ESP-IJACT
ESP International Journal of Advancements in Computational Technology
ISSN: 2583-8628 / Volume 3 Issue 2 April 2025 / Page No: 64-74
Paper Id: IJACT-V3I2P107/Doi: 10.56472/25838628/IJACT-V3I2P107

Original Article

Machine Learning for Secure Network Traffic Analysis: From Flow Classification to Encrypted Threat Detection

Anitha Mareedu
Electrical engineering Texas A&M university - Kingsville 700 University Blvd, Kingsville, TX, USA.

Received Date: 03April 2025 Revised Date: 12 April 2025 Accepted Date: 26 April 2025

Abstract: *The growing adoption of encryption protocols such as TLS 1.3, QUIC, and DNS-over-HTTPS has limited the effectiveness of traditional deep packet inspection, challenging conventional methods of network traffic analysis. In response, machine learning (ML) has emerged as a powerful alternative, enabling the analysis of encrypted and obfuscated traffic through side-channel features, flow metadata, and behavioral patterns. This review systematically examines the evolution of ML-based techniques for secure network traffic analysis, covering supervised flow classification, anomaly detection, and encrypted threat inference. We analyze key components such as feature extraction strategies, learning models, and benchmark datasets, and assess the effectiveness of ML-powered network intrusion detection systems (NIDS) in operational settings. Tools like Zeek, CICFlowMeter, and Suricata extensions are discussed in the context of practical deployment. Furthermore, the review addresses emerging challenges including data privacy, adversarial robustness, and model explainability. We conclude by identifying open research directions focused on integrating ML with threat intelligence, enhancing interpretability, and enabling scalable, privacy-preserving detection in modern enterprise environments.*

Keywords: *Machine Learning (ML), Encrypted Traffic Analysis, Intrusion Detection (NIDS), TLS 1.3, QUIC, Federated Learning, Explainable AI (XAI).*

I. INTRODUCTION

In today's hyperconnected digital environment, network traffic analysis has emerged as a cornerstone of modern cybersecurity [1] [2]. The increasing number of cyberattacks and their complexity demand that defenders should employ advanced methods to identify, categorize, and react to the malicious attempts moving across enterprise and cloud networks. In the conventional sense, Deep Packet Inspection (DPI) was used to offer a high degree of insight into a network payload; most importantly in identifying threats of malware, data or command-and-control (C2) communications[3]. The growing trend of handling encrypted traffic has, however, seriously reduced the effectiveness of the conventional DPI-based solutions [4].

The most prominent change in the last 10 years was the massive shift of institutions to Transport Layer Security (TLS) and the various protocols of encrypting information during transit. This is evidenced by the fact that the amount of plaintext traffic has continued to decline in comparison to the encrypted traffic which has occupied most of the traffic occurring all over the world as is depicted in Figure 1. According to Google Transparency reports and Cisco's Annual Internet Reports, most web traffic is now encrypted, rendering conventional payload inspection methods largely obsolete.

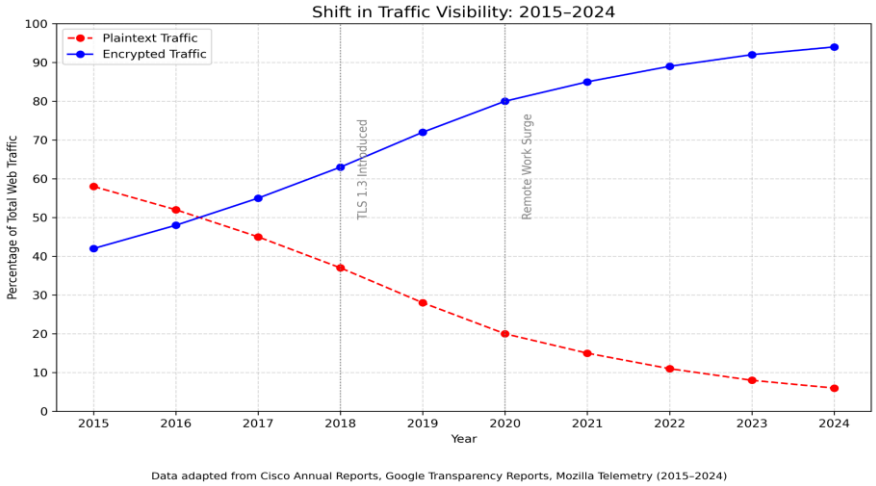

Figure 1: Shift in Traffic Visibility: Plaintext vs. Encrypted Traffic Volume

This growing encryption trend presents new challenges for network defenders[5]. Attackers now commonly embed malicious activity within encrypted sessions, relying on the fact that traditional detection tools cannot access the payload[6]. As a result, there is a growing shift towards machine learning (ML)-based techniques that analyze side-channel features such as packet timing, size, sequence patterns, and statistical flow metadata. ML models have proven capable of inferring application types, user behavior, and even potential anomalies without decrypting the traffic. Table 1 below outlines some of the key challenges faced by security analysts and automated systems in the modern traffic analysis landscape.

