Network Traffic Analysis for Detecting Cyber Attacks (October 2026)

Praveen Kumar Mishra[1], Aastha Singh[2]
[1]*School of Computer Application, Babu Banarsi Das University Lucknow*
[2]*Assistant Professor, Babu Banarsi Das University Lucknow*
doi.org/10.64643/IJIRT209606

***Abstract*—Most attacks have to traverse a network at some point, and when they do so, they leave traces in the traffic. This paper investigates how those traces can be collected, measured and turned into alerts. We cover the most common ways to collect traffic data (packet captures, flow records and logs), the types of characteristics that are generally useful, and three main families of detection methods: signature-based, statistical and machine learning-based. We also describe how several well-known attacks (port scans, denial of service, brute force and botnet command-and-control) manifest themselves on the wire A step-by-step analysis pipeline is presented, as well as the most common public datasets used for evaluation and the corresponding metrics. Finally, we discuss the major challenges that remain in this area: encrypted traffic, class imbalance, concept drift and the gap between laboratory test results and real-world network deployments. The goal is to provide a clear, practical overview for students and junior security analysts rather than presenting a novel algorithm.**

***Index Terms*—network traffic analysis, intrusion detection, anomaly detection, NetFlow, machine learning, DDoS, encrypted traffic.**

I. INTRODUCTION

Every organization that uses the Internet is subjected to scanning attempts frequently. A public server can be scanned shortly after it is brought online. A large percentage of these scans are harmless research crawlers. Endpoint tools such as antivirus software and host logs are useful, but they only see what happens on one machine, and an attacker who has already gained control of that machine can tamper with them. The network is different. Traffic between machines has to travel through switches, routers and gateways, so a monitoring point placed at the right spot sees activity from many hosts at once

On the other hand, the network is different. Traffic traveling between machines must go through switches, routers and gateways, therefore, a monitoring station positioned correctly can observe traffic from multiple machines simultaneously. It is this broader perspective that makes network traffic analysis (NTA) one of the primary components of intrusion detection even though an increasing amount of traffic is becoming encrypted. The basic concept behind NTA is straightforward. Normal traffic exhibits patterns: specific hosts communicate with specific servers during specific times and usually have predictable packet sizes and connection counts. Attacks disrupt these patterns or demonstrate patterns that normal traffic does not exhibit.

For example, a port scan will touch hundreds of ports within seconds. A flood will send significantly more packets than any normal user would. A compromised machine will contact an unknown server every minute like clockwork. If we can measure the patterns of normal traffic, we can detect deviations from those patterns.

II. BACKGROUND AND RELATED WORK

While the concept appears simple in theory, implementing it in practice is considerably more difficult due to several factors. Networks generate significant amounts of noise; traffic volumes are enormous and what constitutes normal varies greatly from one organization to another. As Sommer and Paxson pointed out over a decade ago that anomaly detection in networks is substantially different from image recognition because errors are costly and the data is constantly evolving [1]. That observation remains valid today. This paper has three objectives. Firstly, to define what data is utilized by network traffic analysis. Secondly, to compare the major types

of detection methods and objectively identify the limitations of each method. Thirdly, to outline a practical pipeline that could be developed and tested by a small team.

The remaining sections of the paper follow that order. 2. Background and Related Work Intrusion detection systems have been studied since the late 1980s, and the first practical open-source tools emerged during the late 1990s. Snort used pattern matching based on packet payload and header information and became the reference model for signature-based systems [2]. Bro (later renamed Zeek), on the other hand, utilized raw packets to generate a series of high-level events (e.g., a connection, an HTTP request, a DNS lookup) and allowed analysts to write scripts based upon those events [3]. Both ideas are still widely deployed today. Signature-based systems offer speed and ease of explanation to managers/auditors; however, they are only capable of identifying threats that have previously been documented. Researchers turned towards anomaly detection for identifying new/modified attacks, where a model of normal behavior is developed and deviations from that model are flagged. Garcia-Teodoro et al., provided a review of anomaly-based detection techniques and observed early on that high false alarm rates present significant barriers to widespread adoption [4]. The 2010s witnessed a surge in research employing machine learning.

Buczak and Guven surveyed data mining and learning methods applied to intrusion detection and compared how each family performed and what data it needed [5]. More recently, deep learning has been tried on both raw packets and flow features. Ferrag et al. compared a number of such approaches on several public datasets [6], and Mirsky et al. showed that a lightweight ensemble of autoencoders could run online, packet by packet, on modest hardware [7].

