

Parallel Micro-Batching and Scalable Inferencing for ML-Based Malicious Traffic Detection

Achmad Basuki¹, Widhi Yahya¹, Dzaki R. Malik¹, Rizal Setya Perdana¹, Kasyful Amron¹,
Achmad Husni Thamrin², Andrey Ferriyan² and Muhammad Niswar³

¹*Universitas Brawijaya, Indonesia*

²*Keio University, Japan*

³*Universitas Hasanuddin, Indonesia*
fi

Keywords: NIDS, Machine Learning, High-speed Network, Passive Optical Tapping, Scalability.

Abstract: Network intrusion detection systems based on machine learning (ML-IDS) face significant challenges in high-speed network environments, such as gigabit-scale packet capture and real-time inference under dynamic traffic conditions. Efficiently handling these challenges is critical to maintaining accurate and timely detection without overwhelming the system resources. This paper presents a scalable ML-IDS architecture featuring a novel parallel micro-batching inference framework integrated with passive optical tapping for non-intrusive traffic monitoring. The proposed inference architecture is critical to achieving a balance between high classification accuracy and computational efficiency. Experimental results demonstrate a 2.65× throughput improvement over traditional sequential processing while maintaining sub-5ms decision times, even under variable traffic loads. Furthermore, the architecture supports horizontal scaling to accommodate growing network demands, ensuring sustained low-latency detection performance. These contributions establish a robust foundation for deploying ML-IDS in high-speed network environments.

1 INTRODUCTION

The rapid advancement of computer networking technologies, including digital twins, has enabled the simulation and optimization of complex systems across industrial, residential, banking, and educational sectors—leading to a surge in network users and growing attack surfaces. As a result, cyber threats have become increasingly sophisticated, posing serious risks to individuals, organizations, and society by compromising the core tenets of information security: Confidentiality, Integrity, and Availability (CIA), with consequences ranging from financial and reputational damage to threats to critical infrastructure and human safety (Thakkar and Lohiya, 2023). In response, institutions such as universities and industries are enhancing their network security to provide safer Internet access by managing both controlled factors (e.g., human behavior, access regulation through formal procedures, and policy enforcement) and uncontrolled ones, such as anomalous traffic and malware propagation. To address these challenges, organizations increasingly rely on security technologies like firewalls,

Intrusion Detection Systems (IDS), and Intrusion Prevention Systems (IPS), which play a critical role in identifying and mitigating threats to strengthen overall network resilience (Admass et al., 2024).

Network Intrusion Detection Systems (NIDS) are essential for detecting and responding to anomalous traffic using two primary approaches: signature-based and behavioral (anomaly-based) detection. While signature-based methods rely on predefined rules to identify known threats, they fail to detect novel attacks until updates are applied, creating security gaps. In contrast, behavioral detection leverages artificial intelligence (AI) and machine learning (ML) to model normal network behavior and identify deviations, enabling real-time detection of previously unseen threats. This adaptability makes ML-based NIDS critical in combating evolving cyber threats. Recent studies show that integrating advanced ML techniques significantly improves detection accuracy and reduces false positives, reinforcing their role in modern cybersecurity (Abdulganiyu et al., 2023).

ML-based anomaly detection has emerged as a promising approach for enhancing network security,

offering the ability to learn complex patterns, adapt to evolving threats, and outperform traditional rule-based systems that rely on static signatures. While numerous studies (Dina and Manivannan, 2021; Abdelmoumin et al., 2022; Sahani et al., 2023; Yang and Shami, 2022; Gite et al., 2023; Yu et al., 2024; Gomez et al., 2023; Zeng et al., 2024; Theofanous et al., 2024; Kondaiah et al., 2024; Siyyal et al., 2022; Zang et al., 2024; Ucci et al., 2021; Berbecaru and Petraglia, 2023) have explored ML-based Intrusion Detection Systems (IDS), most evaluate performance in simulation environments, limiting real-world applicability. Even those that use real testbeds (Siyyal et al., 2022; Zang et al., 2024; Ucci et al., 2021; Berbecaru and Petraglia, 2023) often overlook critical challenges in high-throughput networks—particularly computational overload and excessive inference latency—leading to delayed detection and reduced effectiveness under realistic traffic loads.

