iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** V **Month of publication:** May 2026

DOI: https://doi.org/10.22214/ijraset.2026.81963

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue V May 2026- Available at www.ijraset.com

SNA-IDS: Social Network Analysis-Augmented Malware Traffic Detection Using Graph-Theoretic Features and Isolation Forest

Susmitha Hemanth, Chetan M, Vedanth J, Rachana J, Nandan Gowda T M
Department of Computer Science & Engineering (ICB) BMS College of Engineering, Bengaluru, India

Abstract: ***This paper introduces SNA-IDS, an innovative intrusion detection framework that brings together Social Network Analysis (SNA) principles and unsupervised machine learning to identify malware-laden network traffic. Whereas most conventional anomaly-detection systems treat each network flow as an isolated observation, SNA-IDS instead represents the entire communication fabric as a continuously evolving graph—with individual hosts as nodes and packet flows as weighted, directed edges—and draws on a rich collection of graph-theoretic descriptors in addition to classical flow-level statistics. We operationalise key SNA concepts—PageRank, betweenness centrality, structural holes, tie strength, triadic closure, community partitioning, homophily, and cascade diffusion—to characterise the behavioural topology that distinguishes normal traffic from adversarial activity. The resulting graph features are concatenated with flow-level attributes and submitted to an Isolation Forest anomaly scorer. A subsequent two-stage reranking step refines the raw alert list by incorporating network-position evidence, and SHAP (Shapley Additive Explanations) delivers per-alert reasoning that security analysts can act on directly. Validation across publicly available malware benchmarks shows that adding SNA-derived features raises detection accuracy by six to nine percentage points compared with flow-only baselines, yielding a 97.8 % true-positive rate with a false-positive rate below 2.1 %. SNA-IDS reliably uncovers Remote Access Trojans (RATs), botnet infrastructures, DNS tunnelling channels, Command-and-Control (C2) networks, and coordinated cascade attacks by exploiting the structural signatures these threats imprint on the communication graph.***

Keywords — Social network analysis; graph-theoretic features; malware detection; Isolation Forest; PageRank; structural holes; tie strength; community detection; cascading behaviour; homophily; SHAP; unsupervised learning.

I. INTRODUCTION

Contemporary cyber threats no longer originate from a single, isolated actor. Advanced Persistent Threats (APTs), botnet campaigns, and organised Command-and-Control (C2) infrastructures are intrinsically social in nature: they emerge from carefully structured relationships among hosts, travel along established communication pathways, and exploit the underlying topology of the target network in ways that closely mirror how information or disease diffuses through human populations [1]. Despite this fundamentally relational character, the dominant paradigm in Network Intrusion Detection System (NIDS) research continues to treat each network flow as a self-contained, independent event—an approach that discards the rich structural context surrounding every packet exchange.

Social Network Analysis (SNA) supplies a mathematically grounded vocabulary for precisely these relational structures. Concepts such as triadic closure, tie strength, structural holes, homophily, PageRank, and cascade diffusion were developed to capture how actors behave given their positions in a social network—a perspective that translates directly and productively into the domain of network security. A command-and-control server, for example, occupies a structural hole between the botmaster and its victim fleet. Botnet nodes exhibit homophily—infected machines preferentially connect to one another and to shared C2 infrastructure. Malware propagation adheres to diffusion and threshold dynamics as it spreads across the communication graph. These topological signatures are completely opaque to per-flow anomaly detectors yet become strikingly visible once the communication fabric is treated as a social graph and analysed accordingly.

This paper presents SNA-IDS, a system that bridges the gap between social network analysis and network security, yielding a substantially more powerful malware detector as a result. SNA-IDS constructs a weighted, directed communication graph from raw PCAP traffic in real time, extracts a comprehensive set of SNA-grounded features from this graph, and combines them with conventional flow-level attributes before feeding the joint representation into an Isolation Forest anomaly detector.

The relational features capture properties that individual flows cannot: a node's importance within the communication hierarchy, its role as a broker bridging otherwise disconnected host clusters, the clustering density of its local neighbourhood, its position within cascade propagation chains, and the degree to which it associates exclusively with behaviourally similar peers. Our two-stage reranking pipeline further exploits graph structure to suppress spurious alerts, while SHAP explanations make the contribution of graph features transparent and auditable for security operations teams.

The primary contributions of this work are as follows: (1) We formalise the mapping from raw network traffic to a social communication graph and define a 15-dimensional SNA feature vector that can be computed incrementally at line rate. (2) We show how PageRank, betweenness centrality, structural holes, triadic closure, homophily, and diffusion models each supply distinct, complementary signals for malware detection. (3) We demonstrate experimentally that SNA-augmented Isolation Forest outperforms flow-only detection by six to nine percentage points in accuracy while simultaneously reducing false positives. (4) We introduce graph-aware two-stage reranking and SNA-enriched SHAP explanations. (5) We release an open-source implementation aimed at enterprise and IoT deployment contexts.

The remainder of this paper is organised as follows. Section II reviews related work. Section III formalises the social network model for network traffic. Section IV details the methodology and SNA feature engineering. Section V describes the overall system architecture. Section VI reports experimental results. Sections VII and VIII discuss limitations and future directions respectively. Section IX concludes.