Quietly researching datasets, themselves has occurred as well. The original KDD Cup 99 dataset was extensively employed for many years despite containing numerous duplicate records that artificially inflated reported accuracy [8]. More recent collections attempt to be more realistic; however, they are still imperfect; one detailed examination of CICIDS2017 revealed labeling inaccuracies & tool-related artifacts that may mislead classifiers [9]. Anyone reviewing headline accuracy values in this area should bear this in mind.

III. NETWORK TRAFFIC DATA

Prior to selecting an algorithm for network traffic analysis, it is beneficial to specify precisely what will be analyzed. Network traffic can be captured/analyzed at three different levels of granularity/richness-vs-cost. From these sources analysts develop features. Some features are simply counts such as packets/second or distinct destination ports/host in one minute. Other features define timing: the delays between packets, the length of a session or how consistent the intervals between connections are. Still other features define size/direction (e.g., byte ratio sent/bytes received). Tools such as CICFlowMeter & Zeek's connection logs create dozens of such values for each flow. Two significant points should be emphasized. Firstly, window length is critical. A one-second window will identify floods but miss slow scans that intentionally send one probe every few minutes. Secondly, encryption has changed what is visible. Since most web traffic currently employs TLS payload inspection is frequently impossible; therefore, analysts rely more on metadata: packet sizes, timing, handshake fields & destination reputation. Anderson & McGrew demonstrated that such contextual flow data can differentiate malicious encrypted traffic from benign traffic with reasonable accuracy [10].

IV. DETECTION TECHNIQUES

4.1 Signature-based detection

A signature is a rule defining a known malicious pattern: byte sequence in payload; malformed header; connection to blacklisted address etc. When traffic matches a signature, an alert fires. This type of detection is inexpensive to implement/run & simple to explain to managers/auditors since alerts directly reference rules that triggered them. The obvious drawback is that an attacker who alters a single detail of an exploit or utilizes a vulnerability for which no rule exists will easily bypass signature-based detection.

4.2 Statistical and rule-of-thumb anomaly detection

The second family defines deviations from baselines established for current traffic. An example includes thresholding on connection attempts/minute from a single source. Another example includes comparing entropy: during DDoS attacks source address

distribution frequently changes abruptly as does destination port distribution during scans. These methods require minimal data/computing resources & minimal effort in establishing baselines. The limitation of these methods lies in establishing baselines: A new marketing campaign; software update; company-wide video conference call may trigger thresholds/anomalies in naive threshold-based detection methods.

4.3 Machine learning and deep learning

Supervised models (e.g., random forests; gradient-boosted trees; neural networks) learn from labeled examples of normal/attack flows & often perform extremely well on public benchmarks but only recognize attack types encountered during training. Unsupervised/semi-supervised models (e.g., clustering; one-class classifiers; autoencoders) learn only what constitutes normal traffic & flag anything unusual [7]. Theoretically unsupervised models can identify new attacks although they tend to generate more false alarms. Sequence models (LSTMs; transformers) attempt to read traffic as a conversation over time rather than treating each flow individually.

4.4 Which one should you use?

In reality most deployments employ a combination of detection methods: signature-based detection handles large volumes of known threats inexpensively; statistical layer detects floods/scans; & machine learning-based model analyzes remaining traffic. Ultimately the most important aspect of implementing network intrusion detection systems is not which algorithm serves as its core but how alerts are prioritized & responded to subsequently. An analyst receiving thousands of alerts/days will cease reviewing them [1].

V. WHAT TRAFFIC ANALYSIS ACTUALLY OBSERVES

Raw material for NTA resides at three levels of increasing detail/decreasing affordability. Flow records represent the coarsest level. A flow is defined as a collection of packets sharing common five-tuple parameters source address; destination address; source port; destination port; protocol transmitted within defined time window. Instead of storing each packet monitoring station records aggregate information start time; duration; total bytes; total packets regarding the flow when it terminates. NetFlow & IPFIX are dominant export formats; NetFlow being essentially vendor-specific but evolving into de-facto industry standard & IPFIX being formalized successor via IETF specification. The appeal is compression. A flow record is a few dozen bytes standing in for a conversation that may have carried megabytes, and routers can export flows with little performance penalty. The cost is that everything inside the conversation disappears. An analyst knows that 4.2 megabytes moved from an internal host to an external address over 90 seconds; they do not know *whether that was a software update or an exfiltration.*

Protocol metadata logs produced by Zeek provide somewhat richer detail than flow records. Zeek does not primarily perform byte-matching. Zeek reconstructs application-layer sessions & produces structured records describing them: HTTP log contains method; host; user-agent; response code; DNS log contains every query & answer; TLS log contains server name; certificate details; cipher suite; validation status etc.