Table 1: Overview of Key Challenges in Modern Network Traffic Analysis

Challenge	Description
Encrypted Payloads	Limits deep inspection; requires inference from metadata or side-channels
Evasive Traffic Patterns	Adversaries mimic benign behavior or fragment flows
Volume and Velocity of Traffic	High throughput environments overwhelm signature-based systems
Data Labeling for ML Training	Scarcity of labeled malicious flow data hampers supervised learning
Privacy-Preserving Analysis	Balancing detection effectiveness with user data confidentiality

In this context, machine learning has become essential for achieving scalable, adaptive, and protocol-agnostic detection. Unlike signature-based detection, ML systems can generalize from observed traffic patterns and detect previously unseen attack vectors [7] . Furthermore, unsupervised and semi-supervised learning approaches are increasingly used to address data sparsity and class imbalance critical issues in real-world network environments. This review aims to explore the evolution of ML techniques applied to network traffic analysis, covering developments from basic flow classification to advanced encrypted threat detection. We survey key approaches, categorize them based on learning strategies and feature types, and analyze their strengths and limitations. The structure of the review is as follows: Section 2 provides a taxonomy of ML methods used in traffic analysis; Section 3 delves into flow-based classification techniques; Section 4 focuses on ML for encrypted traffic detection; Section 5 presents evaluation metrics and benchmark datasets; and Section 6 discusses limitations, open challenges, and future directions.By charting the progress of ML-powered traffic analysis over the past decade, this review highlights both its transformative impact and the critical hurdles that remain as network security enters a post-DPI era.

II. FOUNDATIONS OF ML-BASED NETWORK TRAFFIC ANALYSIS

The application of machine learning to network traffic analysis relies heavily on the nature of the input data, the formulation of learning tasks, and the design of suitable features. Before diving into detection strategies, it is essential to understand the foundational elements that underpin ML-based network traffic analysis systems.

A. Types of Network Data

Network traffic data can be captured and analyzed at different levels of granularity [8]:

- Packet-level data captures individual packets, including headers and payloads (if unencrypted). It offers fine-grained detail but is high in volume and complexity.
- Flow-level data, such as NetFlow or IPFIX, aggregates packets into sessions based on source/destination IP, port, and protocol. While coarser, it is more scalable for real-time monitoring.
- Metadata, such as TLS handshake fields, DNS queries, or protocol usage, offers privacy-preserving insights and is particularly useful for encrypted traffic.

The choice of data source directly influences model design, feature engineering, and system performance.

B. Common Machine Learning Tasks

ML techniques are typically applied to one or more of the following [9] core tasks in network traffic analysis:

- Classification: Assigning traffic to predefined categories, such as applications (e.g., Skype vs. HTTP) or benign/malicious flows.
- Clustering: Grouping traffic patterns without labels to detect unknown services or anomalies.
- Anomaly Detection: Identifying patterns that significantly deviate from normal behavior, often using unsupervised or semi-supervised approaches[10].

Supervised classification dominates academic literature, but anomaly detection is increasingly important in high-security or zero-trust contexts.

C. Feature Extraction Techniques

Feature engineering plays a critical role in the success of ML models. Approaches can be broadly categorized into:

- Manual Feature Extraction: Involves designing statistical and time-based metrics such as packet count, flow duration, inter-arrival times, and byte ratios. Tools like CICFlowMeter have standardized this process.

- Automated Feature Learning: Deep learning models, particularly Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), have been used to learn representations directly from raw byte streams or flow sequences (e.g., Deep Packet, DeepFlow).

While automated approaches reduce the need for expert-crafted features, they often require substantial training data and computational resources.

D. Overview of Benchmark Datasets

The development and evaluation of ML-based traffic classifiers rely on publicly available datasets. Table 2 lists widely-used datasets, highlighting their relevance, scale, and encryption support.

Table 2: Benchmark Datasets for Network Traffic ML

Year	Size	Data Type	Encryption Support	Notes
2017	~80 GB	Packet/Flow	Partial (VPN)	Widely used for intrusion detection
2016	~12 GB	Packet	Yes	Focused on VPN and Tor traffic
2016	~8 GB	Packet	Yes	Encrypted traffic classification
2020	~50 GB	Flow	Yes	Includes botnet C2 traffic over TLS
2020	Variable	Flow	Mixed	Emphasizes IoT traffic scenarios
2013	~25 GB	Flow	No	Classic malware traffic dataset

These datasets vary in format and scope, with more recent corpora (e.g., CIC-Darknet2020) focusing on encrypted, evasive threats. However, a significant challenge remains in obtaining labeled datasets that represent real-world encrypted traffic with sufficient diversity and up-to-date threats.