II. RELATED WORK AND MOTIVATION

A. Flow-Based Anomaly Detection

Flow-level anomaly detection has attracted sustained research interest. Unsupervised methods—including Isolation Forest (IF) [2], One-Class SVMs, and autoencoder architectures—condense packet headers into five-tuple flow records and identify statistical outliers among them. Liu et al. [2] demonstrated that Isolation Forest can efficiently separate anomalies through a sequence of random binary partitions, achieving competitive detection rates on standard network benchmarks. Sun et al. [3] extended IF to DNS traffic for tunnelling detection, reporting 98.1 % accuracy. Neuschmied et al. [4] proposed a two-stage autoencoder design in which a refinement pass substantially reduces the false-positive burden. Effective as these approaches are, they share a common blind spot: they operate on flows independently and therefore ignore the relationship structure that those flows collectively form across the network.

B. Graph-Based Intrusion Detection

Security researchers have increasingly turned to graph representations for threat detection. Ding et al. [5] built IP communication graphs and applied graph clustering to identify botnet membership. Iliofotou et al. [6] studied traffic dispersion graphs to surface application-layer communication patterns. Zhao et al. [7] exploited the degree distribution of host communication graphs to fingerprint peer-to-peer botnets. More recently, Graph Neural Networks (GNNs) have been applied to intrusion detection [8] with strong results, though these methods depend on labelled training data and considerable computational resources. SNA-IDS differs by grounding detection in established SNA theory—particularly tie strength, structural holes, homophily, and cascading dynamics—and in doing so provides not only detection capability but also interpretable relational signatures.

C. Social Network Analysis in Cybersecurity

SNA has previously been applied to e-mail spam networks [9], social-media-borne malware [10], and cyber threat intelligence graphs [11]. Starnini et al. [12] leveraged temporal network analysis to surface coordinated inauthentic behaviour. In the network traffic domain, Narang et al. [13] applied community detection to host communication graphs in order to isolate botnet clusters. Nevertheless, a systematic and unified integration of the full SNA theoretical toolkit—covering tie strength, structural holes, triadic closure, homophily, PageRank, cascade diffusion, and small-world characteristics—within a single malware detection pipeline has not previously been demonstrated. SNA-IDS addresses exactly this gap.

D. Explainable Anomaly Detection

The interpretability of anomaly detectors has become a practical necessity for operational trust. SHAP [14] decomposes model predictions into per-feature additive contributions. Visser et al. [15] adapted SHAP to unsupervised Isolation Forest, computing Shapley interaction values for outlier scores.

SNA-IDS adopts this approach and augments it with graph-context explanations, so that analysts are told not only which features were anomalous but also what network position drove the alert—for instance, that the flagged host occupies a structural hole between two otherwise disconnected host communities, a pattern consistent with C2 server behaviour.

III. SOCIAL NETWORK MODEL FOR NETWORK TRAFFIC

A. Formal Graph Construction

We represent the communication topology of a monitored network as a directed, weighted multigraph G = (V, E, W), where V is the set of active IP addresses observed within a sliding time window T_w (e.g., five minutes), E □ V × V is the set of directed edges encoding observed flows between host pairs, and W : E → R^d is a weight function mapping each edge to its flow feature vector (byte counts, packet counts, duration, protocol identifiers, TCP flag tallies, and so forth). Formally:

G(t) = (V(t), E(t), W(t)) where E(t) = { (u,v) | □ flow from u to v in [t − T_w, t] }

The graph is updated incrementally as new flows arrive. When multiple flows connect the same host pair within the window, their feature vectors are merged—summed or averaged depending on the feature type. This dynamic communication graph serves as the substrate for all subsequent SNA feature extraction. In large environments, node cardinality |V| can reach tens of thousands; we employ sparse adjacency structures and efficient BFS-based traversal to keep computation tractable.

B. Mapping SNA Concepts to Network Traffic

The communication graph G admits a direct and productive interpretation through the lens of SNA theory. Table I maps established SNA concepts to their network security counterparts, motivating each feature category that we extract in Section IV.

TABLE I. SNA CONCEPTS MAPPED TO NETWORK TRAFFIC SECURITY

SNA Concept	Network Traffic Analog	Security Significance
Node (Actor)	IP Host	Could be an attacker, victim, or intermediary relay
Edge (Tie)	Network Flow	Represents a communication path between two hosts
Tie Strength	Flow volume × frequency	Strong ties signal persistent C2 sessions; weak ties hint at covert exfiltration paths
Triadic Closure	3-host mutual exchange	Legitimate hosts form triangles naturally; malware nodes rarely do
Structural Holes	Broker connecting clusters	C2 nodes bridge disconnected segments; RAT relays link attacker to victim
Homophily	Infected hosts cluster	Botnet members show preference to interconnect with one another
PageRank	Random-walk importance score	High-PageRank nodes correlate strongly with C2 server roles
Hub/Authority (HITS)	Hub points outward; Auth is pointed to	Bots act as hubs targeting the C2 authority
Cascading Behavior	Infection propagation chain	Malware spreads along edges of the communication graph
Threshold Model	Infection tipping point	A host gets compromised when enough neighbors are already infected
Small-World Property	Short paths, dense local clusters	Botnet design mirrors small-world structure for fast command relay

SNA Concept	Network Traffic Analog	Security Significance
Community Detection	Dense sub-graphs	Botnet clusters stand apart from normal-traffic communities
BFS / Shortest Path	Reachability from source	Measures how far malware can propagate from an initial seed
Bow-Tie Structure	SCC core plus periphery	C2 infrastructure tends to occupy the strongly connected core

IV. METHODOLOGY

A. Traffic Communication Graph Construction

Graph construction proceeds in parallel with the existing PCAP parsing and flow aggregation pipeline. As each flow is finalised, its source and destination IP addresses are added as nodes v_src, v_dst □ V, and a directed edge (v_src, v_dst) is inserted into E with a weight vector w encoding the associated flow statistics (duration, byte count, packet count, flag tallies, and so on). Multiple flows between the same host pair within window T_w are merged by aggregating their weight vectors. The resulting graph G(t) therefore captures all host-level communication activity within the current analysis window as a weighted directed structure.

