INTERNATIONAL JOURNAL OF TECHNOLOGY AND ACADEMIC RESEARCH

ISSN: 2224-6878 | Pakistan Volume 6, No. 11.

IF: 14.91 | 2026 http://waerasia.org/index.php/ijtar

NETWORK TRAFFIC FORMED BY ATTACKS IN COMPUTER NETWORKS AND THE ATTACK TRACES IT CARRIES

Ramazonova Madina Shavkatovna

Department of Cybersecurity and Forensics, Tashkent University of Information Technologies, Uzbekistan

Doi: https://doi.org/10.5281/zenodo.24507163

Abstract: Every attack on a computer network leaves a mark on the traffic that carries it. Those marks — shifts in flow duration, in packet and byte counts, in inter-arrival timing, in the entropy of address and port distributions, and in the regularity of connection behaviour — are what a detection system actually observes; the attack itself is never visible directly. This article examines how the main classes of network attacks reshape traffic, and which of the resulting traces are distinct enough and durable enough to support automated detection. Attack classes are mapped onto the traffic properties they alter, and the traces are organised by the level at which they become observable: packet, flow, time window and protocol log. Four feature families — statistical, temporal, entropy-based and behavioural — are defined to encode these traces numerically. The approach was evaluated on 1,473,350 traffic flows drawn from the CTU-13 dataset, segmented into 120-second windows with a 60-second shift. Feature selection by L1 regularisation and SVM-RFE identified flow duration and forward-direction packet counts as the most informative attributes. Five classifiers were compared under identical conditions; a neural network reached 97.65 % accuracy. The results confirm that statistical, entropy and temporal traces, used jointly, separate botnet traffic from normal traffic reliably, and that these traces survive encryption whereas payload signatures do not.

Keywords: attack traces, botnet detection, entropy features, network traffic analysis, traffic flow

Introduction

An attack on a computer network is not an event that can be observed directly. What an analyst or a detection system sees is traffic — sequences of packets, aggregated flow records, protocol logs — and the question is always the same: does this traffic differ from what the network produces when nothing is wrong, and if so, in what measurable way. Every intrusion technique, from a volumetric flood to a patient command-and-control channel, must move data across the network in order to work, and in moving that data it alters the statistical and temporal shape of the traffic. Those alterations are the traces of the attack.

The practical importance of this framing has grown with the spread of encryption. For most of the history of intrusion detection, the dominant technique was payload inspection: a rule matched a byte string characteristic of a known exploit, and the match was the evidence [7]. With the majority of application traffic now encrypted end to end, that evidence is no longer available on most links. What remains available is the envelope — who talks to whom, how often, for how long, with what volume and in what rhythm. A detection method built on the envelope rather than the contents is therefore not merely an alternative; on modern networks it is often the only option [13].

A second consideration concerns unknown attacks. Signature-based systems, by construction, recognise only what has already been catalogued [18]. Traces expressed as deviations from a

ISSN: 2224-6878 | Pakistan Volume 6, No. 11.

IF: 14.91 | 2026 http://waerasia.org/index.php/ijtar

learned profile of normal behaviour do not carry that restriction: a previously unseen botnet family still has to contact its controller, still has to do so repeatedly, and still produces connections whose periodicity and destination diversity differ from ordinary user traffic [20]. The trace survives even when the signature does not.

The aim of this article is to describe systematically which traces the main classes of network attacks leave in traffic, at which level of observation each trace becomes visible, how the traces can be encoded as numerical features, and how reliably those features separate malicious from normal traffic in practice. The experimental part is based on the CTU-13 dataset [10], which contains real botnet traffic captured alongside ordinary network activity, and reports the comparative performance of five classification algorithms on 1,473,350 flows.

Experimental

The experimental work followed a fixed pipeline, applied identically to every configuration so that the reported differences are attributable to the methods rather than to the conditions under which they were measured.

Data source. The CTU-13 dataset was selected after a comparison of eight public intrusion detection datasets, among them KDD Cup 99, NSL-KDD, UNSW-NB15 [12], CICIDS2017 [11], CSE-CIC-IDS2018, Bot-IoT and TON_IoT. The decisive property of CTU-13 is that it records real botnet activity carried on a live university network rather than traffic synthesised in a laboratory, and that normal and malicious traffic coexist in the same capture [10, 17]. The file capture20110815-2.binetflow was used for the experiments reported here.