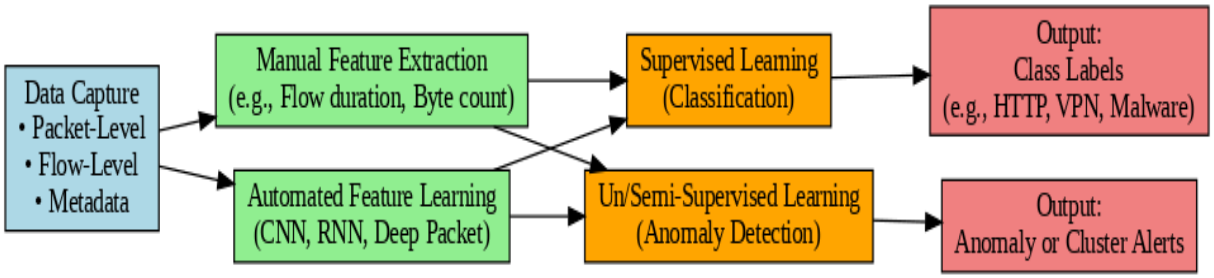

Figure 2: ML Pipeline for Traffic Analysis

III. Flow Classification and Application Identification

One of the earliest and most widespread applications of machine learning in network traffic analysis is flow classification [11], the task of assigning traffic flows to specific applications or protocols based on observable characteristics. Accurate flow classification enables not only traffic engineering and QoS management, but also foundational threat detection and policy enforcement in security operations. With the rise of encryption and obfuscation, traditional port-based or DPI-based classification methods have become less reliable[12]. ML offers a robust alternative by learning statistical and behavioral patterns in network traffic, even when the payload is encrypted.

A. Supervised Learning for Flow Classification

Supervised learning remains the dominant paradigm for flow classification due to the availability of labeled datasets and its effectiveness in controlled environments [13]. These models learn mappings from a set of engineered features such as packet counts, duration, inter-arrival times, and byte volumes to known protocol or application classes.

a) The Five-Tuple Foundation

At the heart of flow classification lies the five-tuple representation of traffic: source IP, destination IP, source port, destination port, and transport protocol [14]. These identifiers are used to group packets into sessions, from which temporal and statistical features can be extracted. As illustrated in Figure 3, these features are then fed into an ML model to output a predicted class.

B. Protocol and Application Inference

The task of flow classification is frequently to determine applications (e.g. Skype or Netflix) or protocol types (e.g. HTTP or VoIP) without examining payload contents. This is especially important in networks where a high percentage of traffic is using TLS or VPN, and deep packet inspection cannot be used.

a) Management of Encrypted and Evasive Traffic

Identification is compounded by protocols such as QUIC, DNS-over-HTTPS (DoH) and VPN tunnel. Being able to learn recognizable flow properties of such traffic, ML models have demonstrated potential in profiling its behavior based on session timing and packet size distributions and connection characteristics without requiring the traffic to be decrypted.

C. ML Models employed in Flow Classification

There is a broad range of machine learning algorithms that have been used to address flow classification: each of them has alternative trade-offs in terms of interpretability, accuracy, and scalability.

a) Classical Algorithms

RFs and SVMs were always preferred because of the property of interpretability and moderate computational needs. Specifically, RF does well in data sets such as CICIDS2017 that have the high dimension manual features.

b) Deep Learning Models

Convolutional Neural Networks (CNNs) have been modified to treat packet streams as images or time-sequences matrices [15], to extract spatial patterns in flow properties. A type of RNN, Long Short-Term Memory (LSTM) networks, have demonstrated their capability to model sequential dependencies in packets suited to rhythmic or periodic traffic. The tabular representation 3 below presents summary of the representative studies and models applied to flow classification.

Table 3: ML Algorithms Used for Flow Classification

Algorithm	Dataset	Notes
Random Forest	CICIDS2017	Manual features; good performance, fast
SVM	ISCXVPN2016	VPN detection; struggles with scalability
CNN	USTC-TFC2016	Learns from byte patterns in packet streams
LSTM	DeepFlow (Custom)	Effective for temporal flow modeling
Hybrid CNN-LSTM	CIC-Darknet2020	Combines spatial and temporal dependencies

D. Performance Considerations and Limitations

While many ML models demonstrate high accuracy in laboratory settings, their performance can degrade significantly in real-world environments due to factors like traffic heterogeneity, class imbalance, and concept drift. Furthermore, encrypted traffic generated by adversarial tools (e.g., Cobalt Strike over HTTPS) can mimic benign application flows, leading to false negatives [16]. Efforts to mitigate these issues include ensemble learning, domain adaptation, and transfer learning, which aim to improve generalization across datasets and evolving threat profiles. However, these approaches remain active areas of research, and standardized evaluation frameworks

IV. ANOMALY AND INTRUSION DETECTION

As encryption and protocol obfuscation render traditional signature-based Network Intrusion Detection Systems (NIDS) less effective, anomaly-based detection, particularly using machine learning (ML) has gained momentum. ML methods can uncover subtle, previously unseen threats such as malware command-and-control (C2) traffic, low-rate DDoS attacks, and stealthy port scans, all without requiring decrypted payloads or fixed rule signatures. Unlike supervised classification models, which require extensive labeled datasets, anomaly detection focuses on identifying deviations from normal network behavior, making it well-suited for zero-day threats and evolving attack patterns.