To address these limitations, we propose a novel ML-based NIDS architecture designed for low-latency, high-throughput operation. The system is divided into two optimized components: (1) traffic capture and preprocessing, and (2) model inference. For the first, we employ passive optical tapping to capture traffic without disrupting production networks, combined with the Argus tool for real-time feature extraction. This ensures timely, lightweight data preparation with minimal overhead. For inference, we introduce a parallel micro-batching architecture that improves detection throughput and scalability, enabling rapid, accurate threat identification in high-speed environments. By co-optimizing both stages for minimal decision time, our design enhances system responsiveness and practical deployability in operational networks.

This paper proposes a high-throughput ML-based NIDS architecture designed for real-time operation in high-speed networks without disrupting production traffic. The system integrates passive optical tapping with Argus for efficient, real-time feature extraction, ensuring timely and relevant input to the detection model. To maximize scalability and detection performance, we introduce a parallel micro-batching architecture for inference, enabling rapid processing of streaming traffic under demanding conditions. Together, these contributions enhance the practicality, responsiveness, and accuracy of intrusion detection in operational network environments.

2 RELATED WORK

In the realm of simulated environments for Machine Learning-based Intrusion Detection Systems (IDS), several studies have provided valuable insights. Traditional approaches employ hybrid signature-anomaly detection (Dina and Manivannan, 2021), ensemble methods combining PCA with deep learning (Abdelmoumin et al., 2022), and diverse ML techniques including SVMs, decision trees, and neural networks (Sahani et al., 2023; Yang and Shami, 2022). Recent advances explore specialized architectures such as graph-based systems (Yu et al., 2024), unsupervised clustering with HDBSCAN (Gomez et al., 2023), and ensemble models integrating deep learning with self-attention mechanisms (Kondaiah et al., 2024). However, these simulation-based studies focus primarily on maximizing detection accuracy while neglecting critical real-world constraints: inference latency, traffic capture efficiency, and end-to-end detection time optimization.

Real testbed implementations reveal additional limitations. Studies using traditional packet capture methods (Wireshark, tcpdump) (Siyyal et al., 2022; Kondaiah et al., 2024) face scalability challenges at high network speeds, while computationally intensive approaches like NLP-inspired deep learning (Zang et al., 2024) introduce prohibitive processing delays. Existing monitoring solutions integrate multiple tools (Suricata, Zeek) (Berbecaru and Petraglia, 2023) but suffer from deployment complexity and lack optimization for high-throughput scenarios. Notably, no prior work addresses the joint optimization of passive traffic capture, ML inference efficiency, and detection time minimization specifically for 10Gbps networks. Our approach uniquely combines passive optical tapping for zero-latency capture with a two-tier architecture optimizing both network-level filtering and ML-based detection, explicitly targeting sub-second response times in high-speed operational environments.

3 SCALABLE ML-BASED IDS FOR HIGH-SPEED NETWORKS

In this section, we explain the overall design of our proposed solution and describe it in detail with a primary goal to achieve real-time detection with high throughput.

3.1 Design Issues

3.1.1 Gigabit Scale Capturing System

In high-speed network environments, such as enterprise or campus network backbones, the sheer volume of traffic necessitates a robust capturing system. The system must collect all traffic with zero packet loss to ensure comprehensive monitoring. Subsequently, live preprocessing of this high-bandwidth data stream is required to reduce its size and complexity. Finally, the preprocessing must efficiently extract only the features relevant to the subsequent inference model, presenting a significant challenge given the real-time constraints. Meeting these requirements simultaneously is crucial for effective intrusion detection.

