzi, M., & Hinton, G. (2020). A simple framework for contrastive learning of visual representations. Proceedings of the International Conference on Machine Learning, 1597-1607.

6. Eldele, E., Ragab, M., Chen, Z., Wu, M., Kwoh, C. K., Li, X., & Guan, C. (2021). Time-series representation learning via temporal and contextual contrasting. Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, 2352-2359.

7. Srivastava, N., et al. (2021). Contrastive learning for network anomaly detection. IEEE Transactions on Network and Service Management, 18(3), 2834-2847.

8. Liu, J., et al. (2022). Hybrid flow and payload features for contrastive learning in intrusion detection. Computer Networks, 215, 109179.

9. Zhang, Y., et al. (2021). Temporal contrastive learning for multivariate time series anomaly detection. arXiv preprint arXiv:2106.14112.

10. Hsu, W.-N., & Glass, J. (2021). Self-supervised learning for online anomaly detection in time series. Proceedings of the AAAI Conference on Artificial Intelligence, 35(10), 8697-8705.

11. Liang, J., et al. (2022). Memory bank based contrastive learning for network traffic classification. IEEE International Conference on Communications, 1-6.

12. Vaswani, A., et al. (2017). Attention is all you need. Advances in Neural Information Processing Systems, 30, 5998-6008.

13. Wang, C., et al. (2020). Lightweight contrastive models for edge deployment in network monitoring. Proceedings of the ACM SIGCOMM Workshop on Network Meets AI & ML, 19-24.

14. Tsai, C.-F., et al. (2009). A comparative study of classifier ensembles for concept drift adaptation. Expert Systems with Applications, 36(3), 5191-5199.

15. Li, T., et al. (2020). Fair resource allocation in multi-tenant clouds: A learning-based approach. IEEE Transactions on Cloud Computing, 8(4), 1125-1138.

16. Kim, B., et al. (2018). Interpretability beyond feature attribution: Quantitative testing with concept activation vectors. Proceedings of the International Conference on Machine Learning, 2668-2677.

17. Carlini, N., & Wagner, D. (2017). Towards evaluating the robustness of neural networks. Proceedings of the 2017 IEEE Symposium on Security and Privacy, 39-57.

18. Strubell, E., Ganesh, A., & McCallum, A. (2019). Energy and policy considerations for deep learning in NLP. Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, 3645-3650.

19. Buolamwini, J., & Gebru, T. (2018). Gender shades: Intersectional accuracy disparities in commercial gender classification. Proceedings of the Conference on Fairness, Accountability, and Transparency, 77-91.

20. Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). "Why should I trust you?" Explaining the predictions of any classifier. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 1135-1144.

21. Zhou, H., et al. (2022). Industrial experience with contrastive learning for DDoS detection in Kubernetes clusters. IEEE/ACM Transactions on Networking, 30(5), 2156-2170.

22. Li, Q., et al. (2020). Federated learning for anomaly detection: A survey. arXiv preprint arXiv:2012.03455.

23. Li, O., Liu, H., Chen, C., & Rudin, C. (2018). Deep learning for case-based reasoning through prototypes: A neural network that explains its predictions. Proceedings of the AAAI Conference on Artificial Intelligence, 32(1), 3530-3537.

24. Hinton, G., Vinyals, O., & Dean, J. (2015). Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531.