A. Unsupervised and Semi-Supervised Learning in NIDS

a) Unsupervised Learning Approaches

Unsupervised models assume no prior knowledge of malicious activity. These models build profiles of normal traffic based on statistical, temporal, or spatial characteristics and flag outliers as potential threats. Techniques used include:

- Autoencoders: Neural networks trained to reconstruct benign traffic features; anomalies yield higher reconstruction errors.
- Clustering (e.g., k-Means, DBSCAN): Groups similar flows together; isolated or loosely connected points are flagged as suspicious.
- Principal Component Analysis (PCA): Reduces dimensionality and isolates traffic that diverges from principal components.

b) Semi-Supervised Learning

Semi-supervised methods blend limited labeled data with large amounts of unlabeled traffic. This hybrid approach improves generalization while reducing annotation costs. One popular strategy is one-class classification, where the model learns only from benign examples and treats any deviation as malicious.

B. Detection of Specific Attack Patterns

a) Malware Command and Control (C2) Channels

C2 traffic is often encrypted and low-volume to avoid detection. ML models have been used to detect these by identifying repetitive patterns in outbound sessions, beaconing intervals, or unusual TLS handshake attributes.

b) Distributed Denial of Service (DDoS)

ML is especially effective in early detection of volumetric and application-layer DDoS attacks. Features such as packet rate, flow diversity, and SYN flag counts are commonly used. Ensemble models and deep learning have been adopted in scenarios requiring high throughput and low false-positive rates.

c) Port Scanning and Probing

Unusual connection attempts to multiple ports or hosts can signal reconnaissance activity. ML models use behavioral indicators such as connection attempt frequency, SYN/ACK ratios, and destination entropy to identify scans, including stealth variants.

C. ML-Based NIDS

Over the last decade, numerous ML-powered NIDS have been proposed and implemented. A few notable systems are:

- Kitsune (2018): A lightweight online autoencoder-based NIDS designed for IoT networks, capable of learning in real time [17].
- DeepPacket (2019): A deep learning model (CNN + BiLSTM) for encrypted packet classification and intrusion detection.
- DeepIDS (2021-2022): Deep learning-based network intrusion intrusion detection measurement of advanced persistent threats (APTs) using stacked autoencoders to extract features of raw flow data.

D. Comparative comparison of ML-NIDS

Although every system has distinct prerogatives when it comes to precision, flexibility, and the complexity of deployment, there is no single-solution. Table 4 gives the comparison of leading ML-based NIDS solutions to 2024 by the basic metrics.

E. Current Limitations

Although tremendous progress has been made around it, ML-based NIDS remain to be problematic in terms of false negatives, concept drift, and generalization over datasets, and evasion attacks. In addition, most of the proposed models test only on synthetic datasets and their capability of performance in the real world is low. Continuous learning, federated model training, and explainable AI (XAI) are some of the efforts that are being tried to fill this gap. By the end of 2024, incorporating such models into the production environments is an intricate but swiftly changing domain.

V. TRAFFIC ENCRYPTION ANALYSIS

Using TLS 1.3, QUIC, and DNS-over-HTTPS (DoH) are Internet-wide encryption protocols that have had an enormous effect on network traffic. At the same time, payload inspection, application recognition and malicious activity detection based on conventional tools such as DPI are hampered by the layer and payload encryption, which add privacy and confidentiality capabilities.

A. Impact of Modern Encryption Protocols

a) TLS 1.3 and QUIC

With the release of TLS 1.3 in 2018 and its growing adoption, critical metadata such as certificate details, cipher cipher suites, and server names (SNI) are often encrypted or removed from the handshake. Similarly, QUIC, which runs over UDP and incorporates encryption into its transport layer, obscures much of the traffic previously visible through TCP-based inspection [18].

b) DNS-over-HTTPS (DoH)

Introduced to encrypt DNS queries over HTTPS, DoH conceals domain resolution activities from both ISPs and local network monitors, disrupting visibility into user intent and blocking strategies based on domain name analysis[19]. These shifts have rendered DPI largely ineffective for encrypted sessions, especially in detecting malware command-and-control (C2) channels, VPN tunnels, and covert HTTPS-based exfiltration.

In response, researchers and practitioners have increasingly promised to use machine learning (ML) to examine encrypted traffic via side-channel characteristics; visible features of traffic flows against which the encrypted payload cannot be hidden.