3.1.2 Near-Real Time Inferencing System

In high-speed networks, the data stream captured is typically processed through sequential inferencing, where each network flow is analyzed in isolation. While this serial processing approach is conceptually straightforward, it introduces substantial delays. The cumulative effect of processing each flow individually results in significantly increased response times, which hinders the timely detection and mitigation of security threats. Such delays can have severe consequences, as they provide attackers with the opportunity to exploit vulnerabilities and inflict considerable damage before the intrusion detection system can respond effectively.

3.2 Proposed IDS System

3.2.1 Passive Capture Techniques with Live Preprocessing for High-Speed Networks

To address gigabit-scale data capture, we employ an LC-to-LC optical tap (OP-TAP) as shown in Fig. 1a, enabling seamless integration into high-speed enterprise or campus networks. The device uses dual live connections (LIVE1 and LIVE2) to passively capture both transmitted (TX) and received (RX) traffic without introducing computational overhead or disrupting production networks—ensuring lossless, real-time monitoring. Captured traffic is forwarded to a dedicated server (Server Netmon) for live preprocessing, enhancing the efficiency of downstream analysis and intrusion detection as shown in Fig. 1b. This setup forms the second tier of our two-tier architecture, where the first tier (firewall) performs initial threat filtering, reducing load on the detection system. The second tier—comprising the OP-TAP and Server Netmon—handles traffic capture, preprocessing, and

real-time analysis as a unified, low-latency pipeline. This separation of concerns ensures scalability and responsiveness in high-throughput environments, meeting the demanding requirements of modern network security.

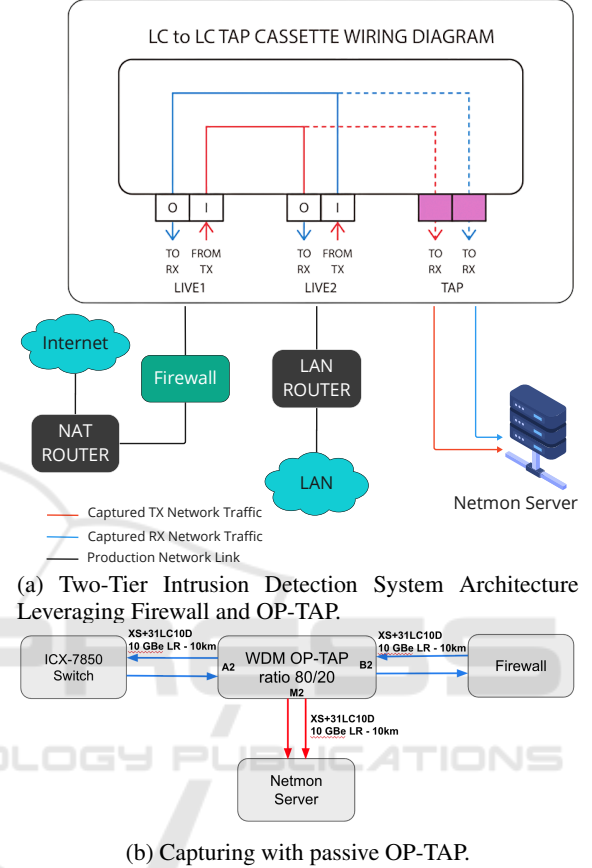


Figure 1: Network capture system implementation.

3.2.2 Parallel Micro-Batching and Scalable Inferencing Architecture

Fig. 2 illustrates the system architecture, which is organized into three key modules to enhance throughput by processing data in micro-batches rather than streams, thereby reducing latency. The workflow comprises two main components: a micro-batching module and an inferencing module, with each inferencing instance connected to a machine learning model for classifying traffic as benign or malicious. The micro-batching module receives incoming data via multiple threads, aggregates it into fixed-size batches, and publishes them to a message queue under a specific topic. The inferencing module subscribes to this topic, polls the queue, and forwards each batch to the ML model for prediction. This decoupled, parallel design enables efficient load balancing, low-latency processing, and scalable real-time inference.