Distinction is important because it determines questions that can be asked. Flow record answers "how much?"; protocol log answers "what kind of thing?". Zeek documentation explicitly states that users requiring classical signature detection should use Suricata or Snort instead & many organizations run both tools concurrently because they address different questions.

Full packet capture represents finest level of detail & only level at which payload content is available. Full packet capture is typically infeasible at enterprise scale due to storage requirements growing linearly with traffic volume & a busy 10 Gbps link can generate terabytes/day. Therefore, full packet capture is generally temporary/selective retained for window of hours/days; triggered on suspicion; limited to specific subnets etc.

Detection methods are influenced by which level of detail organization can afford: Signature systems require payload; therefore, signature systems require capture/inline inspection. Flow-based anomaly detection trades detail for coverage & can operate on exports from infrastructure not originally designed for security.

VI. FROM SIGNATURES TO BEHAVIOUR

First generation of network intrusion detection matched rules against packet content. Snort established model in 1998: rule specifies protocol; direction; pattern to search for in payload; action taken on match. Tens of thousands of such rules are distributed through open/commercial feeds & approach works effectively when attack is known. Publicly available exploit sending distinctive byte sequence will be identified reliably; quickly; & economically computationally.

Failure mode is also clearly defined: signature can only match what someone has described previously. If an attacker modifies single character in payload; adds encoding; routes attack through encrypted channel; or uses technique never made public then signature rule fails to identify attack. Attackers recognized this years ago; resulting practical implication is that signature detection is virtually blind to early stages of targeted intrusion. Additionally, signature detection is prone to false positives on legitimate traffic containing matching pattern; driving operators toward disabling noisy rules; thereby reducing coverage.

Anomaly detection reverses logic: instead of defining what is bad; anomaly detection defines what is normal & flags deviations from normal. Anomaly detection has desirable characteristic of potentially identifying attacks never observed previously precisely type of attacks that matter most. Difficulty lies in defining "normal" as statistical claim & networks are highly dynamic: traffic profiles vary with business hours; software updates; seasonal loads; organizational changes. Model trained on one month baseline will identify legitimate changes in next month as anomalous; resulting high volume alerts represent central practical complaint regarding anomaly-based systems; not optimization issue but structural issue: detector has been tasked with differentiating unusual-and-malicious from unusual-and-fine; only signal available at network layer is unusualness.

Behavioral approaches reside between signature-based & anomaly detection approaches: instead of defining normal traffic volume; behavioral approaches define expected behavior of specific entities host normally communicates with these addresses on these ports; service account normally transfers less than megabyte/day; workstation has never needed to resolve external DNS name resembling random string. Deviations from entity baselines are more interpretable than deviations from network-wide statistics; MITRE's ATT&CK framework provides common language for articulating what observed behavior may represent facilitating alert triage. Trade-off remains unchanged: more specific baselines equate to more false positives when legitimate behavior changes.

VII. MACHINE LEARNING APPROACHES

to them. More recently sequence models & attention mechanisms have been the recent literature is dominated by machine Recent literature is dominated by machine learning; honest assessment indicates field has produced many models demonstrating similar performance on similar data. Standard pipeline involves extracting numerical features from flow records duration; byte/packet counts in each direction; inter-arrival time statistics; flag counts; header-derived quantities such as initial window size & feeding them into classifier. Random forests & gradient-boosted trees remain strong baselines & are difficult to surpass using tabular data produced by flow records. Deep learning emerged due to observation that image classification architectures demonstrate good performance on network data leading to work converting flow features into matrix/image representations & applying convolutional networks applied to packet-length/timing sequences based on theory that rhythm of connection conveys information even when every byte is encrypted.

Two directions are worth separating from the general enthusiasm.