Flow formation and segmentation. Packet records were aggregated into bidirectional flows keyed on the conventional five-tuple of source address, destination address, source port, destination port and protocol. Flows were then segmented into observation windows of 120 seconds with a shift of 60 seconds, so that consecutive windows overlap by half their length. The overlap matters: an attack that begins near a window boundary would otherwise be split across two windows and appear in neither with its full intensity.

Feature formation. Four families of features were computed for every window, corresponding to the four trace categories described in the following section. Statistical features summarise volume; temporal features summarise rhythm; entropy features summarise the spread of addresses and ports; behavioural features summarise the sequence and outcome of connection attempts. Entropy was computed in its relative form, normalised by the logarithm of the number of distinct values observed, so that windows containing different numbers of flows remain comparable [14, 22].

Preprocessing and feature selection. Missing values were imputed, noisy records removed and all attributes standardised to zero mean and unit variance. Because the informativeness of the generated attributes varies widely, two complementary selection methods were applied: L1 regularisation in the manner of the Lasso, which drives the coefficients of weak attributes to zero [8], and SVM-RFE, which ranks attributes by the contribution each makes to the separating hyperplane and eliminates them recursively [9].

Classification and evaluation. Five algorithms were trained on the selected feature set: Logistic Regression, Support Vector Machine [16], Random Forest [15], Gradient Boosting [19] and a feed-forward Neural Network. The dataset was divided into 80 % for training and 20 % for testing. Performance was assessed with Precision, Recall, F1-score and Accuracy, which together describe both the purity of the detections and the proportion of attacks that were found.

Results and Discussion

ISSN: 2224-6878 | Pakistan Volume 6, No. 11.

IF: 14.91 | 2026 http://waerasia.org/index.php/ijtar

3.1. Traffic produced by the main classes of attacks

Different attack techniques distort different properties of traffic, and the distortion is specific enough that the attack class can often be inferred from the shape of the deviation alone. Table 1 sets out this correspondence for the attack classes encountered most often on enterprise networks.

1-Table.

Traffic changes produced by the main classes of network attacks

Attack class	Change produced in the traffic	Most affected measurable property
Volumetric DDoS (UDP/ICMP flood)	Abrupt rise in packet rate and byte volume directed from many sources at one destination; flows become numerous and short	Packets per second; source-address entropy
TCP SYN flood	Large number of half-open connections; flows containing one forward packet and no reply	Forward/backward packet ratio; flow duration
Port scanning	One source contacts many ports on a few hosts within a short period; most flows carry one or two packets	Destination-port entropy; packets per flow
Network sweep	One source contacts many destination addresses on a single port	Destination-address entropy; flows per source
Botnet command and control	Periodic low-volume connections to a small set of external addresses with stable intervals	Connection periodicity; destination diversity
Brute-force authentication	Repeated short sessions to one service port, of nearly identical size	Flow-size variance; connection rate per port
Data exfiltration	Prolonged outbound flows with an unusual forward/backward byte ratio	Byte asymmetry; flow duration
ARP spoofing and man-in-the-middle	Duplicate address mappings, abnormal ARP volume, rerouted sessions	ARP request rate; session path anomalies

ISSN: 2224-6878 | Pakistan Volume 6, No. 11.

IF: 14.91 | 2026 http://waerasia.org/index.php/ijtar

Two patterns in Table 1 deserve emphasis. First, the attacks that generate the largest volume are the easiest to see but not necessarily the most damaging; a volumetric flood announces itself in every counter, whereas a command-and-control channel may move a few hundred bytes an hour. Second, the quiet attacks are not trace-free — they are trace-poor in volume and trace-rich in rhythm. A botnet client that contacts its controller every 300 seconds produces a regularity that ordinary human-driven traffic almost never exhibits [20], and that regularity is a stronger discriminator than any volume statistic.

3.2. Levels at which traces become observable

A trace exists only relative to a level of observation. The same attack that is invisible in a flow record may be obvious in a protocol log, and vice versa. Table 2 organises the traces by the unit at which they can be measured.

2-Table.