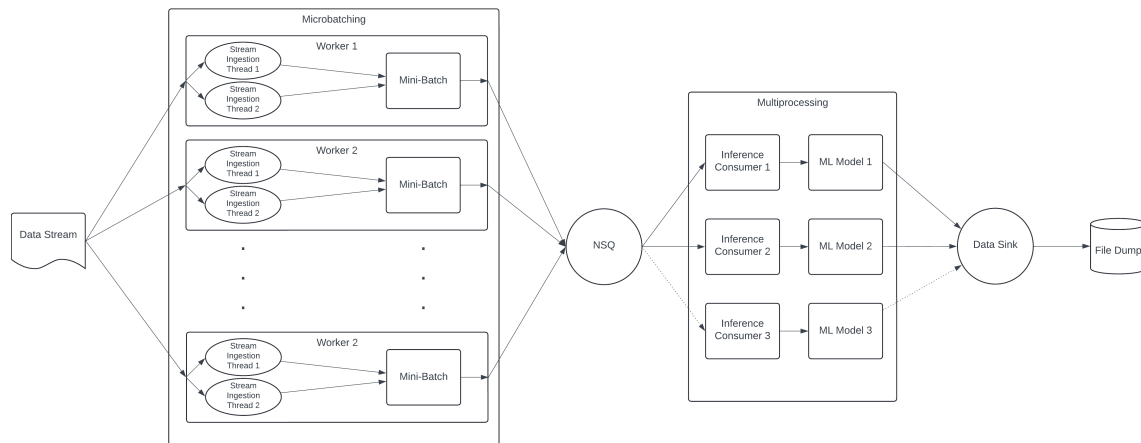


Figure 2: Architecture of the proposed method.

4 IMPLEMENTATION

4.1 Implementation of Gigabit Scale Capturing System

To implement the gigabit-scale capture system, we deployed the setup shown in Fig. 1b. An ICX-7850 switch connects via fiber to a WDM Optical Tap 80/20, which passively splits traffic—80% directed to the firewall, 20% to the traffic capture server—enabling non-intrusive monitoring with zero impact on production performance. The capture server (Netmon) is an HP ProLiant DL380 Gen10 equipped with an Intel Xeon Silver 4214R (12-core), 32 GB RAM, NVIDIA RTX A2000 12 GB GPU, and an Intel Ethernet Converged Network Adapter X710-DA4 (quad-port 10GbE SFP+). Two ports are used to capture TX and RX streams from the tap, while the system runs Ubuntu 22.04 and leverages Argus (<https://openargus.org/>) for flow merging and preprocessing. Using the `-i dup:interface-name` flag, Argus aggregates bidirectional traffic into a unified flow and extracts features in real time. This configuration ensures lossless capture, high throughput, and minimal overhead—aligning with our design goals of passive, scalable, and low-latency preprocessing in 10 Gbps environments.

4.2 Implementation of Near-Real-Time Inferencing System

The near-real-time inferencing system leverages open-source technologies to achieve high throughput and low latency. ZeroMQ (<https://zeromq.org/>) enables asynchronous, non-blocking communication between stream ingestion workers and the micro-

batching module, supporting scalable data transfer in high-speed networks. Redis (<https://redis.io/>) acts as an in-memory FIFO queue, temporarily storing micro-batched data with minimal delay, thereby preventing processing bottlenecks. Inference results are persisted in DuckDB (<https://duckdb.org/>), an embedded analytical database that supports efficient querying and lightweight data management for downstream analysis. Together, these components form a streamlined, scalable pipeline that implements the parallel micro-batching architecture Fig. 3, significantly reducing response times and enhancing the system's detection capabilities.

5 RESULTS AND DISCUSSION

This section evaluates the effectiveness of our scalable ML-based IDS with a focus on optimizing inferencing time without compromising detection performance.