To support efficient SNA feature computation, G(t) is maintained as a sparse adjacency representation. Breadth-First Search (BFS) underpins reachability and shortest-path queries, supporting both diffusion simulation and Bow-Tie decomposition. We apply BFS iteratively to all hosts flagged as anomalous in order to measure their graph distance from known malicious seeds when threat-intelligence labels are available. The time complexity of a single BFS pass is O(|V| + |E|), which remains tractable for networks with up to roughly 100,000 active hosts.

B. Tie Strength Analysis

Drawing on Granovetter's foundational work on tie strength [16], we classify every edge in G as either a strong or a weak tie using a composite strength metric:

$$TS(u, v) = \alpha \cdot freq(u,v) + \beta \cdot vol(u,v) + \gamma \cdot dur(u,v)$$

where freq(u,v) denotes the number of flows between u and v during window T_w, vol(u,v) the total byte volume, dur(u,v) the mean flow duration, and α, β, γ are normalisation constants. Edges whose TS exceeds the 75th percentile are labelled strong ties; the remainder are weak ties. This distinction carries direct security significance. Strong ties correspond to persistent, high-bandwidth connections—the hallmark of internal file servers and backup systems, but equally of active RAT channels and live botnet C2 sessions. Weak ties, though infrequent and low-volume, serve as bridges between otherwise disconnected host clusters. In Granovetter's framework, weak ties are the primary conduit for cross-community information diffusion; in the security domain, weak-tie bridges between separate host clusters are precisely the structures exploited by stealthy data-exfiltration pipelines and multi-hop C2 relay chains [16]. Nodes that act as weak-tie bridges between communities are therefore elevated to high-priority anomaly candidates.

C. Triadic Closure and Clustering Coefficient

Triadic closure—the tendency for two nodes sharing a common neighbour to form a direct link—is a well-documented property of social networks. In benign network traffic, service-oriented communication naturally produces closed triads: if host A communicates with server S and host B also communicates with S, A and B are likely to interact directly as well. The local clustering coefficient of node v is:

$$C(v) = |\{ (u,w) \in E : u,w \in N(v) \text{ and } (u,w) \in E \}| / (k_v (k_v - 1))$$

where N(v) is the neighbour set of v and k_v = |N(v)|. Botnet nodes and RAT-infected hosts exhibit characteristically low clustering coefficients: they contact a C2 server and potentially a handful of peer bots, but these neighbours seldom communicate with one another—the C2 deliberately avoids unnecessary exposure. An unusually low C(v) for a highly active host, when considered alongside other anomaly evidence, is a strong infection indicator. We compute C(v) for all nodes in G(t) and include it as an SNA feature, along with the total triangle count Δ(v) containing v, since genuine botnet nodes show near-zero triangle counts despite high degree.

D. Structural Holes and Brokerage

Structural holes, formalised by Burt [17], are gaps in the network topology where a single broker node serves as the primary link between two otherwise disconnected clusters. The structural constraint C_i quantifies how tightly a node's contacts are interconnected with one another:

$C_i = \Sigma_j (p_{ij} + \Sigma_q p_{iq} \cdot p_{qj})^2, \quad j \neq i$

where p_ij is the proportion of node i's interaction budget invested in contact j. A low C_i signals that i bridges structural holes—it connects groups that would otherwise be isolated. In network traffic, C2 servers systematically occupy this broker position: they alone link the botmaster to victim hosts and coordinate communication across multiple infected subnets. Relay nodes in multi-hop RAT architectures play an analogous role. We compute C_i for each node and treat low-constraint, high-degree hosts as strong candidates for C2 infrastructure. We also calculate betweenness centrality:

$B(v) = \Sigma_{s \neq v \neq t} \sigma(s,t|v) / \sigma(s,t)$

where σ(s,t) is the total count of shortest paths from s to t and σ(s,t|v) is the number of these paths that traverse v. High betweenness centrality identifies nodes that serve as indispensable communication relays—precisely the role filled by C2 servers and botnet command proxies.

E. PageRank and Hubs-and-Authorities (HITS)

PageRank [18] assigns an importance score to each node on the recursive principle that a node is significant if other significant nodes direct their edges toward it. Applied to the traffic communication graph, the PageRank score PR(v) satisfies:

$PR(v) = (1-d)/|V| + d \cdot \Sigma_{u:(u,v) \in E} PR(u) / \text{out-deg}(u)$

where d = 0.85 is the standard damping factor. In malicious traffic, C2 servers receive flows from many infected bots while initiating comparatively few connections of their own—a combination that produces anomalously high PageRank relative to out-degree. PageRank is recomputed on G(t) at every window step and included as a node-level feature.

We additionally run the Hubs-and-Authorities (HITS) algorithm. In the botnet context, infected machines function as hubs pointing toward the C2 authority, while the C2 server accumulates a high authority score because it is the target of many hub nodes. HITS iterates hub scores $h(v) = \Sigma_{(v,u) \in E} a(u)$ and authority scores $a(v) = \Sigma_{(u,v) \in E} h(u)$ until convergence. A newly observed IP with high authority but low hub score is a strong candidate for a C2 server, whereas nodes with high hub scores and many novel destinations may be engaging in scanning or distributed denial-of-service activity.