Table 4: Comparison of ML-Based NIDS Techniques

System	Year	ML Approach	Deployment Target	Strengths	Limitations
Kitsune	2018	Online Autoencoder	IoT/Edge Devices	Real-time detection, lightweight	Limited depth in complex attacks
DeepPacket	2019	CNN + BiLSTM	Enterprise Network	Handles encrypted traffic	Requires GPU for training
DeepIDS	2021	Stacked Autoencoders	General NIDS	Detects APT-like behaviors	Needs high-quality normal data
RCNIDS	2023	Recurrent CNN	Cloud Environments	Effective on bursty traffic	Longer training and inference times
FlowGuard	2024	Hybrid GNN + SVM	SDN/5G Networks	Context-aware detection	Early-stage research, not scalable

B. ML for Encrypted Traffic Classification

To regain visibility, ML models analyze encrypted flows using **non-payload features** such as packet size distributions, flow lengths, and timing sequences [20]. These models can infer application types, tunnel usage, and potentially malicious activity, all without decrypting the payload.

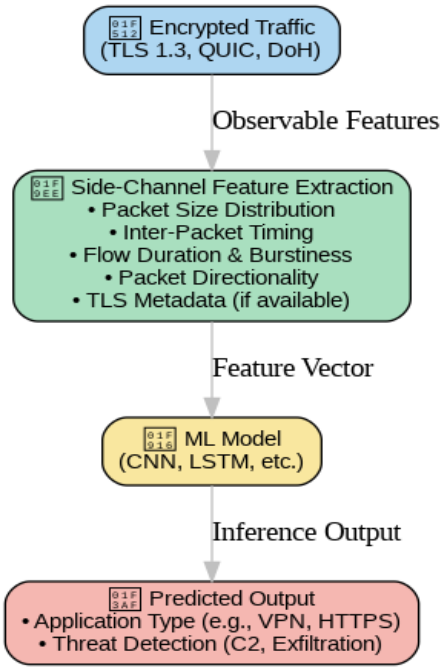

Figure 5: Feature Abstraction from Encrypted Flows

C. Side-Channel Feature Engineering

The most informative and commonly used side-channel features for ML-based encrypted traffic analysis include [21]:

- Packet size distributions: Mean, variance, and histograms of packet sizes.
- Inter-packet timing: Time gaps between consecutive packets.
- Flow duration and burstiness: Duration of flows and concentration of data bursts.
- Packet directionality: Ratios of incoming to outgoing packets.
- TLS handshake metadata (when available): Protocol versions, extension fields, etc.

Deep learning methods, especially CNNs and LSTMs have demonstrated effectiveness in automatically learning these patterns from time-series flow data [22].

D. Use Cases and Examples

a) VPN and Tunneling Detection

ML has been widely applied to VPN detection, even when traffic is encrypted end-to-end. For instance, DeepVPN (2020) and similar models apply CNNs to TLS metadata and flow shape to distinguish between VPN providers such as OpenVPN, WireGuard, and commercial VPNs like NordVPN.

b) HTTPS-Based Threat Detection

Encrypted tunnels using standard ports (e.g., 443) are often abused for malware communication and data exfiltration. ML-based approaches detect such misuse by identifying abnormal patterns in session lengths, server response behavior, and periodicity in outbound traffic. Studies published (e.g., "FlowPrint", "SPLAT", and "TLS-Fingerprinting") have demonstrated >95% accuracy in distinguishing between benign and malicious HTTPS tunnels.

E. Limitations and Emerging Challenges

Despite promising accuracy, many encrypted traffic classifiers face real-world deployment challenges:

- Generalization across networks with different traffic characteristics.
- Adversarial evasion through padding, flow shaping, or mimicry.
- Lack of labeled encrypted datasets, especially for VPN or malware traffic.

Recent research explores federated learning, online adaptation, and explainable AI to improve robustness, interpretability, and real-time applicability in encrypted environments.

VI. PRIVACY-PRESERVING AND ADVERSARIAL CHALLENGES

As machine learning becomes integral to network traffic analysis, it introduces new risks related to data privacy, model robustness, and adversarial manipulation. Sensitive network flows could be trained on models that may leak sensitive information about one or more users; also, a model trained on sensitive network traffic patterns can be easily compromised by an attacker using an evasion attack. To make deployment of ML-powered network security systems trustworthy, it is important to deal with these problems.

A. Training Risk on Traffic Sensitive Data

Personally identifiable information (PII), behavioral patterns, enterprise service metadata, and DNS queries (and even in flow-level form) are quite common in network traffic. In the case of training with the improper use of such data[23]:

- There is a possibility of models learning to memorize patterns that can be read by their attacker in terms of individual session or device information via modeling inversion or reconstruction.
- Centralized data collection from user endpoints or edge devices raises compliance concerns with regulations like GDPR and HIPAA.

These concerns have led to growing interest in privacy-preserving learning frameworks, particularly in security-sensitive and regulated environments.