5.1 Survey of Machine Learning Models and Feature Selection for Optimized Inferencing

The evaluation leverages multiple network traffic datasets—NF-CSE-CIC-IDS2018-v2, NF-ToN-IoT-v2, CIC-IDS2017, NF-UNSW-NB15-v2, and HIKARI 2021—to enhance model generalization across diverse traffic and attack patterns. The combined dataset is split 80%/20% for training and testing, ensuring robust performance assessment.

Table 1 summarizes the performance of various ML models across key metrics. Gradient boosting methods—XGBoost (XGB), LightGBM (LGBM),

and CatBoost (CatB)—achieve near-perfect or perfect scores, with XGB attaining 1.00 across all metrics, attributed to its ability to model complex feature interactions via sequential tree learning. In contrast, simpler models such as Naive Bayes (NVB) and Linear Discriminant Analysis (LDA) underperform, with NVB reaching only 78% accuracy.

Given these results, XGBoost (XGB) is selected for deployment due to its perfect accuracy and favorable inference characteristics: (1) native handling of categorical features, (2) effective tree pruning that reduces model size, and (3) strong parallelization support—well aligned with our micro-batching architecture. These attributes enable high accuracy with minimal inference latency in high-speed networks. For feature selection, we leverage importance weights from training, as shown in Fig. 3, retaining only the most impactful features (e.g., *Rate*, *Dur*, *Dport*, *State*). This approach reduces dimensionality, streamlines preprocessing, and enhances model efficiency for near-real-time detection.

Table 1: Model Survey Results.

Model	Accuracy	Precision	Recall	F1 Score
NVB	0.78	0.76	0.82	0.79
LR	0.92	0.91	0.93	0.92
LDA	0.88	0.87	0.89	0.88
SVM	0.93	0.92	0.94	0.93
XGB	1.00	1.00	1.00	1.00
LGBM	0.99	0.99	0.99	0.99
CatB	0.99	0.99	0.99	0.99
AdaB	0.98	0.97	0.98	0.97

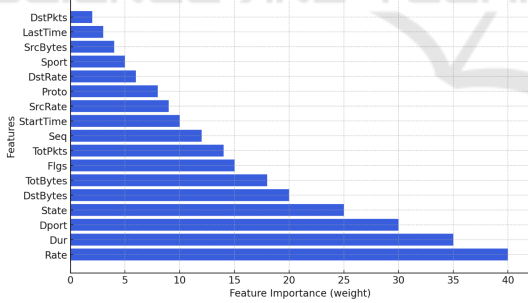


Figure 3: Feature Importance (Weight).

5.2 Performance Evaluation of Inferencing and Capturing System

This subsection presents experimental results on inferencing latency and capturing time. It compares the performance of the proposed parallel micro-batching inferencing system against traditional sequential inferencing, showcasing the improvements in processing speed and system responsiveness enabled by our architecture. As illustrated in Figure 4, the live in-

ferencing process provides real-time network performance metrics alongside AI-based classification results, demonstrating the system’s capability to analyze traffic flows and detect anomalies efficiently.

```
> ./replay.sh
File Cache is enabled
Actual: 3667090 packets (1666832286 bytes) sent in 5.30 second
s
Rated: 314437329.9 Bps, 2515.49 Mbps, 691773.25 pps
Flows: 2844520 flows, 536598.21 fps, 3593221 flow packets, 738
69 non-flow

Received Line from Argus
Inference result: Benign. Inference time: 0.00697062600011224
Received line from Argus
Inference result: Benign. Inference time: 0.0049558890004846035
Received line from Argus
Inference result: Benign. Inference time: 0.0034334120000494295
Received line from Argus
Inference result: Benign. Inference time: 0.003261050999753934
Received line from Argus
Inference result: Malicious. Inference time: 0.0031899989999146783
Received line from Argus
Inference result: Benign. Inference time: 0.0031939890004650806
Received line from Argus
Inference result: Malicious. Inference time: 0.003369609999936074
```