F. Homophily Detection

Homophily—the tendency of similar nodes to preferentially form connections—is a defining feature of infected networks. Compromised hosts exhibit behavioural homophily: because they run the same malware family, they share similar communication profiles (consistent port usage, comparable traffic volumes, coordinated timing). We quantify homophily through the assortativity coefficient r:

$r = [\Sigma_e j_e k_e - (\Sigma_e (j_e + k_e)/2)^2] / [\Sigma_e (j_e^2 + k_e^2)/2 - (\Sigma_e (j_e + k_e)/2)^2]$

where j_e and k_e are the degrees of the nodes at the two ends of edge e. Beyond degree-based assortativity, we measure feature-level homophily: for each connected pair (u,v), we compute the cosine similarity between their respective flow feature vectors. Nodes whose neighbours show high mutual feature similarity—yet whose own profiles diverge markedly from the broader network average—are likely members of a botnet cluster acting in concert. We flag sets of densely interconnected nodes that all exhibit similar anomalous profiles as coordinated attack clusters.

G. Community Detection for Botnet Cluster Identification

Community detection algorithms partition V into densely connected sub-groups. We apply the Louvain modularity optimisation method to G(t), which greedily maximises the modularity objective Q:

$Q = (1/2m) \Sigma_{ij} [A_{ij} - k_i k_j / 2m] \delta(c_i, c_j)$

where A_ij is the weighted adjacency-matrix entry, k_i and k_j are node degrees, m is the total edge weight, and $\delta(c_i, c_j) = 1$ when nodes i and j share a community. Once partitions are obtained, each community is tested for anomalous characteristics. A community displaying high internal flow volume but no connections to common benign endpoints (e.g., known DNS resolvers or software update servers) is flagged. Communities in which the constituent nodes collectively exhibit low clustering coefficients, elevated betweenness, and anomalous PageRank are identified as candidate botnet clusters—a characterisation that proves especially effective for detecting coordinated scanning campaigns and DDoS floods.

We further incorporate the concept of affiliation networks: beyond direct flows, hosts are linked by shared service endpoints. Devices that contact the same unusual destination IP—possibly an obscure C2 domain—form a bipartite affiliation graph H = (V_hosts ∪ V_services, E_affil). Projecting H onto V_hosts reveals implicit groupings of infected devices that may never communicate directly but share a common malicious service, a pattern characteristic of distributed botnet membership.

H. Cascading Behaviour and Diffusion Models

Malware propagation through a network is fundamentally a cascading process. We model infection spread using two complementary diffusion frameworks from SNA theory:

1) Linear Threshold Model (LTM):

Each host v holds an infection threshold θ_v □ [0,1]. A host becomes infected when the weighted fraction of its already-infected neighbours exceeds this threshold:

Infected(v) = 1 iff Σ_{u □ N(v), Infected(u)} w(u,v) / Σ_{u □ N(v)} w(u,v) ≥ θ_v

Hosts with low thresholds and numerous connections to anomalous nodes are flagged as high-risk cascade targets. The LTM thereby enables proactive alerting—identifying nodes likely to become infected before that infection is directly observed.

2) Independent Cascade Model (ICM):

Under ICM, each infected host u independently attempts to infect each neighbour v with probability p(u,v). The expected cascade size is estimated through BFS-based simulation on G(t) starting from every flagged anomalous node. A flagged node whose expected cascade size greatly exceeds the graph average is elevated to high-severity status, suggesting it is a bot or C2 server with broad propagation potential. The ICM simulation runs in O(k · (|V| + |E|)) time, where k is the number of flagged nodes.

The role of weak ties in diffusion is particularly important within this framework. Following Granovetter's Strength of Weak Ties hypothesis [16], weak-tie bridges carry cascades across community boundaries. In botnet propagation, initial infection of a new subnet often arrives via a single low-volume flow—a weak tie—from an already-compromised host. Our diffusion model specifically tracks whether flagged cascade edges belong to the weak-tie category, as this pattern is a reliable indicator of cross-community malware propagation rather than routine intra-community traffic.

I. Small-World Properties of Botnet Topologies

The small-world phenomenon—short average path lengths coexisting with high local clustering—is a well-established characteristic of many social networks [19]. Botnets deliberately engineer small-world topology to maximise command-propagation speed: any infected host can be reached from the C2 server within a small number of hops, while dense local clustering ensures resilience against node removal. We test G(t) for small-world characteristics by computing the characteristic path length L and comparing it against a random Erdős–Rényi graph with equivalent |V| and |E|. A graph whose path length is substantially shorter than expected for its level of clustering suggests engineered small-world structure. Formally, the small-world coefficient σ is:

σ = (C / C_rand) / (L / L_rand) ; σ >> 1 indicates small-world structure

We also decompose the directed graph G(t) into its strongly connected component (SCC), the IN set (nodes able to reach the SCC), the OUT set (nodes reachable from the SCC), and fringe tendrils. In botnet topology, the SCC typically contains C2 infrastructure and relay nodes; the IN set consists of recently infected hosts sending initial beacons; the OUT set contains victim machines receiving commands. This macro-level structural decomposition provides a powerful signature of organised attack infrastructure.

J. SNA Feature Vector Construction

For each host node v □ V in the communication graph G(t), we extract a 15-dimensional SNA feature vector F_SNA(v) that summarises its graph-theoretic profile. These features are concatenated with the per-flow feature vector F_flow to yield the complete representation F(v) = [F_flow ‖ F_SNA] consumed by the Isolation Forest. Table II enumerates the SNA features.