Federated learning addresses real barrier organizations are hesitant to disclose raw traffic because it contains personal data; internal addressing & potentially regulated information. Federated approaches develop shared model by aggregating parameter updates from participants who do not transmit their data. Reviews of federated learning suggest it is maturing field with ongoing security concern aggregation itself represents attack surface; participant submitting manipulated updates can affect global model. Robust aggregation

rules exist & studies have shown single adversarial client can compromise naively aggregated model requiring defense rather than optimization.

Unsupervised/semi-supervised methods align with reality regarding scarcity of labeled attack data. Autoencoders; clustering; one-class classifiers attempt to learn normal structure without labels & flag reconstruction error/distance as anomalies Unsupervised/semi-supervised methods appear appealing on paper but present challenges in production environments similar to all anomaly detection methods absence of labels during training results in absence of mechanism for optimizing what analyst wishes to identify.

Repeatedly reported finding across recent surveys merits emphasis sophisticated model architectures rarely provide substantial improvements over well-tuned classical models utilizing identical dataset. Reported improvements of several percentage points in accuracy frequently disappear when comparing models controlling for feature engineering; class balancing; preprocessing. This pattern is prevalent within applied machine learning & implies that primary constraint in network intrusion detection is not model capacity.

VIII. THE ENCRYPTION WALL

TLS 1.3 encrypted almost everything readable in handshake including server certificate allowing observers to identify destination of HTTPS connection by name. One field survived Server Name Indication extension remaining unencrypted because server requires knowledge of which certificate to present before establishing secure channel. SNI enabled middleboxes/analysts to infer hostname targeted by connection & substantial amount of enterprise security architecture content filtering; data loss prevention; malware detection was silently constructed on premise that this field would remain visible.

That premise is currently being dismantled. Encrypted Client Hello standardized by IETF as extension to TLS encrypts entire ClientHello including SNI utilizing public key published by server enabling clients to encrypt to location server can decrypt. Visible portion of handshake contracts to address of shared fronting server. For defenders this signifies those two remaining valuable identifiers for TLS connection destination address & destination name merge into one identifier & name may belong to provider fronting thousands of disparate domains.

Detection does not become impossible; it becomes Detection does not become impossible; detection becomes inferential. Research into encrypted traffic classification including studies covering QUIC; DNS-over-HTTPS & other contemporary protocols focus on techniques that indirectly recover valuable signals. Packet length/direction sequences establish fingerprint distinguishing categories of applications without decryption. Timing patterns indicate whether flow is interactive; streaming; bulk transfer. Sequence of flows endpoint initiates burst of DNS lookups; connections to set of addresses; lengthy low-volume upload represents behavioral evidence regardless of individual packet. Certificate metadata remains available in many cases even when SNI is masked; along with offered cipher suites. None of this is as explicit as reading hostname & all is susceptible to error.

Inspection middleboxes terminating TLS to review content remain viable options & are widely implemented in managed environments. Costs associated with inspection middleboxes have become increasingly difficult to justify: inspection middleboxes weaken end-to-end assurances protocol was designed to afford; disrupt certificate pinning & certain newer protocol features; establish high-value target at interception point; are entirely inapplicable to traffic originating from unmanaged devices. Certification/browser policies have also shifted away from inspection middleboxes. For an expanding proportion of traffic unable to be intercepted field's focus is toward inference from metadata rather than content.

IX. HOW THESE SYSTEMS ARE EVALUATED AND WHY IT IS A PROBLEM

Detection research relies on benchmark datasets; benchmarks utilized are becoming outdated. CICIDS2017 released by Canadian Institute for Cybersecurity continues to be most extensively utilized benchmark in literature; CICIDS2017 comprises labeled flows from five days of simulated activity on testbed containing fourteen machines with attacks including brute force; denial of service; web attacks & infiltration. NSL-KDD & predecessor KDD Cup 99 are older than CICIDS2017 & are now

primarily utilized for historical comparisons. Newer resources such as IDS2025 present balanced flow features extracted using tools like CICFlowMeter; various curated collections combine existing captures to balance encrypted malicious/benign traffic. Issues associated with this practice are well-documented & rarely acted upon.

Dataset drift. Traffic from 2017 does not resemble traffic today. Protocol mixes have evolved; encryption coverage has expanded; cloud/mobile patterns have replaced enterprise topologies prevalent during testbed era; attack families represented within dataset have largely been superseded. Model achieving near-perfect accuracy on CICIDS2017 has demonstrated ability to classify 2017 traffic not desired claim.