Levels at which attack traces become observable

Level	Unit of observation	Traces visible at this level	Principal limitation
Packet	Individual packet header and payload	Malformed headers, anomalous flag combinations, signature strings	Data volume; payload hidden by encryption
Flow	Aggregated five-tuple connection record	Duration, packet and byte counts, directional asymmetry	Per-packet detail is lost
Time window	Set of flows observed within a fixed interval	Periodicity, entropy of address and port distributions, behavioural patterns	Window length determines what is visible
Protocol log	Structured event records (connection, DNS, HTTP)	Protocol misuse, suspicious domains, repeated authentication failures	Coverage limited by the logging tool

The practical consequence is that a detection architecture should not commit to a single level. Packet-level inspection gives the finest detail and the shortest reach, since it cannot be sustained at high line rates and loses its value once payloads are encrypted. Flow records scale to backbone speeds and retain exactly the properties that survive encryption. The window level is where periodicity and entropy become computable at all, because both require a population of flows rather than an individual one. Protocol logs add semantic context that no amount of statistics can reconstruct.

ISSN: 2224-6878 | Pakistan Volume 6, No. 11.

IF: 14.91 | 2026 http://waerasia.org/index.php/ijtar

3.3. Sources of trace data

Each level of observation corresponds to a concrete data source available on production networks. Table 3 compares the four that were considered for this work.

3-Table.

Sources of attack-trace data

Source	Captured content	Strengths	Weaknesses
PCAP (tcpdump, Wireshark)	Complete packets with payload	Highest level of detail; supports forensic reconstruction	Storage cost; little value once traffic is encrypted
NetFlow / IPFIX	Flow records: five-tuple, counters, timestamps	Scales to high-rate links; widely supported by vendors	No payload; sampling can conceal short events
Zeek (Bro) logs	Protocol-level events in conn.log, dns.log, http.log	Rich semantic context; already structured for analysis	Processing cost on high-throughput links
IDS / IPS alerts	Rule and signature matches	Immediate and directly interpretable	Blind to unknown attacks; prone to false positives

NetFlow and IPFIX records [5, 6] were chosen as the primary source for the experiments, with Zeek logs [21] used for verification. The choice reflects the argument of Section 3.2: flow records carry precisely the statistical, temporal and entropy traces that remain measurable under encryption, and they do so at a data volume two to three orders of magnitude below full packet capture.

3.4. Encoding traces as features

To be usable by a classification algorithm, a trace must be expressed as a number. Table 4 lists the four feature families employed and the traces each is designed to capture.

4-Table.

Feature families used to encode attack traces

Family	Representative features	Attack traces captured
Statistical	Packet count, byte count, mean packet size, flow duration	Volumetric floods, exfiltration, abnormal session length

ISSN: 2224-6878 | Pakistan Volume 6, No. 11.

IF: 14.91 | 2026 http://waerasia.org/index.php/ijtar

Family	Representative features	Attack traces captured
Temporal	Inter-arrival time, connection periodicity, flow rate per window	Command-and-control beaconing, slow scanning
Entropy-based	Relative uncertainty of source and destination address and port distributions	Scanning, distributed floods, address diversification
Behavioural and protocol	Connection sequence, flag distribution, protocol mix, failure ratio	Brute force, protocol misuse, coordinated botnet activity

The entropy family warrants a note of its own. Relative uncertainty measures how evenly a distribution is spread across its possible values, and it reacts to exactly the behaviours that scanning and distributed attacks produce: a port scan collapses the destination-port distribution towards uniformity, while a flood concentrates the destination-address distribution onto a single point [14]. Because the measure is normalised, it is insensitive to the absolute volume of traffic, which makes it robust on links whose load varies by an order of magnitude between day and night [22].

3.5. Experimental evaluation

Table 5 describes the composition of the dataset after flow formation and labelling.

5-Table.

Composition of the experimental dataset

Indicator	Category	Number of flows	Share, %
Traffic classes	Normal traffic	757 742	51.43
	Botnet traffic	715 608	48.57
Sample split	Training	1 178 680	80.00
	Testing	294 670	20.00
Total volume	All traffic flows	1 473 350	100.00

The balance between the two classes — 51.43 % normal against 48.57 % botnet — is unusual for an intrusion detection dataset and is advantageous here, since it removes the need for resampling and makes accuracy a meaningful figure rather than a misleading one.

Feature selection reduced the initial attribute space considerably. Table 6 reports the attributes that received the highest importance scores.

6-Table.

ISSN: 2224-6878 | Pakistan Volume 6, No. 11.