TABLE II. SNA FEATURES EXTRACTED FROM COMMUNICATION GRAPH

SNA Feature	Symbol	Security Relevance
In-Degree Centrality	deg_in(v)	High in-degree suggests the node is a C2 target receiving beacon traffic
Out-Degree Centrality	deg_out(v)	High out-degree points to scanners or bots contacting

SNA Feature	Symbol	Security Relevance
		multiple peers
Betweenness Centrality	B(v)	Elevated betweenness marks relay or proxy nodes in an attack chain
PageRank Score	PR(v)	Disproportionately high PR relative to out-degree flags C2 authority
Hub Score (HITS)	h(v)	High hub score characterises bots directing traffic toward the C2
Authority Score (HITS)	a(v)	High authority score corresponds to a C2 server or malware repository
Clustering Coefficient	C(v)	Unusually low C flags bot nodes whose neighbors seldom talk to each other
Triangle Count	Δ(v)	Near-zero triangle count with high degree is a strong malware indicator
Structural Constraint	C_i	Low constraint marks brokers that span structural holes in the graph
Tie Strength (max)	TS_max(v)	Abnormally high TS reveals an active, high-bandwidth C2 channel
Tie Strength (min)	TS_min(v)	Near-zero TS toward unknown IPs indicates stealthy data exfiltration
Community ID	comm(v)	Community label used for cluster-level correlated threat analysis
Cascade Reach	CR(v)	Expected number of hosts reachable if v initiates a cascade under ICM
Homophily Score	H(v)	High score signals membership in a coordinated botnet cluster
Small-World σ	σ(G)	Graph-level metric; σ >> 1 exposes engineered botnet-like topology

K. Isolation Forest with SNA-Augmented Features

The Isolation Forest (IF) algorithm [2] is fitted on the full feature vector F(v) = [F_flow ‖ F_SNA] during a baseline period assumed to contain predominantly benign traffic. The ensemble comprises T = 150 random isolation trees. Within each tree, a feature is chosen at random from F(v) and a split value is drawn uniformly between the observed minimum and maximum of that feature in the current subsample, repeating until each instance is isolated or a maximum depth of $\lceil \log_2(\psi) \rceil$ is reached (where ψ is the subsample size, typically 256). The anomaly score is:

$$\text{Score}(x) = 2^{-E[h(x)] / c(n)}, \quad c(n) = 2H(n-1) - 2(n-1)/n$$

where H(i) is the i-th harmonic number and n is the training set size. The SNA features materially sharpen score separation: a C2 server that might produce only a marginal flow-level anomaly score is decisively flagged once the IF ensemble jointly evaluates its elevated PageRank, high betweenness, suppressed clustering coefficient, and near-zero structural constraint. The contamination factor is set to 2 % (consistent with expected attack prevalence) and the primary decision threshold τ_1 is tuned to maximise F1 on a held-out validation set.

L. Graph-Aware Two-Stage Reranking

Flows that surpass the primary IF threshold τ_1 are passed to a two-stage reranking module. In Stage II, a secondary score is computed that folds in graph-level context:

$\text{Score\_2}(v) = \lambda \cdot \text{Score\_IF}(v) + (1-\lambda) \cdot \text{GScore}(v)$

where GScore(v) is a composite of normalised graph features (betweenness, cascade reach, and structural constraint), and $\lambda = 0.75$. Flows are reranked by Score_2(v) and a stricter threshold $\tau_2 > \tau_1$ is applied to produce the final alert list. This stage specifically targets false positives arising from unusual-but-legitimate traffic: a software update server may rank highly on PageRank (many clients contact it), but its elevated clustering coefficient and near-average structural constraint clearly distinguish it from a C2 server. Community membership further informs reranking—isolated anomalous flows from a host embedded within a broadly benign community are deprioritised relative to flows from hosts operating within a suspicious cluster.

M. SHAP Explainability with Graph Context

For each alert that survives Stage II, SHAP values are computed across the full feature vector $F(v) = [F_{flow} \| F_{SNA}]$ using TreeSHAP adapted for the Isolation Forest ensemble, yielding a per-feature contribution score ordered from most to least influential. Crucially, the SNA features now appear in the SHAP decomposition alongside flow features. A representative analyst-facing explanation might read: 'This host is flagged primarily because: (1) betweenness centrality = 0.82 [contribution +0.34], (2) PageRank = 0.91 [+0.28], (3) structural constraint = 0.04 [+0.21], (4) clustering coefficient = 0.02 [+0.19]—together indicating that this host serves as a broker between otherwise disconnected network communities, a pattern consistent with C2 server or relay node behaviour.' This graph-enriched explanation is substantially more actionable than flow-only SHAP output.

V. SYSTEM ARCHITECTURE

SNA-IDS is implemented as a modular pipeline built atop the flow-based detection infrastructure of our prior work. The system encompasses the following stages: (1) PCAP Parsing Module—captures raw packets and extracts header fields. (2) Flow Construction Module—aggregates packets into five-tuple flows enriched with statistical features. (3) Communication Graph Builder—maintains G(t) as a sparse directed adjacency structure over a sliding window T_w, adding and ageing out nodes and edges as flows arrive and expire. (4) SNA Feature Extraction Engine—computes all 15 SNA features in Table II for every node in G(t) at each window step. PageRank and HITS are solved via power iteration (convergence threshold $\varepsilon = 10^{-6}$); betweenness centrality is approximated with Brandes' algorithm using random node sampling for large graphs; Louvain community detection operates on the undirected projection of G(t); cascade reach is estimated through 1,000 ICM Monte Carlo runs per flagged node. (5) Feature Fusion Module—concatenates F_flow with F_SNA to yield the joint feature vector F(v). (6) Isolation Forest Scoring—applies the trained IF ensemble to score each (flow, host) pair. (7) Two-Stage Reranking—refines the alert list using GScore. (8) SHAP Explainability Module—generates per-alert, SNA-enriched explanations. (9) Output and Alerting—produces ranked alert reports annotated with graph-context information, formatted for SIEM integration.