Testbed artefacts. Simulated captures contain regularities absent from production environments. Attack traffic generated by automated tools possesses distinctive machine-timing/packet-size distributions; benign traffic generated by scripts differs from traffic generated by human clicks. Model can learn distinction between generator/attack rather than distinction between benign/malicious behavior & will report excellent metrics while doing so.

Cross-dataset generalisation. Most detrimental finding in this literature is that models trained on one benchmark & tested on another degrade significantly. Recent study on cross-dataset generalization for machine learning intrusion detection makes point explicitly: performance appearing strong within dataset crumbles across datasets implying that learned features are specific to dataset rather than specific to attack. This undermines typical interpretation of accuracy values & implies substantial portion of reported progress is measurement artifact.

Label quality and imbalance. Labels in datasets are generated by testbed operator not independent analyst & multi-class labels in particular are noisy. Attack classes are significantly less prevalent than benign classes therefore accuracy serves as misleading summary statistic classifier predicting benign for everything achieves high accuracy on realistic distribution. Balanced versions of datasets facilitate training but eliminate imbalance making deployment challenging.

Metric selection. Precision; recall; false positive rate; cost of analyst time are metrics relevant operationally. Much of literature reports accuracy & F1 score occasionally on artificially balanced test sets. Detector achieving 99% recall/1% false positive rate applied to million daily connections yields ten thousand alerts. Metrics reported & metrics determining whether system is usable are not equivalent metrics.

X. DETECTORS AS TARGETS

A less-discussed implication of shifting detection into machine learning is that detector becomes entity attacker can manipulate. Adversarial machine learning research has established that neural classifiers can be fooled by small/deliberate perturbations to inputs a substantial body of work now applies this to network intrusion detection.

Network settings differ from image settings in ways that matter features are not pixels features must satisfy protocol constraints limiting attacker's ability to freely modify them However network settings also allow latitude image settings do not many flow features depend on attacker-controlled behavior attacker desiring to evade flow-based classifier can adjust packet sizes introduce delays split transfers across connections pad traffic reshape timing etc. all without changing what attack accomplishes.

Model extraction where attacker probes detector to determine its decision boundary then crafts inputs staying on benign side is feasible when detector is exposed to traffic attacker can generate. Defenses parallel those in broader field adversarial training input sanitization ensemble diversity detection of adversarial samples as separate classification problem & share same limitation robustness against known attack methods does not imply robustness against future ones.

Poisoning presents different problem particularly in federated settings where manipulated training contributions can bias shared model away from detecting attacker's own method. This is structural reason federated intrusion detection research devotes so much attention to aggregation honest participants' privacy is protected but model's integrity relies on rules assuming most participants are honest.

XI. DEPLOYMENT REALITIES

Published accuracy values characterize models. Deployed systems are socio-technical & discrepancy between two explains much of disappointment experienced in practice.

Scale and cost. Line-rate inspection requires either specialized hardware or careful sampling sampling degrades detection for precisely low-volume/low-and-slow behaviors most relevant to security. Storage for full capture is costly & often first budgetary line item eliminated. Flow export from high-capacity routers competes with router's primary function.

Alert fatigue. Analysts receiving more alerts than they can investigate prioritize alerts using heuristic reasoning heuristic reasoning becomes actual decision-making process. Operational precision for detector is therefore not precision measured on test set but precision as filtered by whoever reads queue at three AM. This explains why interpretability has practical significance: alert stating "host A contacted address never observed before in same hour it received email attachment" is actionable alert stating "anomaly score 0.87" is not.

Baseline maintenance. Entity-based detection requires knowledge of normal behavior per host requiring inventory ownership attribution process for maintaining baselines when systems change. Organizations lacking reliable inventory obtain unreliable baselines more often binding constraint than model quality.

Complementarity. Mature pattern is not single detector but layered one: signature rules for known threats flow analytics for breadth protocol logs for context endpoint telemetry for depth correlation across layers. Each layer addresses blind spots of other layers. Acknowledging this produces desirable byproduct: reduces pressure for any single detector to be complete shifts attention toward how effectively layers feed each other.

Privacy and legal exposure. Traffic data includes personal communications & can disclose health political financial information concerning individuals. Therefore, monitoring programs require governance model does not provide retention limits access controls logging who queried what etc. In jurisdictions with data protection law collection itself requires defensible rationale independent of its security objective.