IF: 14.91 | 2026 http://waerasia.org/index.php/ijtar

Most informative attributes after feature selection

Rank	Attribute	Importance	Trace encoded by the attribute
1	duration	0.214	How long a connection is held open
2	tot_fwd_packets	0.178	Packet volume in the forward direction
3	total_fwd_packets	0.152	Aggregated forward-direction packet count

The ranking is informative in itself. Flow duration being the single most important attribute confirms the argument of Section 3.1: what distinguishes botnet traffic is not how much it carries but how long and how regularly its connections are held. The two forward-direction packet counters that follow reflect the directional asymmetry typical of client-initiated malicious sessions, in which the infected host sends far more than it receives during beaconing and the ratio inverts during payload delivery.

Table 7 reports the classification performance of the five algorithms on the held-out test set.

7-Table.

Classification performance of the compared models, %

Model	Precision	Recall	F1-score	Accuracy
Logistic Regression	85.23	84.19	84.70	86.12
Support Vector Machine	91.45	90.28	90.86	91.90
Random Forest	94.07	93.15	93.60	94.42
Gradient Boosting	96.03	95.21	95.62	96.41
Neural Network	97.31	96.58	96.94	97.65

The ordering of the results follows the capacity of the models to represent non-linear interactions among the features. Logistic Regression, restricted to a linear decision boundary, reaches 86.12 % accuracy — respectable, and a useful indication that the selected features carry genuine signal, but insufficient for operational use given that roughly one flow in seven is misclassified. The Support Vector Machine gains almost six percentage points. The two ensemble methods gain

ISSN: 2224-6878 | Pakistan Volume 6, No. 11.

IF: 14.91 | 2026 http://waerasia.org/index.php/ijtar

further: Random Forest reaches 94.42 % and Gradient Boosting 96.41 %, the latter benefiting from its sequential correction of the errors of earlier weak learners [19]. The neural network attains 97.65 % accuracy with Precision of 97.31 % and Recall of 96.58 %, the balance between the two indicating that the gain comes from genuinely better separation rather than from a bias towards one class.

3.6. Durability of traces

For an operational system, the accuracy of a trace matters less than whether the trace can be relied upon to persist. Three pressures erode traces: encryption removes content, sampling removes detail, and a deliberate adversary removes regularity. Table 8 assesses the main trace categories against these pressures.

8-Table.

Durability of the main categories of attack trace

Trace category	Survives encryption	Resistance to deliberate evasion	Extraction cost
Payload signature	No	Low	Low
Volume statistics	Yes	Medium	Low
Timing and periodicity	Yes	Medium to high	Medium
Entropy of address and port distributions	Yes	High	Medium
Behavioural sequences	Yes	High	High

The table explains why the feature design adopted here places little weight on payload and considerable weight on entropy and behaviour. Payload signatures, the historical foundation of intrusion detection, are the first casualty of encryption and provide no protection against attacks that have not yet been catalogued. Volume statistics survive encryption but are the easiest traces for an adversary to suppress, simply by sending less. Timing regularity is harder to disguise, because the operational requirement that drives it — a client must check in with its controller — does not disappear; randomising the interval widens the distribution but rarely reproduces the irregularity of human-driven traffic. Entropy and behavioural traces are the most resistant of all, since suppressing them requires the attacker to abandon the distributed or repetitive structure that makes the attack effective in the first place.

This also sets the limits of the approach honestly. The results reported above were obtained on one dataset, captured on one network, with botnet families active at the time of capture. The relative ordering of the algorithms and the dominance of duration-based features are expected to transfer; the absolute figures should not be assumed to hold on a different network without fresh measurement. This caution applies with particular force to machine learning in intrusion detection, a field in which laboratory accuracy has repeatedly failed to survive deployment [18].

ISSN: 2224-6878 | Pakistan Volume 6, No. 11.

IF: 14.91 | 2026 http://waerasia.org/index.php/ijtar

Conclusion

Attacks on computer networks are detectable because they cannot act without moving data, and moving data changes the measurable shape of traffic. This article has set out those changes class by class, located them at the level of observation where each becomes visible, and organised them into four feature families — statistical, temporal, entropy-based and behavioural — that encode them in a form a classifier can use.

The experimental results support three conclusions. First, flow-level records are a sufficient basis for detection: on 1,473,350 flows from the CTU-13 dataset, features derived from flow records alone separated botnet traffic from normal traffic with up to 97.65 % accuracy, without any access to packet payloads. Second, the most informative single attribute was flow duration (importance 0.214), ahead of the forward-direction packet counters, which confirms that the temporal structure of malicious connections is a stronger discriminator than their volume. Third, model capacity matters: the progression from Logistic Regression (86.12 %) through Support Vector Machine (91.90 %), Random Forest (94.42 %) and Gradient Boosting (96.41 %) to the neural network (97.65 %) shows that the relationships among these features are substantially non-linear.