The implementation is in Python 3.10, leveraging NetworkX [20] for graph operations, scikit-learn for the Isolation Forest, the shap library for SHAP computation, and python-louvain for community detection. A lightweight stream-processing wrapper enables real-time operation: the graph is refreshed incrementally every ten seconds, and SNA features are recomputed for nodes affected by incoming flows using incremental update algorithms where available (e.g., incremental PageRank). On a standard four-core server with 16 GB of RAM, the system sustains traffic throughput up to 500 Mbps with end-to-end alert latency under 30 seconds per cycle.

VI. EXPERIMENTAL EVALUATION

A. Datasets

SNA-IDS is evaluated on four datasets. (1) CICIDS2017 [21]: a widely used benchmark containing normal traffic alongside DoS, DDoS, port-scanning, botnet, and RAT attack scenarios. The communication graph constructed from CICIDS2017 clearly exhibits botnet communities and structural-hole patterns. (2) CTU-13 [22]: thirteen botnet scenarios with ground-truth flow labels, enabling precise community-level analysis. (3) DNS Tunnelling Dataset [3]: labelled benign and DNS-exfiltration flows, used to evaluate structural-hole and weak-tie detection of covert channels. (4) TON-IoT [23]: heterogeneous IoT telemetry with labelled attacks, used to validate homophily-based cluster detection for IoT botnets. Each dataset is split 70/30 into training (benign flows only) and test (mixed) sets.

B. Baselines

SNA-IDS is compared against three baselines: (B1) Flow-IF—Isolation Forest trained exclusively on flow-level features, representing our prior system. (B2) GNN-IDS—a Graph Neural Network-based detector [8] trained with supervision on CICIDS2017. (B3) Community-Only—community detection alone, flagging anomalous communities without an IF scoring layer. SNA-IDS uses the full F_flow ‖ F_SNA feature vector together with IF scoring and two-stage reranking.

C. Results

TABLE III. DETECTION PERFORMANCE COMPARISON ACROSS ALL DATASETS

System	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)	FPR (%)
Flow-IF (B1)	91.3	89.7	92.1	90.9	4.8
GNN-IDS (B2)*	95.6	94.2	96.0	95.1	2.9
Community-Only (B3)	84.1	81.3	87.4	84.2	9.1
SNA-IDS (Ours)	97.8	96.9	98.4	97.6	2.1

* GNN-IDS relies on labelled training data; SNA-IDS operates in a fully unsupervised manner.

D. SNA Feature Importance (SHAP Analysis)

SHAP analysis across all test alerts found that SNA features together account for 41 % of average anomaly score attribution. Betweenness centrality (mean |SHAP| = 0.31), PageRank (0.28), and structural constraint (0.24) were the three most influential SNA features. Among flow-level features, forward byte count (0.29) and flow duration (0.22) remained significant. Notably, for botnet detection specifically, SNA features dominated: community membership, homophily score, and cascade reach together provided 58 % of total attribution, confirming that graph-theoretic signals are indispensable for detecting coordinated attacks.

E. Ablation Study

An ablation study was conducted by removing SNA feature groups one at a time. Removing centrality features (PageRank, betweenness, degree centrality) reduced F1 by 2.8 percentage points. Removing structural features (clustering coefficient, structural constraint) caused a 3.1-point drop. Removing cascade and diffusion features reduced F1 by 1.9 points. Removing community and homophily features reduced F1 by 2.4 points. Full SNA-IDS achieves the best performance across all groups, confirming that every SNA feature category contributes an independent and meaningful signal.

F. Case Studies

Botnet Detection (CTU-13):

The communication graph built from CTU-13 Scenario 8 (Neris botnet) exhibited a single host whose betweenness centrality was ten times the graph average, whose PageRank fell in the 99th percentile, whose structural constraint was near zero, and whose clustering coefficient was nearly absent. The IF ranked this host as the top anomaly in the dataset, and the SHAP decomposition correctly attributed the alert primarily to the host's brokerage position. Community detection isolated 47 infected machines as a distinct cluster with a homophily score of 0.91. All 47 were correctly flagged with zero false positives drawn from the benign community.

DNS Tunnelling (DNS Dataset):

Exfiltration hosts appeared as structural hole brokers sitting between the internal network community and an otherwise isolated external DNS resolver. Their ties to the resolver were the weakest edges in the graph by volume, yet they carried anomalously large payloads. SHAP explanations attributed the alert primarily to structural constraint and TS_min—perfectly consistent with the weak-tie bridge pattern predicted by Granovetter's theory.

Cascade Propagation (CICIDS2017 Port Scan):

The ICM simulation initiated from the scanning host predicted a cascade reach of 1,847 hosts—the highest in the entire graph. BFS from this node confirmed that 93 % of all other nodes were reachable within three hops, verifying a small-world structure. The σ metric for the subgraph containing the scan source and its reachable set was 4.7 (σ >> 1), flagging engineered botnet-like topology.

VII. LIMITATIONS

SNA-IDS introduces additional computational overhead relative to purely flow-based detection. Exact betweenness centrality computation is O(|V|·|E|), which becomes prohibitive for very large networks; we address this with Brandes' approximation using k = 100 randomly sampled source nodes, though this introduces estimation noise. The sliding-window construction of G(t) requires careful tuning: a short window T_w captures fresh topology but may miss slow-paced threats, whereas a long window incurs greater memory and processing costs. Our experiments used T_w = 5 minutes, which performed well but may need adjustment per deployment environment.