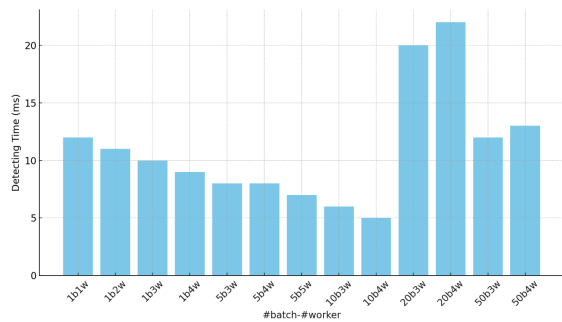
Figure 4: Live inferencing process and performance summary.

Figs. 5a and 5b analyze the trade-off between throughput (flow/s) and decision time (ms) across batch size and worker count configurations. The 10b4w setup achieves an optimal balance, delivering 730 flows/s at just 5 ms decision time—representing a 265% throughput gain over the baseline 1b1w (200 flows/s). Larger configurations (e.g., 20b3w, 50b3w) further increase throughput (up to 1300 flows/s, +450%), but at the cost of elevated decision times (tens of ms), indicating diminishing returns due to Netmon server limitations.

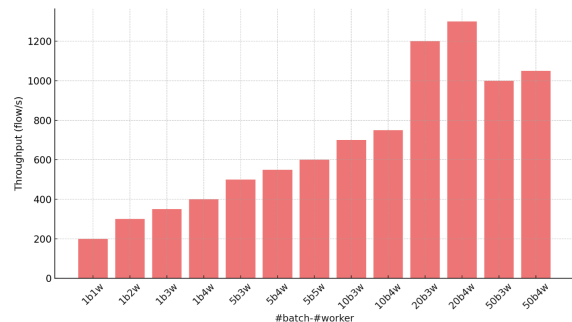
This highlights a key design insight: while single-server performance is strong, scalability is bounded by resource saturation. To address this, our architecture supports distributed deployment with multiple inferencing servers—a path for maintaining low latency under growing workloads. This forward-looking scalability is a core strength, enabling adaptation to larger, more complex networks. Future work will explore multi-server configurations to optimize the throughput-latency trade-off and fully realize the system’s scalable potential.

6 CONCLUSIONS

The proposed architecture advances high-speed network intrusion detection through three key contributions. First, the parallel micro-batching inferencing system achieves up to 730 flows/s—a 265% improvement over sequential processing—while maintaining a low decision time of 5 ms, effectively balancing throughput and latency. Second, XGBoost



(a) Average detecting time per configuration.



(b) Average throughput per configuration.

Figure 5: Performance comparison of detecting time and throughput.

delivers perfect accuracy, precision, recall, and F1-score, enabling high-confidence detection without compromising inference speed. Third, the architecture is inherently scalable, supporting distributed deployment across multiple inferencing servers to sustain performance under growing network loads. Together, these results demonstrate a practical, high-performance NIDS solution for real-time, large-scale environments.

Future work will explore multi-server scaling and distributed processing frameworks to further enhance throughput. We will also integrate MLOps practices for automated model monitoring, retraining, and deployment, ensuring long-term reliability. Additionally, federated learning will be investigated to enable privacy-preserving, collaborative model training across distributed network nodes, improving adaptability and robustness in heterogeneous environments.

ACKNOWLEDGEMENTS

This work is part of the project “Real-time Malicious TLS Traffic Detection using a Machine Learning Classifier,” supported by Keio University and the APNIC Foundation under the Community Based Research (CBR) scheme (2023–2025). The authors used AI tools solely for language editing, including proofreading for grammar, syntax, and clarity. All research design, methods, implementation, and results were independently developed by the authors.