The wider implication concerns the direction in which intrusion detection should develop. As encryption becomes universal, the traces that detection can rely upon are no longer those carried in the content of communication but those carried in its pattern. Methods built on statistical, entropy and behavioural traces are not a supplement to signature-based detection under such conditions; they are its replacement. Future work will extend the evaluation to additional datasets and network environments, and will examine how far these traces persist when an adversary actively attempts to suppress them.

References

1. Stallings, W. (2020). Network Security Essentials: Applications and Standards. Pearson Education.
2. Anderson, R. (2020). Security Engineering: A Guide to Building Dependable Distributed Systems (3rd ed.). Wiley.
3. Bhuyan, M. H., Bhattacharyya, D. K., & Kalita, J. K. (2014). Network anomaly detection: methods, systems and tools. IEEE Communications Surveys & Tutorials, 16(1), 303–336.
4. Shannon, C. E. (1948). A mathematical theory of communication. Bell System Technical Journal, 27(3), 379–423.
5. Claise, B. (2004). Cisco Systems NetFlow Services Export Version 9. RFC 3954, IETF.
6. Claise, B., Trammell, B., & Aitken, P. (2013). Specification of the IP Flow Information Export (IPFIX) Protocol. RFC 7011, IETF.
7. Scarfone, K., & Mell, P. (2007). Guide to Intrusion Detection and Prevention Systems (IDPS). NIST Special Publication 800-94.
8. Tibshirani, R. (1996). Regression shrinkage and selection via the Lasso. Journal of the Royal Statistical Society, Series B, 58(1), 267–288.
9. Guyon, I., Weston, J., Barnhill, S., & Vapnik, V. (2002). Gene selection for cancer classification using support vector machines. Machine Learning, 46(1–3), 389–422.
10. Garcia, S., Grill, M., Stiborek, J., & Zunino, A. (2014). An empirical comparison of botnet detection methods. Computers & Security, 45, 100–123.

ISSN: 2224-6878 | Pakistan Volume 6, No. 11.

IF: 14.91 | 2026 http://waerasia.org/index.php/ijtar

11. Sharafaldin, I., Lashkari, A. H., & Ghorbani, A. A. (2018). Toward generating a new intrusion detection dataset and intrusion traffic characterization. In Proceedings of ICISSP (pp. 108–116).
12. Moustafa, N., & Slay, J. (2015). UNSW-NB15: a comprehensive data set for network intrusion detection systems. In Military Communications and Information Systems Conference (MilCIS).
13. Mirsky, Y., Doitshman, T., Elovici, Y., & Shabtai, A. (2018). Kitsune: an ensemble of autoencoders for online network intrusion detection. In Proceedings of NDSS.
14. Lakhina, A., Crovella, M., & Diot, C. (2005). Mining anomalies using traffic feature distributions. In Proceedings of ACM SIGCOMM (pp. 217–228).
15. Breiman, L. (2001). Random forests. Machine Learning, 45(1), 5–32.
16. Cortes, C., & Vapnik, V. (1995). Support-vector networks. Machine Learning, 20(3), 273–297.
17. Ring, M., Wunderlich, S., Scheuring, D., Landes, D., & Hotho, A. (2019). A survey of network-based intrusion detection data sets. Computers & Security, 86, 147–167.
18. Sommer, R., & Paxson, V. (2010). Outside the closed world: on using machine learning for network intrusion detection. In IEEE Symposium on Security and Privacy (pp. 305–316).
19. Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. Annals of Statistics, 29(5), 1189–1232.
20. Gu, G., Perdisci, R., Zhang, J., & Lee, W. (2008). BotMiner: clustering analysis of network traffic for protocol- and structure-independent botnet detection. In Proceedings of USENIX Security Symposium (pp. 139–154).
21. Paxson, V. (1999). Bro: a system for detecting network intruders in real-time. Computer Networks, 31(23–24), 2435–2463.
22. Nychis, G., Sekar, V., Andersen, D. G., Kim, H., & Zhang, H. (2008). An empirical evaluation of entropy-based traffic anomaly detection. In Proceedings of ACM IMC (pp. 151–156).
23. Buczak, A. L., & Guven, E. (2016). A survey of data mining and machine learning methods for cyber security intrusion detection. IEEE Communications Surveys & Tutorials, 18(2), 1153–1176.