XII. OPEN PROBLEMS

Several problems remain unsolved rather than merely under-optimized.
Evaluation representing deployment. Field requires benchmarks comprising current protocol mixes containing adversarial traffic generated by capable operators rather than default tool configurations evaluated using metrics weighting false positives based on analyst cost. Until cross-dataset generalization improves accuracy on single benchmark should be regarded as hypothesis rather than result.

Detection under full encryption. Encrypted Client Hello renders destination-name signal unavailable for connections adopting it. Whether flow-level/timing-level inference can bear detection burden currently carried by hostname inspection is open empirical question current results suggest it can substitute for some portion of burden not all portions.

Explainability as a requirement. If detector output must withstand human triage model must produce reasons not scores. Work on attention-based/graph-based methods suggest this but does not resolve it. Robustness without paralysis. Defenses against adversarial evasion are strongest against known attack methods hardening detector often sacrifices accuracy on normal traffic. Establishing defensible operating point represents engineering question literature has not resolved. Cost-aware/resource-constrained detection. Proliferation of IoT/industrial networks where endpoints cannot host agents & links are slow necessitates detectors operating from very thin observations. Most contemporary research assumes data volumes unobtainable in these settings.

XIII. CONCLUSION

Network traffic analysis remains among few security capabilities observing organization's activity independently of devices generating activity independence constitutes why network traffic analysis

survives every evolution in threat landscape. Nature of available evidence however has evolved plaintext era where detection could be reduced to string matching is ending encrypted era necessitates inference from metadata that is indirect probabilistic easily manipulated. Machine learning represents field's response generating actual capability concurrently with body of evaluation practice exaggerating capability. Benchmark accuracy serves as weak proxy for deployed performance models generalizing across datasets are scarcer than literature implies techniques enabling detection adaptiveness also render detector vulnerable to attack

Additionally operational constraints scale; cost; alert volume; baseline maintenance; privacy obligations dictate outcomes more profoundly than model architecture does. Most pragmatic means for conceptualizing traffic analysis is as one instrument among several: Traffic analysis offers broad coverage/shallow depth complements endpoint telemetry rather than substituting for endpoint telemetry generates hypotheses for analyst to validate rather than conclusions automatically implementable. Pursuit of research enhancing precision explainability robustness against manipulation is worthwhile; pursuit of research enhancing accuracy on anachronistic benchmarks is less valuable than number of publications implies.

REFERENCES

[1] S. Ennaji *et al*., "Adversarial Challenges in Network Intrusion Detection Systems: A Survey," *IEEE Access*, 2025.

[2] S. Ghosh et al., "Network traffic analysis based on cybersecurity intrusion detection through an effective Automated Separate Guided Attention Federated Graph Neural Network," Applied Soft Computing, 2025.

[3] Internet Engineering Task Force (IETF), TLS Encrypted Client Hello, draft-ietf-tls-esni-25, 2025.

[4] F. A. Khan et al., "Balanced Multi-Class Network Intrusion Detection Using Machine Learning," IEEE Access, 2024.

[5] MITRE Corporation, "ATT&CK Data Component DC0078: Network Traffic Flow," MITRE ATT&CK, 2021–2026.

[6] R. Panigrahi and S. Borah, "A detailed analysis of CICIDS2017 dataset for designing Intrusion Detection Systems," in Proc. International Conference on Electrical, Computer and Communication Technologies (ICECA), 2018.

[7] I. Pathak et al., "Malware Classification of Encrypted Network Traffic (2020–2025): A Review," IEEE Access, 2025.

[8] V. Paxson, "Bro: A System for Detecting Network Intruders in Real-Time," Computer Networks, vol. 31, nos. 23–24, pp. 2435–2463, 1999.

[9] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization," in Proc. International Conference on Information Systems Security and Privacy (ICISSP), 2018, pp. 108–116.

[10] K. Shrivastava et al., "A Federated Learning and Deep Learning-Based Intrusion Detection System," IEEE Access, 2025.

[11] R. Sommer and V. Paxson, "Outside the Closed World: On Using Machine Learning for Network Intrusion Detection," in Proc. IEEE Symposium on Security and Privacy, 2010, pp. 305–316.

[12] Z. Wang et al., "Feature mining for encrypted malicious traffic detection with deep learning and other machine learning algorithms," *Computers & Security*, vol. 128, 2023.