REFERENCES

Abdelmoumin, G., Rawat, D. B., and Rahman, A. (2022). On the Performance of Machine Learning Models for Anomaly-Based Intelligent Intrusion Detection Sys-

tems for the Internet of Things. *IEEE Internet of Things Journal*, 9(6):4280–4290.

Abdulganiyu, O. H., Ait Tchakouch, T., and Saheed, Y. K. (2023). A Systematic Literature Review for Network Intrusion Detection System (IDS). *International journal of information security*, 22(5):1125–1162.

Admass, W. S., Munaye, Y. Y., and Diro, A. A. (2024). Cyber security: State of the art, challenges and future directions. *Cyber Security and Applications*, 2:100031.

Berbecaru, D. G. and Petraglia, G. (2023). TLS-Monitor: A Monitor for TLS Attacks. In *2023 IEEE 20th Consumer Communications & Networking Conference (CCNC)*, pages 1–6.

Dina, A. S. and Manivannan, D. (2021). Intrusion detection based on Machine Learning techniques in computer networks. *Internet of Things*, 16:100462.

Gite, P., Chouhan, K., Murali Krishna, K., Kumar Nayak, C., Soni, M., and Shrivastava, A. (2023). ML Based Intrusion Detection Scheme for various types of attacks in a WSN using C4.5 and CART classifiers. *Materials Today: Proceedings*, 80:3769–3776.

Gomez, G., Kotzias, P., Dell’Amico, M., Bilge, L., and Caballero, J. (2023). Unsupervised Detection and Clustering of Malicious TLS Flows. *Security and Communication Networks*, 2023(1):3676692.

Kondaiah, C., Pais, A. R., and Rao, R. S. (2024). Enhanced Malicious Traffic Detection in Encrypted Communication Using TLS Features and A Multi-Class Classifier Ensemble. *Journal of Network and Systems Management*, 32(4):76.

Sahani, N., Zhu, R., Cho, J.-H., and Liu, C.-C. (2023). Machine Learning-based Intrusion Detection for Smart Grid Computing: A Survey. *ACM Trans. Cyber-Phys. Syst.*, 7(2).

Siyyal, S. A., Khuawar, F. Y., Saba, E., Memon, A. L., and Shaikh, M. R. (2022). Analyzing ML-Based IDS Over Real-Traffic. *International Journal of Innovations in Science & Technology*, 4(3):621–640.

Thakkar, A. and Lohiya, R. (2023). A Review on Challenges and Future Research Directions for Machine Learning-Based Intrusion Detection System. *Archives of Computational Methods in Engineering*, 30(7):4245–4269.

Theofanous, A., Papadogiannaki, E., Shevtsov, A., and Ioannidis, S. (2024). Fingerprinting the Shadows: Un-

- masking Malicious Servers with Machine Learning-Powered TLS Analysis. In *Proceedings of the ACM Web Conference 2024*, WWW '24, page 1933–1944, New York, NY, USA. Association for Computing Machinery.
- Ucci, D., Sobrero, F., Bisio, F., and Zorzino, M. (2021). Near-Real-Time Anomaly Detection in Encrypted Traffic using Machine Learning Techniques. In *2021 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 01–08.
- Yang, L. and Shami, A. (2022). IDS-ML: An open source code for Intrusion Detection System development using Machine Learning. *Software Impacts*, 14:100446.
- Yu, L., Tao, J., Xu, Y., Sun, W., and Wang, Z. (2024). Tls fingerprint for encrypted malicious traffic detection with attributed graph kernel. *Computer Networks*, 247:110475.
- Zang, X., Wang, T., Zhang, X., Gong, J., Gao, P., and Zhang, G. (2024). Encrypted Malicious Traffic Detection Based on Natural Language Processing and Deep Learning. *Computer Networks*, 250:110598.
- Zeng, Z., Xun, P., Peng, W., and Zhao, B. (2024). Toward identifying malicious encrypted traffic with a causality detection system. *Journal of Information Security and Applications*, 80:103644.