B. Federated Learning and Differential Privacy

a) Federated Learning (FL)

Federated learning enables ML models to be trained locally at endpoints or edge devices, with only the updated model parameters shared with a central server. This technique eliminates the need to transmit raw traffic data to a central repository[24].

- In the context of traffic analysis, FL has been explored in projects like FedPacket and FedNIDS, which train flow classifiers across distributed environments without exposing local datasets.
- Challenges remain in model drift, non-IID data, and synchronization overhead, especially in real-time detection systems.

b) Differential Privacy (DP)

Differential privacy introduces statistical noise into either the input data or model parameters to ensure that individual contributions remain untraceable.

- Applied to NIDS systems, DP protects user-level patterns in training data.
- However, adding noise can reduce detection sensitivity, especially for rare or subtle anomalies like beaconing C2 traffic.

Research in DP-compliant autoencoders and LSTM models for encrypted traffic classification is ongoing, with trade-offs between utility and privacy still being evaluated.

c) Adversarial Examples in Traffic Classification

- ML models are inherently vulnerable to adversarial examples input data that has been subtly manipulated to mislead the classifier. In the context of traffic analysis:
- Attackers can shape packet timings, add dummy packets, or pad flow lengths to evade detection.

For instance, adversaries have used mimicry techniques to make malware flows resemble Netflix or HTTPS sessions, bypassing ML-based detectors. Studies such as AdvNet and NetFool demonstrated the feasibility of generating adversarial flows capable of deceiving both classical and deep models without altering the core malicious functionality.

d) Robustness Testing of Models

Given the growing threat of adversarial evasion, robustness testing is becoming a necessary step in the ML pipeline for NIDS. Key techniques include:

- Adversarial training: Incorporating adversarial flows during model training to improve resilience.
- Gradient masking: Obfuscating decision boundaries to prevent precise gradient-based attacks.
- Black-box evaluation: Testing detection models against unknown and adaptive threats without direct access to model internals.

Despite these efforts, no standard benchmarking framework exists for evaluating the robustness of ML-based traffic analysis models under real-world adversarial conditions. This remains a pressing research gap in the domain.

VII. TOOLS, FRAMEWORKS, AND REAL-WORLD DEPLOYMENT

While machine learning has demonstrated strong potential in network traffic analysis within academic literature, the transition from research to real-world deployment has proven to be complex. Practical adoption depends not only on detection accuracy but also on integration feasibility, performance, interpretability, and maintainability within live enterprise environments. This section surveys prominent ML-supported frameworks, contrasts research prototypes with operational deployments, and outlines key deployment challenges.

A. Key Frameworks and Tools

A variety of open-source and research-driven frameworks have emerged to facilitate ML-based traffic analysis. These tools typically focus on feature extraction, real-time processing, or detection logic integration with existing network monitoring systems.

a) Zeek + ML Integration

Zeek (formerly Bro) is a powerful network security monitor that has gained popularity for its extensibility [25]. While Zeek itself does not natively support machine learning, several research efforts and third-party extensions have enabled ML workflows:

Feature exports from Zeek logs (e.g., conn.log, ssl.log) serve as structured input to external classifiers. Projects like ML-Zeek (2022) and custom Python-based pipelines integrate pre-trained models with Zeek logs to detect anomalies in TLS handshakes or behavioral flows.

b) CICFlowMeter

Developed by the Canadian Institute for Cybersecurity, CICFlowMeter is a widely adopted tool for converting packet capture (PCAP) files into labeled flow-based datasets with over 80 statistical features per flow. It is the backbone for numerous datasets including CICIDS2017, ISCXVPN2016, and CIC-Darknet2020.

- Benefits: Compatible with Python/Scikit-learn, easy integration with ML pipelines.
- Limitations: Designed for offline analysis, less suitable for real-time environments.

c) Suricata ML Extensions

Suricata, an open-source IDS/IPS engine, introduced support for EVE JSON logs and metadata tagging, making it feasible to couple with ML inference systems. Research prototypes (e.g., Suricata-ML Bridge) demonstrated flow-based detection using external classifiers. Still in experimental stages; lacks native ML module.

B. Research Prototypes vs. Commercial Integrations

While academia has produced a wide array of promising models, most commercial network detection and response (NDR) systems still rely heavily on heuristic, signature, or behavioral baselining methods. ML is used in the background, often with proprietary datasets and little transparency.

a) Research Prototypes

- Typically built on synthetic or open datasets.
- Favor accuracy over operational constraints like latency or memory usage.
- Emphasize innovation (e.g., adversarial detection, encrypted traffic classification).

b) Commercial Implementations

Some commercial security solutions began integrating ML, including:

- Darktrace: Uses self-learning ML to model network behavior and detect anomalies.
- Vectra AI: Leverages flow metadata and deep learning to identify lateral movement and encrypted threat activity.
- Cisco Secure Network Analytics (formerly Stealthwatch): Introduced ML-based anomaly detection modules for encrypted traffic.