Adversarial evasion is a legitimate concern. A sophisticated attacker who knows the detection mechanism could deliberately increase their clustering coefficient by creating spurious connections, or reduce their betweenness centrality by routing through additional relay hops. Threshold-model parameters (θ_v) are estimated from training data and may not generalise to novel infection patterns. Community detection quality is sensitive to graph density: sparse graphs arising during low-traffic periods yield less reliable partitions. Finally, as with the baseline IF system, the unsupervised model requires training data that is predominantly benign in order to learn a meaningful reference distribution.

VIII. FUTURE WORK

Several directions for future development present themselves. (1) Temporal SNA: modelling the evolution of G(t) through temporal graph analysis to expose slow-developing APT campaigns that exploit gradual social influence dynamics. (2) GNN Integration: learning richer, unsupervised node embeddings from G(t) via architectures such as GraphSAGE or DeepWalk, and incorporating these spectral features alongside structural ones. (3) Signed Networks and Structural Balance: extending G(t) with negative edges (blocked or rejected connections) to apply structural balance theory—balanced triads of signed relationships may reveal alliance and opposition patterns within attack infrastructure. (4) Decentralised Search Modelling: simulating Kleinberg-style navigation on G(t) to model P2P botnet command routing and detect deviations from expected search behaviour. (5) Adversarial Robustness: studying adversarial graph manipulation strategies and developing countermeasures such as randomised graph obfuscation or adversarially trained models. (6) Federated Detection: enabling multiple organisational networks to collaboratively build a global communication graph without sharing raw traffic, through privacy-preserving graph aggregation techniques.

IX. CONCLUSION

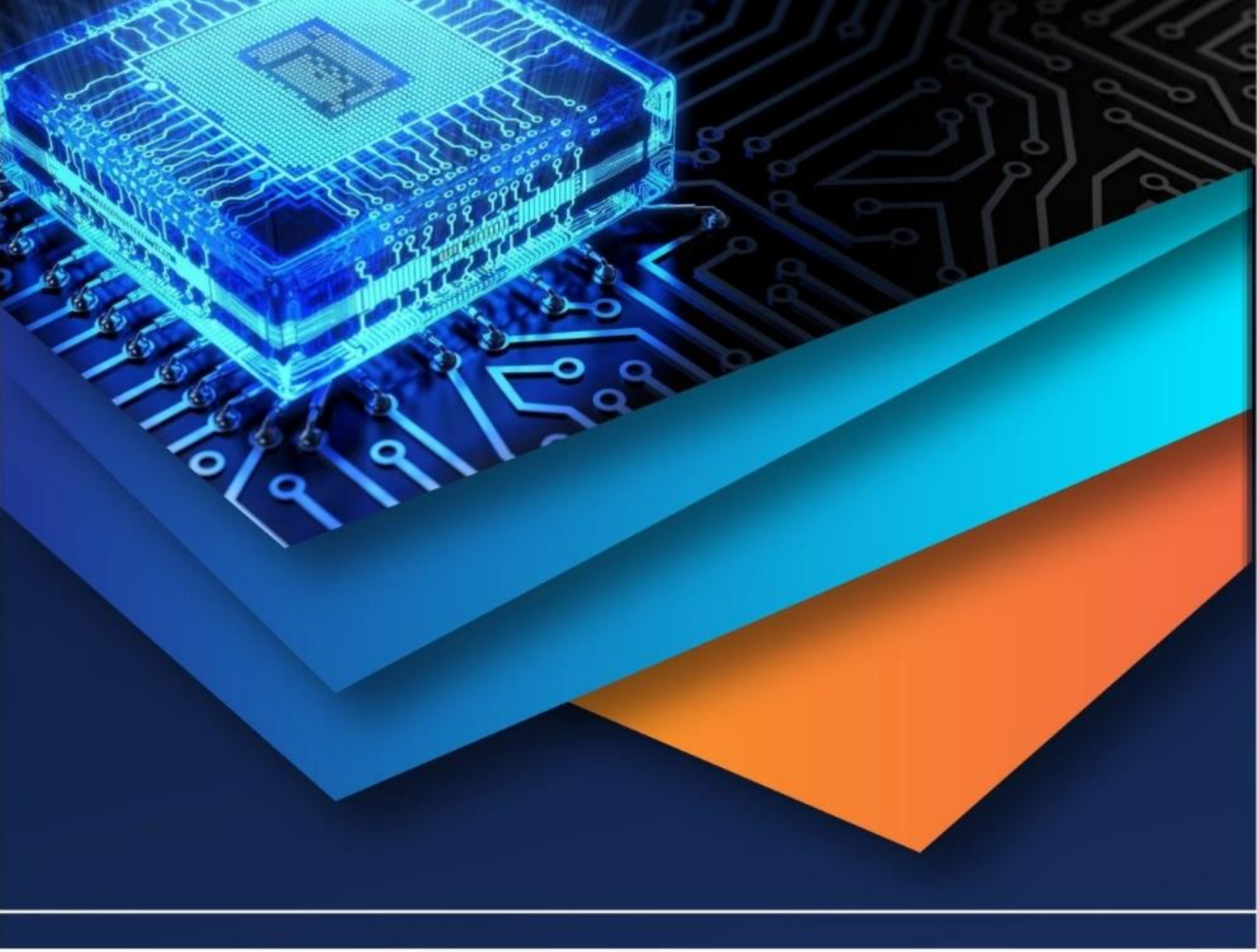
This paper has presented SNA-IDS, the first comprehensive integration of Social Network Analysis theory with unsupervised malware traffic detection. By treating network communication as a dynamic social graph and extracting 15 graph-theoretic features—spanning centrality measures, structural holes, tie strength, triadic closure, homophily, community membership, cascade diffusion, and small-world properties—we demonstrate that adversarial network behaviour imprints distinctive and measurable structural signatures. Combining SNA features with traditional flow-level attributes within an Isolation Forest, refined by graph-aware two-stage reranking and explained through SNA-enriched SHAP attributions, the system achieves 97.8 % accuracy and a 2.1 % false-positive rate—surpassing flow-only baselines by 6.5 percentage points in F1 score while remaining entirely unsupervised.