However, full ML-based detection pipelines remain rare in enterprise deployments due to operational risks.

C. Barriers to Real-World Deployment

Despite technical maturity in experimental settings, several barriers still prevent widespread deployment of ML-based network detection models:

- Performance Constraints: Many deep learning models require GPUs or extensive CPU resources, limiting scalability in high-throughput environments.
- Lack of Explainability: Security analysts are reluctant to trust "black box" models, especially in regulated sectors like healthcare or finance.
- Data Labeling and Drift: Realistic training data is scarce, and models degrade over time due to concept drift.
- Integration Overhead: Linking ML workflows to existing SIEMs, firewalls, and data lakes introduces additional operational complexity.

Table 7: ML Frameworks for Network Traffic Analysis

Framework	Type	Strengths	Limitations
Zeek + ML	Log-based	Highly extensible, used in SOC workflows	Requires external ML integration
CICFlowMeter	Flow converter	Rich feature set, widely adopted in research	Not real-time, mostly offline processing
Suricata-ML Bridge	IDS extension	Real-time capability, open-source foundation	Still experimental, lacks native ML module
ML-NIDS Prototypes	Academic systems	High detection accuracy on testbeds	Rarely production-ready, limited interpretability
Darktrace / Vectra AI	Commercial	Scalable, anomaly detection over encrypted flows	Proprietary models, limited transparency

VIII. FUTURE RESEARCH DIRECTIONS

As machine learning continues to reshape network traffic analysis, several promising research directions are emerging—particularly in response to growing encryption and adversarial sophistication. One major opportunity lies in developing context-aware models capable of correlating encrypted traffic patterns with behavioral baselines, device profiles, and endpoint signals to infer intent without payload access. Additionally, integrating ML with threat intelligence feeds and graph-based analytics (e.g., using communication graphs or flow graphs) could enable systems to detect lateral movement, coordinated attacks, or multi-stage intrusions that are difficult to capture using isolated flow analysis. Another pressing need is for interpretable and auditable ML models that can justify detections to human analysts, satisfy compliance requirements, and facilitate forensic investigations, especially in sectors such as healthcare, critical infrastructure, and finance.

Techniques like attention mechanisms, rule extraction, and explainable AI (XAI) frameworks are being explored to address this. Lastly, for ML-based NIDS to achieve real-world impact, future research must emphasize scalable and realistic deployment models within enterprise Security Operations Centers (SOCs). This involves working on resource-efficient models, modular architectures which interface well with SIEMs and SOAR systems, and strong concept-drift and moving threats mechanisms. The challenge that is most serious and interesting in this area is how to fill the gap between the innovation and the application of the research work.

IX. CONCLUSION

Machine learning has recently become an emerging core technology in a new approach to secure network traffic analysis, providing more powerful techniques of visibility, detection and automated responding to threats in a world where in far too many cases it is clear that computing really is encryption marked and money talked when it comes to getting direct access to traffic at the interface level. The review described the evolvement of ML methods, where the initial ML methods were classical supervised classifiers used to solve the task of flow identification, to most recent advanced deep learning models that can deal with encrypted patterns and discover hidden threats. We discussed vital aspects of the ML pipeline such as feature engineering, benchmarking datasets, model architectures, and evaluation methods. Our academic and practice survey showed us encouraging new developments and existing gaps.

While research prototypes have demonstrated high accuracy in identifying application types, VPN usage, and anomalous behaviors, real-world deployment still faces challenges related to computational overhead, lack of interpretability, model brittleness under adversarial conditions, and limited access to diverse, up-to-date datasets. Moreover, the growing use of privacy-preserving protocols like TLS 1.3 and QUIC, as well as adversarial techniques like flow mimicry and traffic shaping, underscore the need for more robust, context-aware, and adaptive detection systems. The integration of ML with

emerging technologies such as graph analytics, federated learning, and explainable AI offers promising pathways to address these limitations.

However, for machine learning to become truly operational in enterprise Security Operations Centers (SOCs), it must go beyond accuracy metrics and address real-world constraints such as alert fatigue, system interoperability, and regulatory compliance. In essence, while ML is no longer a speculative tool in network security, its practical adoption requires a shift from isolated model development to end-to-end, resilient, and interpretable systems. Bridging the gap between research innovation and operational deployment remains the foremost challenge—and opportunity—for the cybersecurity community in the years ahead.

X. REFERENCES

[1] A. S. George, et al., "Innovative traffic management for enhanced cybersecurity in modern network environments," *Partners Univ. Int. Res. J.*, vol. 3, no. 4, pp. 1–13, 2024.

[2] J. Jangid and S. Malhotra, "Optimizing software upgrades in optical transport networks: Challenges and best practices," *Nanotechnol. Percept.*, vol. 18, no. 2, pp. 194–206, 2022. [Online]. Available: https://nano-ntp.com/index.php/nano/article/view/5169

[3] J. Jangid, S. Dixit, S. Malhotra, M. Saqib, F. Yashu, and D. Mehta, "Enhancing security and efficiency in wireless mobile networks through blockchain," *Int. J. Intell. Syst. Appl. Eng.*, vol. 11, no. 4, pp. 958–969, 2023. [Online]. Available: https://ijisae.org/index.php/IJISAE/article/view/7309

[4] M. Çelebi, A. Özbilen, and U. Yavanoğlu, "A comprehensive survey on deep packet inspection for advanced network traffic analysis: Issues and challenges," *Niğde Ömer Halisdemir Univ. J. Eng. Sci.*, vol. 12, no. 1, pp. 1–29, 2023.

[5] Y. Zou, et al., "A survey on wireless security: Technical challenges, recent advances, and future trends," *Proc. IEEE*, vol. 104, no. 9, pp. 1727–1765, 2016.

[6] J. Jangid, "Secure microservice communication in optical networks," *J. Inf. Syst. Eng. Manage.*, vol. 10, no. 21s, 2025. doi: 10.52783/jisem.v10i21s.3455

[7] A. Bachir, et al., "A signature and NLP-based network traffic detection model for SQL injections for enhancing web security," in *Proc. 2024 IEEE/ACM Int. Conf. Big Data Comput., Appl. Technol. (BDCAT)*, IEEE, 2024.

[8] W. J. Buchanan and W. J. Buchanan, "Networking types," in *The Handbook of Data Communications and Networks: Volume 1. Volume 2*, pp. 743–769, 2004.

[9] Y. Yan, "Machine learning fundamentals," in *Machine Learning in Chemical Safety and Health: Fundamentals with Applications*, pp. 19–46, 2022.

[10] J. Jangid, "Efficient training data caching for deep learning in edge computing networks," *Int. J. Sci. Res. Comput. Sci. Eng. Inf. Technol.*, vol. 7, no. 5, pp. 337–362, 2020. doi: 10.32628/CSEIT20631113

[11] [11] T. T. T. Nguyen and G. Armitage, "A survey of techniques for internet traffic classification using machine learning," *IEEE Commun. Surv. Tutor.*, vol. 10, no. 4, pp. 56–76, 2008.

[12] A. Ghosh and A. Senthilrajan, "Classifying network traffic using DPI and DFI," *Int. J. Sci. Technol. Res.*, vol. 8, no. 11, pp. 1019, 2019.

[13] M. M. Raikar, et al., "Data traffic classification in software defined networks (SDN) using supervised-learning," *Procedia Comput. Sci.*, vol. 171, pp. 2750–2759, 2020.

[14] A. N. Mahmood, et al., "Network traffic analysis and SCADA security," in *Handbook of Information and Communication Security*, pp. 383–405, 2010.

[15] H. Whitworth, et al., "5G aviation networks using novel AI approach for DDoS detection," *IEEE Access*, vol. 11, pp. 77518–77542, 2023.

[16] M. Abolfathi, *Enhancing Encrypted Network Traffic Security Against Advanced Traffic Analysis Attacks*, Ph.D. dissertation, Univ. of Colorado at Denver, 2024.

[17] G. Skibyak, *Kitsune: A Look into the Lasting Presence of the Fox Spirit in Japanese Culture*, M.S. thesis, New Mexico State Univ., 2020.

[18] T. Jager, J. Schwenk, and J. Somorovsky, "On the security of TLS 1.3 and QUIC against weaknesses in PKCS#1 v1.5 encryption," in *Proc. 22nd ACM SIGSAC Conf. Comput. Commun. Security*, 2015.

[19] T. Böttger, et al., "An empirical study of the cost of DNS-over-HTTPS," in *Proc. Internet Meas. Conf.*, 2019.

[20] T. Furlong, *Tools, Data, and Flow Attributes for Understanding Network Traffic Without Payload*, Ph.D. dissertation, Carleton Univ., 2007.

[21] E. Cagli, *Feature Extraction for Side-Channel Attacks*, Ph.D. dissertation, Sorbonne Univ., 2018.

[22] K. Bhargavan, V. Cheval, and C. Wood, *Handshake Privacy for TLS 1.3—Technical Report*, Diss. Inria Paris; Cloudflare, 2022.

[23] J. Zhou, et al., "Challenges and advances in analyzing TLS 1.3-encrypted traffic: A comprehensive survey," *Electronics*, vol. 13, no. 20, p. 4000, 2024.

[24] X. Wu, et al., "An adaptive federated learning scheme with differential privacy preserving," *Future Gener. Comput. Syst.*, vol. 127, pp. 362–372, 2022.

[25] V. Gustavsson, *Machine Learning for a Network-Based Intrusion Detection System: An Application Using Zeek and the CICIDS2017 Dataset*, 2019.