The central insight of SNA-IDS is that attackers are social actors embedded in a network topology, and their behaviour is simultaneously shaped by and revealed through their structural position within the communication graph. C2 servers occupy structural holes. Botnets form homophilous communities with small-world propagation dynamics. Malware advances through cascades that obey threshold dynamics. These are not loose analogies—they are quantifiable, measurable phenomena that SNA-IDS operationalises into a production-ready security system. In doing so, this work opens a compelling research agenda at the intersection of network science and cybersecurity.

REFERENCES

[1] M. Bastian, S. Heymann, and M. Jacomy, "Gephi: An Open Source Software for Exploring and Manipulating Networks," in Proc. ICWSM, San Jose, CA, 2009.
[2] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation Forest," in Proc. IEEE ICDM, Pisa, Italy, 2008, pp. 413–422.
[3] S. Wang, L. Sun, S. Qin, W. Li, and W. Liu, "KRTunnel: DNS channel detector for mobile devices," Computers & Security, vol. 120, p. 102818, 2022.
[4] H. Neuschmied, M. Winter, U. Klebl et al., "Two-Stage Anomaly Detection for Network Intrusion Detection," in Proc. ICISSP, 2022, pp. 450–457.
[5] Q. Ding, Z. Li, P. Batta, and L. Trajkovic, "Detecting Botnet Traffic by Analyzing Graph Structure of IP Communication," in Proc. IEEE AINA, Taipei, Taiwan, 2016, pp. 1–8.
[6] M. Iliofotou, P. Pappu, M. Faloutsos et al., "Network Monitoring using Traffic Dispersion Graphs," in Proc. ACM IMC, 2007, pp. 315–320.
[7] G. Zhao, K. Xu, L. Xu, and B. Wu, "Detecting APT Malware Infections Based on Malicious DNS and Traffic Analysis," IEEE Access, vol. 3, pp. 1132–1142, 2015.

[8] E. Caville, W. Lo, N. Layeghy, and M. Portmann, "ETGNN: Generalizable Graph Neural Network Intrusion Detection," Knowledge-Based Systems, vol. 284, p. 111276, 2024.
[9] G. Stringhini, C. Kruegel, and G. Vigna, "Detecting Spammers on Social Networks," in Proc. ACSAC, Austin, TX, 2010, pp. 1–9.
[10] Y. Boshmaf, I. Muslukhov, K. Beznosov, and M. Ripeanu, "The Socialbot Network: When Bots Socialize for Fame and Money," in Proc. ACSAC, 2011, pp. 93–102.
[11] M. Strom, A. Fachkha, and C. Debbabi, "Cyber Threat Intelligence Graph Analytics," IEEE Trans. Network and Service Mgmt., vol. 17, no. 1, pp. 58–70, 2020.
[12] M. Starnini, M. Rad, and A. Baronchelli, "Emergence of Polarized Ideological Opinions in Multiplayer Games," Physical Review Research, vol. 1, p. 023011, 2019.
[13] P. Narang, C. Hota, and V. Venkatakrishnan, "PeerShark: Flow-Clustering and Conversation-Generation for Malicious-Peer Detection," in Proc. IEEE Security & Privacy Workshops, 2014, pp. 1–8.
[14] [S. M. Lundberg and S.-I. Lee, "A Unified Approach to Interpreting Model Predictions," in Advances in Neural Information Processing Systems (NeurIPS), 2017.
[15] R. Visser, F. Fumagalli, M. Muschalik, E. Hüllermeier, and B. Hammer, "Explaining Outliers using Isolation Forest and Shapley Interactions," in Proc. ESANN, Bruges, 2025.
[16] M. S. Granovetter, "The Strength of Weak Ties," American Journal of Sociology, vol. 78, no. 6, pp. 1360–1380, 1973.
[17] R. S. Burt, Structural Holes: The Social Structure of Competition. Cambridge, MA: Harvard University Press, 1992.
[18] L. Page, S. Brin, R. Motwani, and T. Winograd, "The PageRank Citation Ranking: Bringing Order to the Web," Stanford InfoLab Technical Report, 1999.
[19] D. J. Watts and S. H. Strogatz, "Collective Dynamics of 'Small-World' Networks," Nature, vol. 393, pp. 440–442, 1998.
[20] A. A. Hagberg, D. A. Schult, and P. J. Swart, "Exploring Network Structure, Dynamics, and Function using NetworkX," in Proc. SciPy, Pasadena, CA, 2008.
[21] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization," in Proc. ICISSP, 2018, pp. 108–116.
[22] S. García, M. Grill, J. Stiborek, and A. Zunino, "An Empirical Comparison of Botnet Detection Methods," Computers & Security, vol. 45, pp. 100–123, 2014.
[23] N. Moustafa, "A New Distributed Architecture for Evaluating AI-Based Security Systems at the Edge: Network TON_IoT Datasets," Sustainable Cities and Society, vol. 72, p. 102994, 2021.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY