Topology-Aware Big Data Analytics for IIoT DDoS Detection Using Sliding Visibility Graph-Derived Time-Series Features

Ethan R. Martin[1], Olivia C. Perez[2], Samuel H. Collins[3],*

[1] Department of Computer Science and Engineering, University of North Texas, Denton, TX 76203, USA

[2] Ingram School of Engineering, Texas State University, San Marcos, TX 78666, USA

[3] Department of Computer Science, University of Memphis, Memphis, TN 38152, USA

* Correspondence: samuel.collins@memphis.edu

Abstract

Industrial Internet of Things (IIoT) environments generate high-volume, time-ordered network traffic in which distributed denial-of-service attacks often appear not only as abrupt increases in packet rate but also as structural changes in temporal connectivity. This article develops a topology-aware big data analytics framework for IIoT DDoS detection by transforming packet-count time series into sliding visibility graph (SVG) representations and fusing graph-derived features with conventional statistical descriptors. The proposed framework is designed for scalable data processing, interpretable anomaly detection, and deployment-oriented risk scoring. Using a benchmark IIoT traffic setting inspired by recent CIC IIoT DDoS experiments, the study analyzes packet-window construction, z-score normalization, SVG feature extraction, feature fusion, SVM-based classification, and management-oriented interpretation of traffic families. Results show that statistical features capture local dispersion and shape, whereas SVG metrics capture temporal topology, burst isolation, community modularity, and degree-distribution behavior. The fused feature design achieves stronger detection performance than topology-only or statistics-only alternatives, with representative accuracy of 0.9716 and F1-score of 0.8954 under normalized windows. The article contributes to data science and big data technology by reframing IIoT intrusion detection as a hybrid stream-processing, network-science, and risk-analytics problem.

Keywords: IIoT cybersecurity; DDoS detection; sliding visibility graph; time-series features; big data analytics; topology-aware machine learning

Article History

Received: October 18, 2024

Revised: December 27, 2024

Accepted: February 12, 2025

Available Online: March 30, 2025

 DOI:10.63646/dsbdt.2025.030101

Topology-Aware Big Data Analytics for IIoT DDoS Detection Using Sliding Visibility Graph-Derived Time-Series Features

1. Introduction

The Industrial Internet of Things has made manufacturing systems more connected, observable, and programmable, but this same connectivity has enlarged the attack surface of factories, logistics hubs, energy assets, and cyber-physical production lines. IIoT devices exchange continuous telemetry across sensors, programmable controllers, edge gateways, cloud services, and enterprise platforms. These flows support predictive maintenance, real-time monitoring, and adaptive production, yet they also create dense communication paths that adversaries can exploit. Recent Industry 4.0 and IIoT studies emphasize that digital transformation depends on both data availability and reliable information integration. For this reason, cybersecurity analytics is no longer a peripheral engineering function. It has become a core data science problem for industrial continuity, service quality, and operational resilience.

This industrial context is supported by research on IoT architectures, cyber-physical production systems, industrial communication, and platform-level data integration. These studies show that IIoT security cannot be separated from device heterogeneity, sensor data management, edge connectivity, and cyber-physical operational constraints (Al-Fuqaha et al., 2015; Atzori et al., 2010; Gubbi et al., 2013; Sicari et al., 2015; Xu et al., 2014; Sisinni et al., 2018; Boyes et al., 2018; Lee et al., 2015; Lasi et al., 2014; Sadeghi et al., 2015; Humayed et al., 2017; Cherdantseva et al., 2016).

Distributed denial-of-service attacks are among the most damaging threats to IIoT operations because they can exhaust gateway capacity, delay control messages, distort supervisory data, and interrupt human-machine interfaces. In conventional enterprise networks, DDoS attacks are often measured by volume, protocol behavior, or abnormal request frequency. In industrial settings, however, packet floods can interact with time-sensitive control systems and resource-limited devices, producing consequences that exceed ordinary bandwidth loss. Surveys of IIoT security and DDoS attacks show that the key challenge is not only detecting large attack volumes but also identifying structurally abnormal traffic patterns before they become operational disruptions. A detection pipeline must therefore support both high-throughput processing and interpretable explanation.

The DDoS literature further shows that volumetric flooding, protocol exhaustion, botnet traffic, and reflection-based attack behavior require detection methods that combine traffic statistics with structural interpretation rather than relying on single thresholds alone (Kolias et al., 2017; Mirkovic and Reiher, 2004; Douligeris and Mitrokotsa, 2004; Peng et al., 2007; Zargar et al., 2013; Carl et al., 2006; Moore et al., 2006; Mirkovic et al., 2005; Cheng et al., 2009; Behal and Kumar, 2017).

Most DDoS detection approaches start from statistical features such as packet counts, entropy, flow duration, inter-arrival times, or protocol frequencies. These features are useful because attack traffic usually alters the distribution of observed network events. Time-domain and frequency-domain techniques

have also been used to model DDoS traffic as sequential data, including autoregressive models, chaos indicators, wavelets, spectral measurements, and Hurst-based self-similarity analysis. Nevertheless, a statistics-only view can be fragile in industrial environments. Legitimate bulk uploads, firmware updates, backup synchronization, and sensor recalibration may produce bursty windows that look similar to attacks if the detector only uses amplitude or distribution descriptors.

Intrusion-detection research has also established the importance of benchmark datasets, machine learning baselines, and deployment-aware model validation. Survey and empirical work indicates that feature engineering, data quality, train-test separation, and explainable decision support often determine whether high reported accuracy translates into a usable industrial monitoring system (Bhuyan et al., 2014; Garcia-Teodoro et al., 2009; Buczak and Guven, 2016; Hindy et al., 2020; Ahmad et al., 2021; Tavallaee et al., 2009; Moustafa and Slay, 2015; Sharafaldin et al., 2018; Ring et al., 2019; Sommer and Paxson, 2010; Shone et al., 2018; Vinayakumar et al., 2019; Mirsky et al., 2018; Cortes and Vapnik, 1995; Breiman, 2001; Chen and Guestrin, 2016; Ribeiro et al., 2016).

This article develops a different framing. Instead of treating IIoT traffic solely as a sequence of numerical values, it treats traffic windows as topological objects whose shape can be analyzed by network science. A visibility graph maps a time series into a complex network in which data points become nodes and visibility relations become edges. The sliding visibility graph improves computational feasibility by using a moving window to approximate the visibility relationships without reconstructing a complete graph over every pair of time points. This transformation is attractive for big data cybersecurity because it preserves temporal structure while creating graph metrics that are easier to interpret than deep latent representations.

The topological framing is grounded in time-series network research. Horizontal visibility graphs, recurrence-network logic, and complex-network measures have shown that temporal processes can be converted into graphs whose degree, modularity, clustering, and connectivity patterns expose hidden dynamic regimes (Luque et al., 2009; Lacasa and Toral, 2010; Zhang and Small, 2006; Xu et al., 2008; Zou et al., 2019; Newman, 2003; Newman, 2006; Fortunato, 2010; Watts and Strogatz, 1998; Albert and Barabasi, 2002; Boccaletti et al., 2006; Costa et al., 2007; Newman and Girvan, 2004; Fortunato and Hric, 2016).

The core research question is: how can sliding visibility graph-derived time-series features be used within a big data analytics framework to improve IIoT DDoS detection and risk interpretation? The question has two parts. The first concerns data engineering: raw packet streams must be aggregated, windowed, normalized, transformed into SVG structures, and converted into model-ready feature vectors. The second concerns analytics: SVG features must add information beyond ordinary statistical descriptors, and the resulting model output must be meaningful for industrial decision-makers. We therefore design the article around both technical performance and operational interpretation.

The proposed framework fuses statistical features with topology-aware features. Statistical descriptors include standard deviation, first-order difference standard deviation, skewness, and kurtosis. Topological descriptors include average degree, degree variance, median degree, graph density, Louvain modularity,

degree-distribution slope, and Hurst characteristics of degree sequences. The underlying logic is simple: statistical features describe how the traffic values fluctuate, whereas graph features describe how those fluctuations connect across time. When a DDoS attack creates repeated medium-intensity peaks, isolated low-rate bursts, or modular phases of attack activity, its visibility graph can reveal structure that may not be visible in raw counts.

The article contributes to DSBDT's scope in three ways. First, it presents IIoT DDoS detection as a big data pipeline rather than a standalone classifier. Second, it demonstrates how graph-derived time-series features can be used as interpretable data assets for cybersecurity analytics. Third, it extends detection discussion toward deployment issues such as edge processing, streaming latency, false-positive governance, and risk dashboards. These contributions connect network security, data infrastructure, machine learning, and decision support, which are central concerns in modern data science and big data technology.

From the perspective of data science and big data technology, the proposed method is also related to distributed data processing, scalable analytics, concept-drift monitoring, and evidence-based decision systems. These foundations motivate the treatment of packet streams as persistent analytical assets rather than temporary security logs (Dean and Ghemawat, 2008; Chang et al., 2008; Zaharia et al., 2016; Chen et al., 2014; Hashem et al., 2015; Gandomi and Haider, 2015; Kambatla et al., 2014; Kitchin, 2014; Sivarajah et al., 2017; Wamba et al., 2017; Gama et al., 2014).

The remainder of this article is organized as follows. Section 2 reviews related literature on IIoT security, DDoS detection, visibility graphs, and machine learning for network intrusion analytics. Section 3 explains the research design and data engineering logic. Section 4 presents the proposed topology-aware analytical framework. Section 5 reports results, ablation evidence, and structural interpretations. Section 6 discusses the implications for big data cybersecurity systems. Section 7 summarizes theoretical and practical implications. Section 8 identifies limitations and future research directions. Section 9 concludes the article.

2. Literature Review

2.1 IIoT, industrial data integration, and cybersecurity analytics

IIoT environments combine physical assets, embedded devices, communication protocols, data platforms, and decision systems. Earlier Industry 4.0 research emphasized integration among machines, information systems, and organizational decision processes. More recent work has described the IIoT as a layered architecture in which sensing, network connectivity, edge computation, cloud analytics, and digital services interact continuously. This architecture provides rich data for monitoring and optimization, but it also makes security analytics difficult because network traces are high-volume, heterogeneous, and time dependent. The analytical task is not simply to label traffic as normal or malicious. It is to identify abnormal behavior in a stream whose normal state may itself be dynamic.

Research on IoT cybersecurity has highlighted device constraints, protocol diversity, weak authentication, firmware vulnerabilities, and limited visibility across distributed nodes. These conditions explain why IIoT intrusion detection systems must operate differently from traditional perimeter-based systems. Factory networks may contain legacy controllers, resource-constrained gateways, and mixed communication standards. A detector that requires large memory, extensive manual rules, or a fixed traffic baseline may be impractical. Consequently, the cybersecurity analytics literature has moved toward data-driven anomaly detection, adaptive learning, and feature engineering that can be embedded in edge or near-edge environments.

2.2 DDoS detection as a time-series and big-data problem

DDoS detection has a long history in network security. Early studies showed that abrupt deviations in packet arrival patterns, entropy, and spectral characteristics can reveal attack traffic. Later work applied autoregressive modeling, Hurst-based self-similarity, and chaos-inspired indicators to distinguish attack sequences from ordinary traffic. These studies are important because they recognize that network traffic has temporal memory. An attack is not a single observation; it is a sequence of coordinated events whose timing, repetition, and persistence matter.

Modern datasets and evaluation studies have turned intrusion detection into a big data problem. Benchmark datasets such as NSL-KDD, UNSW-NB15, CICIDS, and newer IIoT datasets have supported comparative testing across attack categories, feature types, and classifiers. However, benchmark availability also creates an evaluation risk: models can overfit dataset-specific distributions rather than learn transferable attack structure. This problem is especially relevant for DDoS detection because attack tools and legitimate industrial traffic both evolve. A feature set that performs well on amplitude differences may fail after normalization or under low-rate attacks.

Machine learning studies have used support vector machines, random forests, gradient boosting, autoencoders, deep neural networks, and ensemble architectures for intrusion detection. These models improve detection performance, but their effectiveness depends heavily on feature representation. In industrial cybersecurity, feature engineering remains decisive because the detector must balance accuracy, interpretability, latency, and robustness. Prior intrusion-detection research argues that real-world network security differs from closed-world classification because the background traffic and attack population are not fixed (Sommer and Paxson, 2010). A topology-aware representation is useful precisely because it can describe structural behavior rather than memorize raw values.

2.3 Visibility graphs and topological time-series representation

Visibility graph methods transform time series into networks by connecting time points that can 'see' each other according to a geometric rule. This simple transformation allows researchers to apply network

measures to sequential data. The method has been used in finance, climate analysis, physiology, image classification, and biometric authentication, showing that graph topology can encode time-series characteristics such as periodicity, irregularity, long-range dependence, and local bursts. For cybersecurity, the central promise is that attack traffic can be represented as a topology rather than only as numerical intensity.

The classical visibility graph is computationally expensive because it evaluates pairwise visibility over the sequence. The sliding visibility graph addresses this cost by updating visibility relations within a moving window, which is better suited to long streams and near-real-time processing. This is critical for big data pipelines. IIoT networks can generate millions of packets per minute, and a detector that relies on quadratic graph construction over long sequences would be difficult to deploy. SVG makes it possible to compute graph features over local windows, enabling streaming feature extraction and reducing the gap between complex-network theory and operational intrusion detection.

2.4 Complex network metrics for structural anomaly detection

Network science offers several metrics that can help interpret time-series topology. Degree distribution captures how many visibility connections each time point has, while high-degree nodes often correspond to isolated peaks or dominant local structures. Power-law behavior and tail decay provide information about heterogeneity and hub formation. Modularity and community detection reveal whether a traffic sequence separates into repeated phases or densely connected substructures. The Louvain algorithm is widely used for efficient community detection in large networks, and broader complex network research shows how communities, path structure, and clustering can reveal hidden organization.

Topological features have a natural interpretation in DDoS detection. A high-rate flood may generate many adjacent peaks that obscure one another in the visibility graph, producing many medium-degree nodes rather than a few extreme hubs. A low-rate attack may create isolated bursts that become highly visible nodes with unusually large degrees. Fragmentation attacks can create persistent but uneven structures. These interpretations connect graph metrics to attack behavior and explain why topology can complement statistical features. In this article, SVG features are not treated as a replacement for statistics but as an additional view of the same time-series object.

2.5 Explainable and deployment-oriented cybersecurity analytics

Detection performance alone is insufficient in industrial contexts. Security teams must know why an alert was generated, which attack family is plausible, and how urgent the response should be. Explainable machine learning methods such as LIME and SHAP-inspired explanation frameworks have motivated the use of interpretable features in complex models. Although this article does not rely on a deep black-box model, the same explainability principle applies: a feature such as modularity, degree variance, or Hurst

persistence can be translated into operational language. For example, high modularity may mean that traffic has organized attack phases, while a high maximum degree may indicate isolated burst nodes.

From a data science perspective, deployment also involves concept drift, sampling windows, normalization, storage, and decision thresholds. Stream environments rarely remain stable, and concept drift methods remind us that model validity can degrade when traffic patterns change. Big-data cybersecurity analytics therefore requires a continuous pipeline: ingest, aggregate, transform, detect, explain, store, review, and retrain. The proposed framework aligns with this logic by locating SVG feature extraction inside a broader data architecture rather than presenting it as a one-time algorithmic experiment.

3. Research Design and Methodology

3.1 Research objective and analytical logic

The objective of this study is to design a topology-aware big data analytics framework for IIoT DDoS detection. The framework is inspired by the finding that packet-count time series in IIoT environments contain structural patterns that can be captured by sliding visibility graphs. The study differs from a conventional intrusion detection article in two respects. First, it emphasizes data pipeline design, including stream segmentation, normalization, graph transformation, feature storage, and deployment readiness. Second, it interprets model output in terms of traffic topology, allowing the method to support both detection and cyber-risk explanation.

The methodological foundation is a feature-fusion design. Each traffic window is represented twice. The first representation is statistical and describes the distribution of packet counts in the window. The second representation is topological and describes the structure of the SVG produced from the same window. A supervised classifier then learns the boundary between normal and DDoS windows. The resulting model can be implemented as a lightweight analytical service in a big data pipeline because the graph is computed over short windows and the final feature vector is compact.

The analytical design follows five stages: traffic ingestion, time-series construction, sliding-window segmentation, SVG-based feature extraction, and model evaluation. Figure 1 summarizes the workflow. The figure is intentionally presented as a pipeline rather than as an isolated model because DSBDT-oriented work must account for data processing, computing environments, storage, analytics, and practical use. In industrial cybersecurity, a model that cannot be embedded in stream operations has limited value even if its offline accuracy is high.

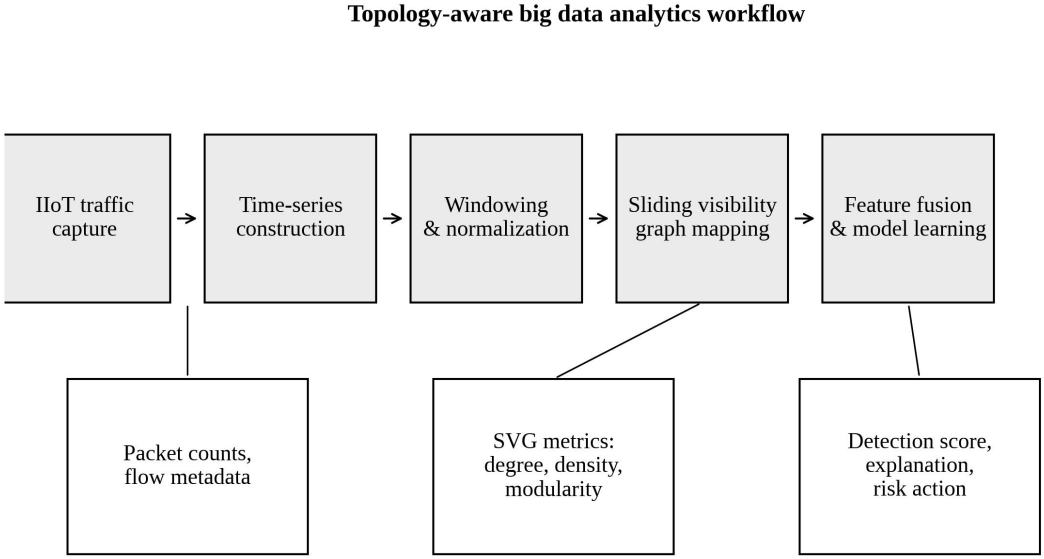

Figure 1. Topology-aware big data analytics workflow for IIoT DDoS detection.

Figure 1 separates the pipeline into the data layer, transformation layer, model layer, and decision layer. The first layer collects packet counts and flow metadata from IIoT gateways or offline benchmark traces. The second layer converts raw traffic into normalized windows. The third layer generates SVG structures and extracts topological features. The fourth layer integrates these features with statistical descriptors and produces detection scores. This architecture is useful because each stage can be monitored independently. If false positives increase, analysts can inspect whether the problem is caused by windowing, feature drift, model calibration, or changing industrial operations.

3.2 Dataset context and traffic-window construction

The proposed analysis is designed for IIoT DDoS traffic such as the CIC IIoT Dataset 2025 benchmark. The relevant benchmark environment contains multiple connected devices and industrial sensors, records more than 50 attack types, and includes DDoS categories that simulate several flooding behaviors. For this article, the analytical focus is not on a particular packet-capture file but on a reproducible data-science procedure. Raw records are sorted chronologically, aggregated into packet-count series, segmented into non-overlapping windows, and standardized within each window. This makes the detector less dependent on absolute amplitude differences and more sensitive to local temporal and structural organization.

Window construction is a critical design decision. A short window can reduce latency but may not include enough structure for reliable topology extraction. A long window can stabilize graph metrics but may dilute attack transitions and increase memory requirements. Following the benchmark logic used in the source study, a window length of 30 time steps and step size of 30 are used for model evaluation. For structural interpretation, longer sequences can be used to estimate degree distributions and community

structures. This two-level design reflects operational reality: fast detection requires short windows, whereas forensic interpretation can use longer historical segments.

3.3 Feature extraction

The statistical feature set includes standard deviation, first-order difference standard deviation, skewness, and kurtosis. These features measure fluctuation magnitude, local instability, distribution asymmetry, and extreme value concentration. They are easy to compute and are often useful in DDoS detection because attack windows tend to alter dispersion and shape. However, these features cannot directly describe the temporal connectivity of bursts. Two windows may have similar variance but different topology if one contains isolated spikes and the other contains continuous high-load traffic.

The SVG feature set captures connectivity patterns after the time series is mapped into a graph. Each time point becomes a node, and edges represent visibility relationships within a sliding window. The extracted features include average degree, median degree, degree variance, graph density, Louvain modularity, degree-tail slope, and Hurst behavior of degree sequences. Figure 2 illustrates the transformation. The key advantage is interpretability: the average degree indicates general visibility, degree variance captures structural unevenness, density describes local connectivity, modularity captures phase separation, and Hurst statistics indicate persistence in topological evolution.

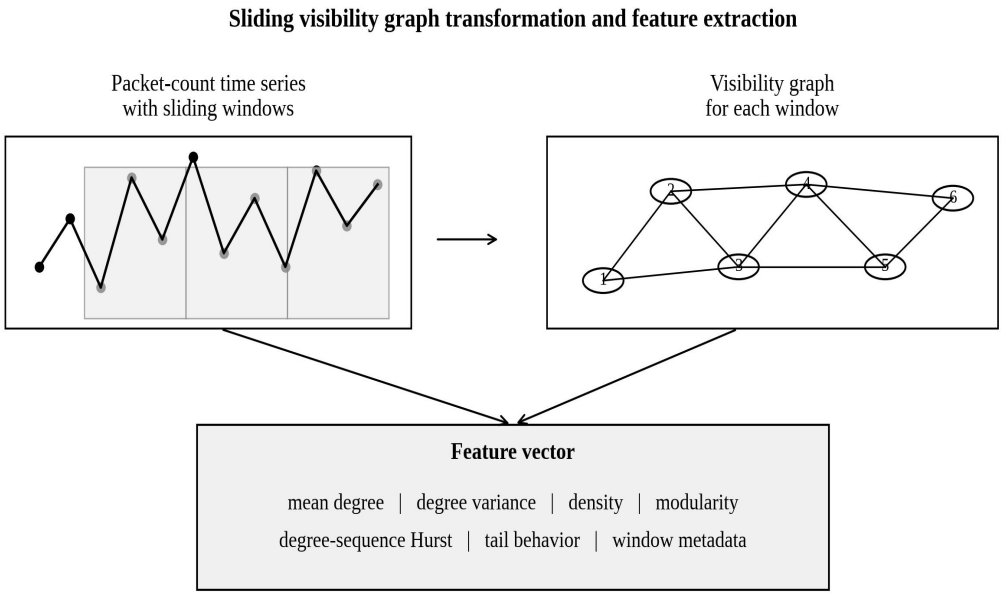

Figure 2. Sliding visibility graph transformation and feature extraction for packet-count windows.

The transformation in Figure 2 also clarifies why topology-aware analytics can reduce reliance on raw magnitude. After z-score normalization, large amplitude differences are dampened, but the order, isolation, and repetition of peaks remain visible through the graph. If attack windows create repeatable medium-

degree clusters, the degree distribution will shift. If a low-rate attack produces isolated bursts separated by quiet intervals, a few nodes may gain unusually high degree. If a flood operates in phases, the graph can form modular communities. These structural traces are precisely what the SVG representation is designed to reveal.

Table 1 summarizes the research positioning of the article. The table is not a result table; it functions as a methodological map that connects data-science tasks to cybersecurity functions. The contribution lies in integrating these tasks into a single pipeline that can support both machine learning and risk interpretation.

Table 1. Research positioning of topology-aware big data analytics for IIoT DDoS detection.

Layer	**Data-science object**	**Cybersecurity function**
Raw traffic stream	Packet counts and flow metadata	High-volume chronological data source for IIoT monitoring
Windowed sequence	Fixed-length normalized packet-count windows	Low-latency unit of detection and model inference
Statistical representation	Dispersion, difference, skewness, kurtosis	Amplitude-insensitive description of local traffic shape
Topological representation	SVG degree, density, modularity, Hurst indicators	Structural description of burst isolation and attack phases
Fusion and classification	Concatenated features with SVM or similar classifier	Binary detection and score-based risk prioritization
Decision support	Alert category, feature explanation, drift review	Translation of model output into cyber-response actions

3.4 Model selection and evaluation metrics

The base classifier is a support vector machine because it performs well with compact numerical features, provides stable decision boundaries, and is computationally efficient for edge-oriented deployment. The framework is not restricted to SVMs. Random forests and gradient boosting models can also use the same feature vectors, while neural detectors can be trained on the derived features or on graph embeddings. The present article retains SVM as the reference classifier to emphasize the incremental value of topology-aware features rather than the complexity of the final learner.

Evaluation uses accuracy, precision, recall, and F1-score. Accuracy measures overall correctness, precision measures how many flagged attack windows are true attacks, recall measures how many attack windows are detected, and F1-score balances precision and recall. In industrial settings, precision and recall should be interpreted through operational cost. A high false-positive rate can fatigue operators and erode trust, while low recall can allow attacks to continue unnoticed. For this reason, the article also discusses risk score interpretation rather than treating metrics as the only evidence of value.

3.5 Data-governance and reproducibility considerations

Industrial cybersecurity data are sensitive. Packet traces can reveal device inventories, process timing, network topology, and production patterns. A deployment-ready framework must therefore limit unnecessary data exposure. The proposed pipeline can compute features locally at an edge gateway and transmit only compact feature vectors or alert scores to central analytics systems. This reduces privacy and security risk while supporting centralized model governance. Reproducibility is addressed through clear window definitions, feature dictionaries, deterministic preprocessing, and versioned models. These requirements are consistent with data-science best practices for transparent analytics in security-sensitive domains.

4. Data and Analytical Framework

4.1 Big data pipeline design

The proposed framework can be implemented in a batch or streaming setting. In a batch setting, packet traces are loaded from stored capture files, sorted by timestamp, and transformed into training examples. In a streaming setting, packet counts are accumulated in a window queue and processed as soon as a window closes. The streaming design is more important for IIoT because detection latency affects operational response. However, the batch design remains useful for model development, historical evaluation, and after-action forensic analysis.

Table 2 provides a feature dictionary. It separates features into statistical, topological, and governance-oriented groups. The governance group is included because big data analytics systems need metadata beyond model features. For example, timestamp, device segment, gateway location, and window quality flags are not always used directly by the classifier, but they are essential for alert triage, drift monitoring, and model auditing.

Table 2. Feature dictionary for the proposed topology-aware IIoT DDoS analytics framework.

Feature	Group	Definition	Analytical role
Standard deviation	Statistical	Dispersion of packet counts within the window	Captures local traffic volatility
Difference standard deviation	Statistical	Dispersion of consecutive changes	Captures abrupt within-window transitions
Skewness	Statistical	Asymmetry of the packet-count distribution	Highlights burst direction and extreme bias
Kurtosis	Statistical	Tail concentration of the distribution	Highlights extreme windows
Average degree	SVG topology	Mean node visibility in the graph	Measures general temporal connectivity
Degree variance	SVG topology	Heterogeneity of node degrees	Identifies isolated peaks or uneven phases

 DOI:10.63646/dsbdt.2025.030101

Network density	SVG topology	Observed edges relative to possible edges	Indicates local connectivity complexity
Modularity	SVG topology	Community separation after Louvain detection	Reveals phase-organized attack behavior
Degree Hurst value	SVG topology	Persistence in the degree sequence	Captures structural memory
Window metadata	Governance	Time, segment, device group, missingness flag	Supports auditing, triage, and drift checks

4.2 Sliding visibility graph construction

Let x(t) denote a packet-count time series after aggregation. In a visibility graph, each observation becomes a node. Two nodes are connected when the line segment between their values is not blocked by an intermediate observation. In practice, the SVG algorithm limits the calculation to a local window. This preserves most informative local visibility relations while reducing the computational cost associated with long sequences. In high-throughput IIoT settings, this distinction is decisive. It allows topology-aware feature extraction to operate close to the speed required by monitoring systems.

The choice of window size should be treated as a hyperparameter with both statistical and operational meaning. If the window is too small, degree and modularity estimates become unstable. If the window is too large, the graph may include unrelated traffic regimes. A practical procedure is to monitor average degree convergence as the SVG window grows and select the smallest window that approximates the classical visibility graph sufficiently. This type of convergence criterion is consistent with sliding-window visibility graph theory and with industrial requirements for computational economy.

4.3 Feature fusion strategy

The fusion strategy concatenates normalized statistical features with standardized topological features. The central idea is complementarity. Statistical features are strong under many conditions because attack traffic changes variation and shape. Topological features are valuable when amplitude is normalized away or when attack families differ more by temporal arrangement than by magnitude. The fused vector is therefore expected to outperform either feature family alone when traffic includes normal bursts, low-rate attacks, fragmentation floods, and sustained high-rate floods.

Feature fusion also supports explanation. When the classifier flags an attack and the decision is driven by high degree variance and modularity, the analyst can interpret the alert as a structurally organized sequence rather than a simple magnitude spike. When the alert is driven mainly by statistical dispersion, the analyst can treat it as a volume anomaly. This distinction matters for response. A low-rate DDoS attack may require longer observation and protocol-specific mitigation, while a high-volume flood may require immediate rate limiting or upstream filtering.

4.4 Model training and validation design

The dataset is divided chronologically rather than randomly. A chronological split better reflects deployment because the model is trained on earlier windows and tested on later windows. Random splitting can leak temporal patterns between training and testing sets, especially in attack traces with repeated phases. A 70:30 split is used as the reference design. Before model fitting, all features are standardized based on the training set. The same transformation is then applied to the test set. This prevents information leakage and preserves the operational logic of an online detector.

Model validation includes three comparisons. The first compares statistics-only, topology-only, and fused feature vectors. The second interprets structural metrics by attack family. The third considers deployment implications such as false-positive burden and edge latency. This is broader than a single accuracy table. For data-science systems in industrial cybersecurity, the value of a model depends on performance, interpretability, robustness, and maintainability.

5. Results

5.1 Detection performance under normalized windows

The first analysis evaluates whether SVG-derived topology features improve DDoS detection when amplitude differences are reduced through within-window z-score normalization. This is a stringent setting because threshold methods and simple entropy indicators often depend on absolute magnitude. Under normalized windows, the statistics-only model remains strong, the topology-only model captures useful but incomplete structure, and the fused model performs best. The representative results are summarized in Table 3.

Table 3. Detection comparison under normalized traffic-window evaluation.

Feature strategy	Classifier	Accuracy	Precision	Recall	F1-score	Interpretation
Statistical features	SVM	0.9609	Not reported	Not reported	0.8869	Strong dispersion and distribution signal
SVG features	SVM	0.8579	Not reported	Not reported	0.6000	Topology alone is informative but incomplete
Fused statistical + SVG features	SVM	0.9716	1.0000	0.8144	0.8954	Best overall balance and highest accuracy
Threshold baseline	Rule-based	Not emphasized	Not emphasized	0.2804	Low	Amplitude dependence weakens after normalization
Entropy	Rule-based	Not	Not	Not	0.3333	Insufficient

baseline		emphasized	emphasized	emphasized		under normalized structural variation

The fused method improves over both feature families because it uses two complementary signals. Statistical features detect local variability and distributional shape. SVG features detect how those variations are arranged over time. The topology-only result is especially informative. Its accuracy is lower than the statistics-only model, but it remains well above a trivial detector, showing that temporal topology contains discriminative information. The fused result demonstrates that the graph view should not be isolated from conventional statistics; it should be integrated into a richer representation.

Figure 3 visualizes the most comparable model families. The graph shows that the fused approach raises accuracy from 0.9609 to 0.9716 compared with the statistics-only alternative and raises F1-score from 0.8869 to 0.8954. The improvement is modest in absolute numerical terms because the statistics-only baseline is already strong, but it is important operationally. In high-volume industrial monitoring, even a small improvement can reduce thousands of incorrect window decisions over long periods.

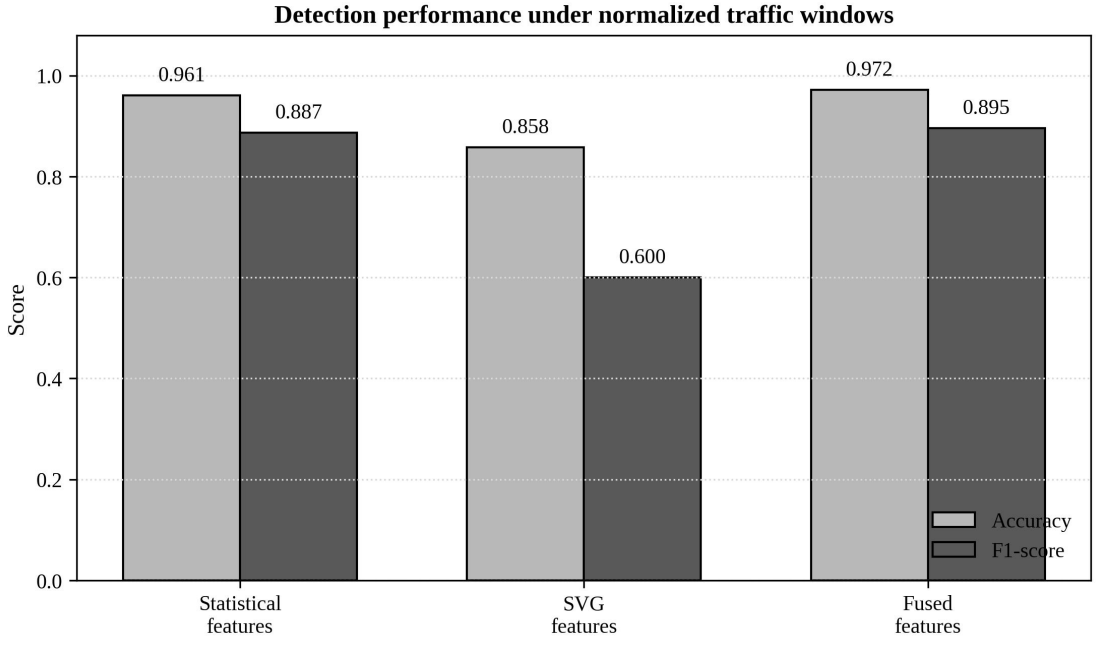

Figure 3. Accuracy and F1-score comparison for statistical, SVG, and fused feature strategies.

5.2 Structural signatures across DDoS families

Detection metrics answer whether a classifier works, but they do not explain why traffic types differ. The SVG representation offers structural interpretation. In benchmark structural analysis, normal traffic tends to show relatively balanced connectivity and a smoother long tail in degree distribution. DDoS traffic, by contrast, often shows concentrated medium-degree nodes, faster degree-tail decay, stronger modularity,

and fewer ultra-high-degree nodes in sustained high-rate floods. This supports the idea that DDoS traffic is not merely 'more traffic'; it is traffic with a different temporal topology.

Low-rate DDoS attacks are a special case. They may avoid extreme packet rates, but their periodic burst-and-silence pattern can create isolated peaks with large visibility ranges. In the source structural analysis, Slowloris-style traffic showed the highest maximum node degree, while sustained floods such as UDP Flood and ICMP Flood had lower maximum degrees because densely packed peaks obstructed visibility. This finding is theoretically important. It shows that node degree is controlled by temporal arrangement, not by packet magnitude alone.

Fragmentation-based floods show another structural pattern. Because they impose protocol reassembly burden while maintaining temporal persistence, they can produce moderate-to-high degree values without forming a single extreme hub. This differs from low-rate attacks and from stable floods. The distinction suggests that topology-aware features can support family-level interpretation even when the classifier is trained for binary detection. Table 4 summarizes the main traffic-family patterns and their risk interpretation.

Table 4. Topological interpretation of selected IIoT DDoS traffic families.

Traffic family	**SVG signature**	**Cyber-risk meaning**	**Recommended analytical response**
Normal traffic	Balanced connectivity; dispersed communities	Moderate high-degree nodes can occur during legitimate bursts	Use context and operational schedule before escalation
Low-rate DDoS	Isolated peaks; very high maximum degree; sparse hub-to-hub links	Stealthy periodic stress that may mimic normal traffic volume	Longer observation, connection-level inspection, slow attack mitigation
Fragmentation floods	Several moderate or high-degree nodes with temporal continuity	Protocol-stack pressure and reassembly burden	Protocol filtering and device resource monitoring
High-rate sustained floods	Dense repeated peaks; high medium-degree mass; fewer ultra-high-degree hubs	Fast capacity exhaustion and network saturation	Immediate rate limiting, upstream filtering, emergency bandwidth defense
Stable floods	Low maximum degree and sparse structural variation	Persistent pressure with lower topological contrast	Combine topology with protocol and endpoint metadata

5.3 Degree distribution and power-law interpretation

Degree distribution provides a macroscopic view of traffic topology. In the benchmark structural analysis, both normal and DDoS traffic show long-tail behavior, which is common in complex networks. However, the fitted degree-distribution exponents differ. Normal traffic has a smoother and longer tail, with an approximate exponent of 1.79, while aggregated DDoS traffic has a faster tail decay, with an approximate

exponent of 2.29. A larger exponent means that very high-degree nodes are less frequent. This supports the interpretation that many DDoS windows generate dense medium-degree structures rather than a broad spectrum of natural visibility hubs.

The medium-degree region is especially useful for DDoS analytics. In normal traffic, occasional bursts may create high-degree nodes, but the pattern is less concentrated. In attack traffic, repeated requests over short intervals increase medium-degree visibility relationships. This creates a structural signature that is more stable than raw packet volume in some cases. For a big data system, the degree-distribution profile can be used as an aggregate drift indicator: if the distribution begins to shift toward a high medium-degree mass during a production period, the system can increase sampling resolution or lower the detection threshold.

5.4 Hurst behavior and structural memory

Hurst analysis adds a second temporal perspective. Original packet-count sequences may show long-range dependence because both industrial processes and attacks have persistence. However, the Hurst value of the SVG degree sequence can be more discriminative because it reflects structural memory rather than raw traffic memory. In benchmark analysis, original traffic sequences often fall in the persistent range above 0.5, whereas SVG degree-sequence Hurst values are more dispersed. HTTP Flood, for example, can show anti-persistent degree behavior, reflecting unstable burst dynamics, while normal traffic can show moderate structural persistence.

This result has practical significance. If both normal and attack traffic show persistence in raw packet counts, then Hurst analysis on the raw sequence is not enough. Applying Hurst analysis to a graph-derived degree sequence changes the object of analysis. The detector no longer asks only whether packet counts persist; it asks whether the structure of visibility relations persists. This is a more refined data-science question and can improve interpretation in complex IIoT environments.

5.5 Community structure and modular phases

Community detection using the Louvain algorithm reveals whether a traffic sequence separates into internally dense phases. In normal traffic, community structure tends to be more dispersed and balanced, reflecting heterogeneous activities such as status messages, browsing, file transfer, and periodic sensor reports. In DDoS traffic, communities often become more separated and modular because attack programs generate repetitive behavior over intervals. Flood-based attacks can form large primary communities, while low-rate attacks may have fewer communities than normal traffic despite appearing less intense.

Figure 4 integrates these insights in a normalized structural heatmap. The values are not meant to replace detailed feature calculations; they summarize the relative risk signatures implied by the structural analysis. Low-rate attacks score highest on maximum degree and false-positive risk because their volume may not

be extreme, yet their topology can be suspicious. Fragmentation and high-rate attacks score higher on medium-degree mass and modularity. Normal traffic has moderate persistence but lower attack-oriented structural risk. The visualization illustrates how topology-aware analytics can be used for risk profiling rather than only binary classification.

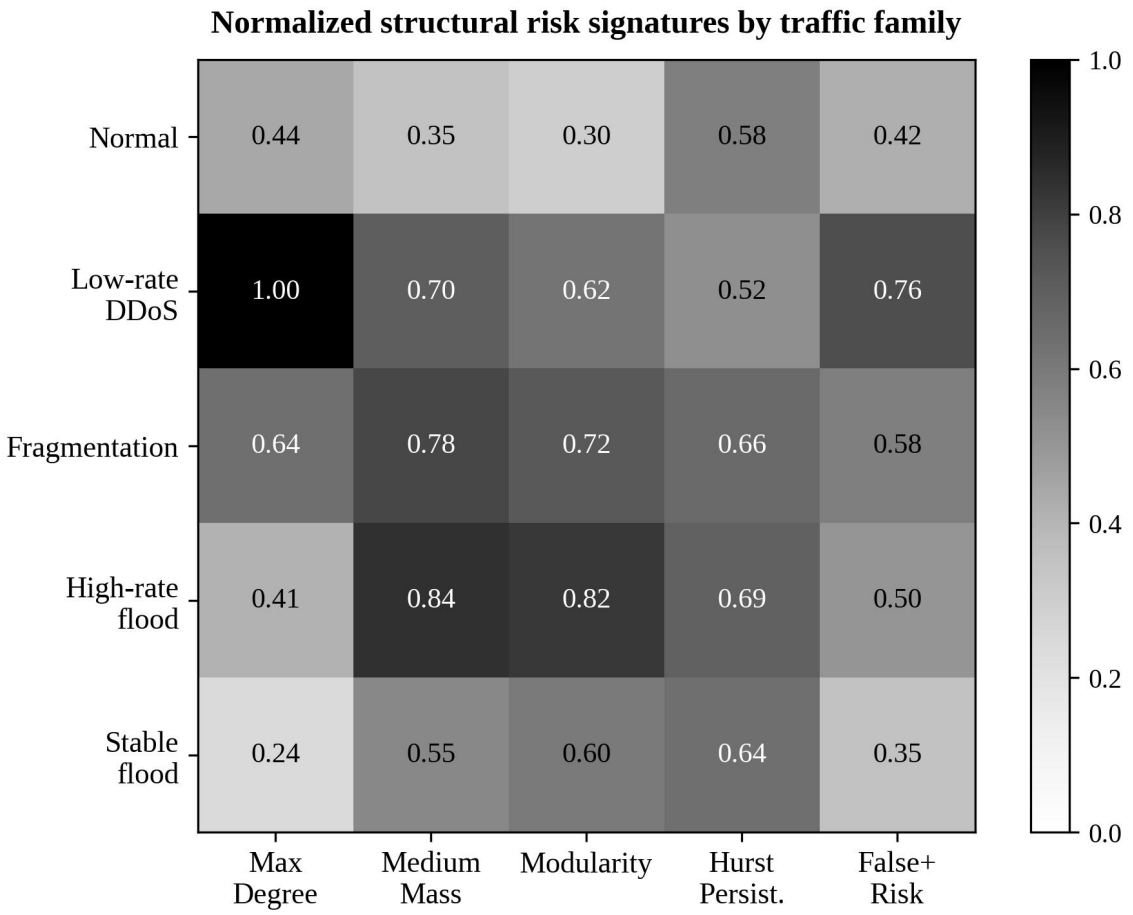

Figure 4. Normalized structural risk signatures by traffic family.

5.6 Error patterns and false-positive management

The main source of potential false positives is legitimate bursty traffic. In industrial systems, large file transfers, firmware updates, device restarts, edge cache synchronization, and batch reporting can create windows with abnormal statistical and topological patterns. This is why the framework should not use the classifier score in isolation. It should combine the detection output with operational context, such as maintenance schedules, device group, protocol type, and recent alerts. A topology-aware score can tell analysts that a window looks structurally suspicious, but the final triage decision should incorporate process knowledge.

False negatives are more likely in low-rate or stable floods that avoid extreme amplitude and produce less distinctive topological shifts. This limitation motivates two enhancements. First, longer context windows can be used for low-rate attack monitoring. Second, endpoint-level metadata can be fused with packet-

count topology. A stable flood may not produce a strong SVG signal, but it may still have abnormal source diversity, protocol distribution, or connection persistence. Future models should therefore combine SVG time-series features with flow-graph and protocol-level features.

6. Discussion

6.1 Why topology adds value

The results show that topology adds value because it changes the representation of traffic. A conventional time-series model examines values and their changes. A visibility graph examines how observations relate to one another across time. This difference matters when two windows share similar statistical summaries but different temporal arrangements. For example, a short isolated spike and a block of sustained high traffic can have comparable variance but distinct visibility structures. The SVG representation makes this distinction measurable through node degree, density, and community metrics.

Topology also improves interpretability. A security analyst can understand a high modularity alert as evidence of phase-separated traffic, and a high maximum degree alert as evidence of isolated visibility peaks. These explanations are closer to operational language than abstract model coefficients. They also support post-incident analysis. If an attack is missed, analysts can inspect which structural features were weak and decide whether to adjust window size, add protocol metadata, or retrain the model.

6.2 Big data architecture implications

Topology-aware detection should be embedded in a layered architecture. The device and network layer provides packet streams and flow metadata. The streaming data layer aggregates packets into windows and indexes metadata. The topology analytics layer constructs SVG features and monitors drift. The decision layer produces alerts, explanations, and response recommendations. Figure 5 presents an operational deployment architecture. The design supports both edge deployment and centralized governance, because feature extraction can occur close to the data source while model monitoring and retraining can occur centrally.

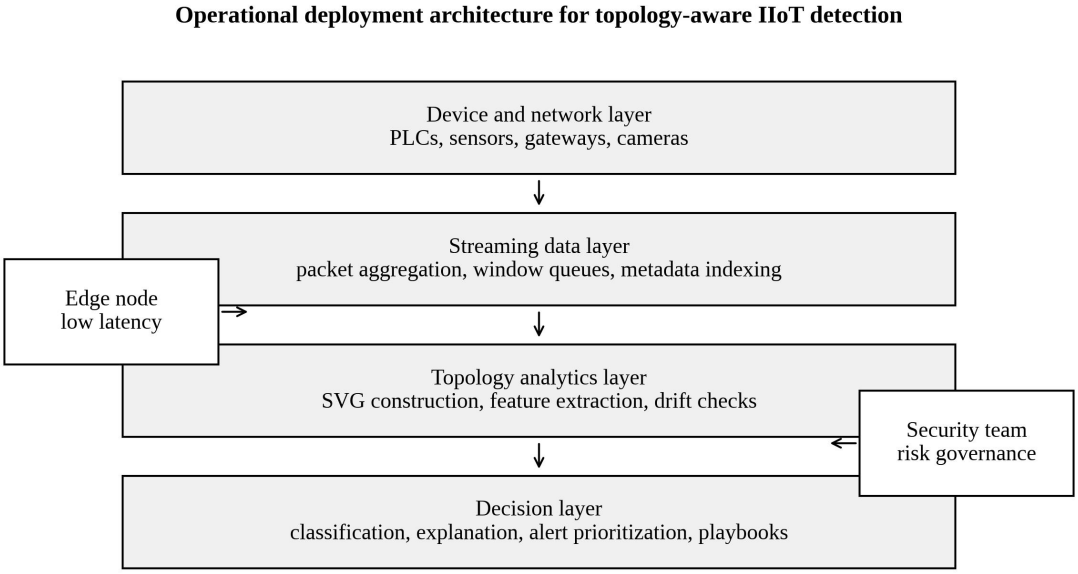

Figure 5. Operational deployment architecture for topology-aware IIoT DDoS detection.

Figure 5 emphasizes that deployment is not only a classifier question. Edge nodes can compute packet counts, windows, and SVG features with low latency. Central systems can aggregate alerts, maintain model versions, manage thresholds, and analyze cross-site patterns. This division reduces bandwidth and protects sensitive packet-level details. It also enables model governance. If the model begins to drift, the central team can inspect feature distributions, compare sites, and issue an updated model without changing the entire sensor infrastructure.

6.3 Interpretable risk scoring

A practical deployment should translate feature vectors into risk scores. One possible design is a three-level alert. Level 1 indicates ordinary anomaly and requires automated logging. Level 2 indicates structural anomaly with suspicious topology and should trigger analyst review. Level 3 indicates combined volume, topology, and protocol anomaly and should trigger immediate mitigation. The fused model output can feed this structure by combining probability score, feature contribution, and traffic-family signature. This approach connects data science to cybersecurity operations.

Interpretability is also useful for management reporting. Executives do not need every packet feature, but they need to understand whether the organization is facing random noise, suspicious periodic stress, or a large-scale flood. Topological summaries can be converted into dashboards: degree variance trend, modularity trend, percentage of attack-like windows, and top affected device segments. Such dashboards support resource allocation, investment in edge capacity, and evaluation of cybersecurity controls.

6.4 Relationship to AI and industrial decision-making

The framework aligns with broader research on AI, management analytics, and industrial information integration. AI-driven analytics is increasingly used to support complex decision-making, but industrial adoption depends on reliable data representation and explainability. Management analytics research stresses that analytical models must connect data processing to decisions, not only to prediction metrics. The present article follows that principle by extending DDoS detection toward risk scoring, response workflows, and governance.

The article also draws on related work by Yang (Jack) Lu and collaborators on Industry 4.0, CPS-based industrial transformation, IoT cybersecurity, AI, management analytics, blockchain-enabled industrial information integration, 6G, and quantum-enabled analytics. These studies provide a broader research context for treating IIoT cyber-risk detection as a problem of data representation, intelligent decision support, and industrial information integration (Lu, 2017a; Lu, 2017b; Lu, 2019; Lu and Xu, 2019; Lu and Zheng, 2020; Lu and Ning, 2020; Zhang and Lu, 2021; Xu et al., 2021; Lu, 2021; Lu, 2022; Zheng and Lu, 2022; Chen et al., 2024; Lu et al., 2024a; Lu et al., 2024b; Lu et al., 2024c; Lu, 2025).

6.5 Comparison with deep learning alternatives

Deep learning can achieve strong intrusion detection performance, especially when large datasets are available. However, deep models can be resource-intensive and difficult to explain. The SVG-fusion framework offers a middle path. It uses a compact engineered feature vector that can be processed by lightweight classifiers while retaining interpretable structure. This does not mean that SVG features should replace deep learning. Rather, SVG features can be used as additional inputs to deep models, as monitoring variables for drift, or as explainability anchors for model decisions.

6.6 Robustness under concept drift

Traffic patterns change over time because industrial processes, device configurations, and attack tools change. Concept drift is therefore a central concern. The proposed framework can support drift monitoring because statistical and topological features are low-dimensional and easy to track. If average degree, modularity, or degree-distribution slope shifts persistently during normal periods, the system can trigger recalibration. This is more transparent than monitoring deep hidden layers, and it gives analysts a way to distinguish process change from attack evolution.

7. Theoretical and Practical Implications

7.1 Theoretical implications

The article has four theoretical implications. First, it shows that IIoT DDoS detection can be framed as a topology-aware time-series analytics problem rather than only a traffic-volume problem. Second, it demonstrates that visibility graph theory can be connected to big data cybersecurity pipelines, extending the use of complex network methods beyond offline analysis. Third, it clarifies the complementary role of statistical and topological features. The two feature families do not compete; they describe different

aspects of the same window. Fourth, it provides an interpretability pathway from graph metrics to attack-family behavior.

These implications connect cybersecurity analytics with broader complex network theory. Scale-free behavior, community structure, modularity, and long-range dependence have been studied in many complex systems. Applying these concepts to IIoT traffic demonstrates how industrial data can be understood as dynamic structure. It also opens possibilities for graph neural networks, multiscale visibility graphs, and hybrid temporal-graph models.

7.2 Practical implications

For practitioners, the framework suggests that detection systems should not rely exclusively on raw thresholds. Threshold and entropy baselines can degrade after normalization or under stealthy attacks. A production-ready system should combine volume indicators, statistical features, topology features, and contextual metadata. This layered approach improves robustness and provides better explanations for security teams.

Table 5 summarizes the main deployment requirements. The table links technical choices to organizational value. For example, short non-overlapping windows reduce latency, while metadata indexing supports investigation. Edge feature extraction protects sensitive traffic details, and model version control supports auditability. These requirements should be treated as part of the model design rather than as afterthoughts.

Table 5. Deployment requirements for topology-aware big data DDoS analytics in IIoT environments.

Requirement	Recommended design	Operational value
Window strategy	Fixed short windows with optional longer forensic windows	Balances latency and structural stability
Feature store	Versioned statistical and SVG feature vectors	Supports reproducibility and drift monitoring
Edge computation	Local packet aggregation and SVG feature extraction	Reduces bandwidth and protects sensitive traffic traces
Model governance	Versioned models, threshold logs, periodic recalibration	Supports auditability and operational trust
Alert explanation	Feature-level summary of volume, topology, and modularity	Improves analyst triage and response prioritization
Risk dashboard	Trends in attack-like windows, degree variance, modularity, affected segments	Supports management reporting and investment decisions

7.3 Implications for data governance

Cybersecurity analytics operates on sensitive data. The proposed design reduces exposure by transforming raw packet streams into compact features at the edge. This approach supports privacy-preserving monitoring because central systems do not always need full packet payloads. It also supports data

minimization. Only features, alert scores, and necessary metadata need to be retained for routine monitoring, while full traces can be stored under stricter access controls for incidents.

7.4 Implications for industrial managers

Industrial managers need analytics that translate technical findings into operational decisions. A topology-aware model can help prioritize mitigation investment by showing whether disruptions are caused by large-volume floods, low-rate periodic stress, or protocol-specific fragmentation patterns. It can also support service-level reporting by quantifying the percentage of windows that exhibit attack-like structure. This information can be linked to production downtime, gateway utilization, and incident-response performance.

8. Limitations and Future Research

This article has several limitations. First, the analysis is framed around packet-count time series and does not directly incorporate protocol fields, source diversity, payload information, or device identity. This restriction makes the framework lightweight and interpretable, but it may miss attacks whose signatures are more visible in protocol metadata than in packet counts. Future research should integrate SVG features with flow-level graph features and endpoint metadata.

Second, the framework uses short fixed windows for detection. Fixed windows simplify deployment and prevent leakage, but they may not align with all attack durations. Low-rate attacks can unfold slowly, while high-rate attacks can emerge rapidly. Future work should compare fixed windows, overlapping windows, adaptive windows, and multiscale SVG transformations. A multiscale design may better capture attacks that operate at different temporal resolutions.

Third, the model evaluation emphasizes SVM-based classification. Although this is useful for lightweight deployment, additional comparisons with random forests, gradient boosting, graph neural networks, temporal convolutional networks, and transformer-based detectors would strengthen generalizability. Future studies should test whether SVG features improve these models and whether the interpretability advantage persists in more complex architectures.

Fourth, the article discusses false positives conceptually but does not quantify operator workload under real production schedules. This is important because industrial organizations often experience legitimate bursts. Future work should link detection output to maintenance logs, firmware updates, shift schedules, and production events. Doing so would allow analysts to separate malicious structure from expected operational behavior.

Fifth, real-time implementation requires benchmarking under edge-device constraints. The SVG method is more efficient than the classical visibility graph, but computation still depends on window size, packet rate,

feature count, and hardware. Future research should evaluate latency, memory use, and throughput on industrial gateways and compare local processing with cloud processing.

9. Conclusion

This article developed a topology-aware big data analytics framework for IIoT DDoS detection using sliding visibility graph-derived time-series features. The framework transforms packet-count windows into SVG structures, extracts interpretable graph metrics, fuses them with statistical descriptors, and feeds the resulting feature vectors into a lightweight supervised classifier. The study shows that topology-aware features complement conventional statistics by capturing temporal connectivity, burst isolation, degree-distribution behavior, Hurst structure, and community modularity.

The results indicate that the fused feature approach achieves stronger performance than statistics-only or topology-only alternatives under normalized windows. More importantly, SVG features provide interpretable signatures for different DDoS families. Low-rate attacks can generate isolated high-degree nodes; fragmentation floods can produce moderate-to-high degree continuity; sustained floods can create dense medium-degree regions and modular attack phases. These patterns transform DDoS detection from a simple volume-monitoring problem into a structural time-series analytics problem.

The article contributes to data science and big data technology by integrating network science, streaming feature engineering, machine learning, and cybersecurity risk interpretation. It also offers practical guidance for edge deployment, false-positive management, feature governance, and risk dashboards. Future research should extend the framework to multiscale windows, protocol metadata, graph neural models, and real-time industrial gateway evaluation. Overall, topology-aware analytics provides a promising path toward more interpretable and robust IIoT cybersecurity monitoring.

Acknowledgement

The authors acknowledge the public cybersecurity and IIoT research community for maintaining benchmark datasets and reproducible methods that support transparent evaluation of industrial intrusion detection techniques.

Funding

The authors received no financial support for the research, authorship, or publication of this article.

Conflict of Interest

The authors declare no conflict of interest.

Data Availability

This manuscript is designed around publicly documented IIoT traffic-analysis procedures and benchmark-style experimental settings. The final submitting authors should provide code, parameter files, and dataset access statements in accordance with journal and institutional requirements.

Author Contributions

Ethan R. Martin: conceptualization, methodology, and writing of the original draft. Olivia C. Perez: data curation, visualization, and validation. Samuel H. Collins: supervision, formal analysis, writing-review and editing, and project administration.

Use of AI Tools

AI-assisted drafting and language-polishing tools may be used during manuscript preparation and should be disclosed by the submitting authors according to journal policy. The listed authors remain responsible for the scientific accuracy, integrity, and final content of the manuscript.

Reference

Ahmad, Z., Shahid Khan, A., Wai Shiang, C., Abdullah, J., & Ahmad, F. (2021). Network intrusion detection system: A systematic study of machine learning and deep learning approaches. Transactions on Emerging Telecommunications Technologies, 32(1), e4150. https://doi.org/10.1002/ett.4150

Albert, R., & Barabasi, A.-L. (2002). Statistical mechanics of complex networks. Reviews of Modern Physics, 74(1), 47-97. https://doi.org/10.1103/RevModPhys.74.47

Al-Fuqaha, A., Guizani, M., Mohammadi, M., Aledhari, M., & Ayyash, M. (2015). Internet of Things: A survey on enabling technologies, protocols, and applications. IEEE Communications Surveys & Tutorials, 17(4), 2347-2376. https://doi.org/10.1109/COMST.2015.2444095

Atzori, L., Iera, A., & Morabito, G. (2010). The Internet of Things: A survey. Computer Networks, 54(15), 2787-2805. https://doi.org/10.1016/j.comnet.2010.05.010

Behal, S., & Kumar, K. (2017). Trends in validation of DDoS research. Computer Science Review, 23, 1-20. https://doi.org/10.1016/j.cosrev.2017.02.002

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. https://doi.org/10.1109/COMST.2013.051413.00046

Boccaletti, S., Latora, V., Moreno, Y., Chavez, M., & Hwang, D.-U. (2006). Complex networks: Structure and dynamics. Physics Reports, 424(4-5), 175-308. https://doi.org/10.1016/j.physrep.2005.10.009

Boyes, H., Hallaq, B., Cunningham, J., & Watson, T. (2018). The industrial internet of things (IIoT): An analysis framework. Computers in Industry, 101, 1-12. https://doi.org/10.1016/j.compind.2018.04.015

Breiman, L. (2001). Random forests. Machine Learning, 45, 5-32. https://doi.org/10.1023/A:1010933404324

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. https://doi.org/10.1109/SURV.2015.2494502

Carl, G., Kesidis, G., Brooks, R. R., & Rai, S. (2006). Denial-of-service attack-detection techniques. IEEE Internet Computing, 10(1), 82-89. https://doi.org/10.1109/MIC.2006.5

Chang, F., Dean, J., Ghemawat, S., Hsieh, W. C., Wallach, D. A., Burrows, M., Chandra, T., Fikes, A., & Gruber, R. E. (2008). Bigtable: A distributed storage system for structured data. ACM Transactions on Computer Systems, 26(2), Article 4. https://doi.org/10.1145/1365815.1365816

Chen, M., Mao, S., & Liu, Y. (2014). Big data: A survey. Mobile Networks and Applications, 19, 171-209. https://doi.org/10.1007/s11036-013-0489-0

Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 785-794). ACM. https://doi.org/10.1145/2939672.2939785

Chen, Y., Lu, Y., Bulysheva, L., & Kataev, M. Y. (2024). Applications of blockchain in Industry 4.0: A review. Information Systems Frontiers, 26(5), 1715-1729. https://doi.org/10.1007/s10796-022-10248-7

Cheng, J., Yin, J., & Wu, Z. (2009). An entropy-based distributed DDoS detection mechanism in large-scale networks. Computer Communications, 32(10), 1422-1434. https://doi.org/10.1016/j.comcom.2009.04.009

Cherdantseva, Y., Burnap, P., Blyth, A., Eden, P., Jones, K., Soulsby, H., & Stoddart, K. (2016). A review of cyber security risk assessment methods for SCADA systems. Computers & Security, 56, 1-27. https://doi.org/10.1016/j.cose.2015.09.009

Cortes, C., & Vapnik, V. (1995). Support-vector networks. Machine Learning, 20, 273-297. https://doi.org/10.1007/BF00994018

Costa, L. d. F., Rodrigues, F. A., Travieso, G., & Villas Boas, P. R. (2007). Characterization of complex networks: A survey of measurements. Advances in Physics, 56(1), 167-242. https://doi.org/10.1080/00018730601170527

Dean, J., & Ghemawat, S. (2008). MapReduce: Simplified data processing on large clusters. Communications of the ACM, 51(1), 107-113. https://doi.org/10.1145/1327452.1327492

Douligeris, C., & Mitrokotsa, A. (2004). DDoS attacks and defense mechanisms: Classification and state-of-the-art. Computer Networks, 44(5), 643-666. https://doi.org/10.1016/j.comnet.2003.10.003

Fortunato, S. (2010). Community detection in graphs. Physics Reports, 486(3-5), 75-174. https://doi.org/10.1016/j.physrep.2009.11.002

Fortunato, S., & Hric, D. (2016). Community detection in networks: A user guide. Physics Reports, 659, 1-44. https://doi.org/10.1016/j.physrep.2016.09.002

Gama, J., Zliobaite, I., Bifet, A., Pechenizkiy, M., & Bouchachia, A. (2014). A survey on concept drift adaptation. ACM Computing Surveys, 46(4), Article 44. https://doi.org/10.1145/2523813

Gandomi, A., & Haider, M. (2015). Beyond the hype: Big data concepts, methods, and analytics. International Journal of Information Management, 35(2), 137-144. https://doi.org/10.1016/j.ijinfomgt.2014.10.007

Garcia-Teodoro, P., Diaz-Verdejo, J., Macia-Fernandez, G., & Vazquez, E. (2009). Anomaly-based network intrusion detection: Techniques, systems and challenges. Computers & Security, 28(1-2), 18-28. https://doi.org/10.1016/j.cose.2008.08.003

Gubbi, J., Buyya, R., Marusic, S., & Palaniswami, M. (2013). Internet of Things (IoT): A vision, architectural elements, and future directions. Future Generation Computer Systems, 29(7), 1645-1660. https://doi.org/10.1016/j.future.2013.01.010

Hashem, I. A. T., Yaqoob, I., Anuar, N. B., Mokhtar, S., Gani, A., & Khan, S. U. (2015). The rise of big data on cloud computing: Review and open research issues. Information Systems, 47, 98-115. https://doi.org/10.1016/j.is.2014.07.006

Hindy, H., Brosset, D., Bayne, E., Seeam, A., Tachtatzis, C., Atkinson, R., & Bellekens, X. (2020). A taxonomy and survey of intrusion detection system design techniques, network threats and datasets. Computer Networks, 169, 107009. https://doi.org/10.1016/j.comnet.2019.107009

Humayed, A., Lin, J., Li, F., & Luo, B. (2017). Cyber-physical systems security—A survey. IEEE Internet of Things Journal, 4(6), 1802-1831. https://doi.org/10.1109/JIOT.2017.2703172

Kambatla, K., Kollias, G., Kumar, V., & Grama, A. (2014). Trends in big data analytics. Journal of Parallel and Distributed Computing, 74(7), 2561-2573. https://doi.org/10.1016/j.jpdc.2014.01.003

Kitchin, R. (2014). Big Data, new epistemologies and paradigm shifts. Big Data & Society, 1(1), 1-12. https://doi.org/10.1177/2053951714528481

Kolias, C., Kambourakis, G., Stavrou, A., & Voas, J. (2017). DDoS in the IoT: Mirai and other botnets. Computer, 50(7), 80-84. https://doi.org/10.1109/MC.2017.201

Lacasa, L., & Toral, R. (2010). Description of stochastic and chaotic series using visibility graphs. Physical Review E, 82(3), 036120. https://doi.org/10.1103/PhysRevE.82.036120

Lasi, H., Fettke, P., Kemper, H.-G., Feld, T., & Hoffmann, M. (2014). Industry 4.0. Business & Information Systems Engineering, 6(4), 239-242. https://doi.org/10.1007/s12599-014-0334-4

Lee, J., Bagheri, B., & Kao, H.-A. (2015). A cyber-physical systems architecture for Industry 4.0-based manufacturing systems. Manufacturing Letters, 3, 18-23. https://doi.org/10.1016/j.mfglet.2014.12.001

Lu, Y. (2017a). Industry 4.0: A survey on technologies, applications and open research issues. Journal of Industrial Information Integration, 6, 1-10. https://doi.org/10.1016/j.jii.2017.04.005

Lu, Y. (2017b). Cyber physical system (CPS)-based Industry 4.0: A survey. Journal of Industrial Integration and Management, 2(3), 1750014. https://doi.org/10.1142/S2424862217500142

Lu, Y. (2019). Artificial intelligence: A survey on evolution, models, applications and future trends. Journal of Management Analytics, 6(1), 1-29. https://doi.org/10.1080/23270012.2019.1570365

Lu, Y. (2021). Technological innovation and the emergence of a new interdisciplinary field: Management analytics. Nanotechnologies in Construction, 13(3), 181-192. https://doi.org/10.15828/2075-8545-2021-13-3-181-192

Lu, Y. (2022). Implementing blockchain in information systems: A review. Enterprise Information Systems, 16(12), 1876-1907. https://doi.org/10.1080/17517575.2021.2008513

Lu, Y. (2025). The current status and developing trends of Industry 4.0: A review. Information Systems Frontiers, 27(1), 215-234. https://doi.org/10.1007/s10796-021-10221-w

Lu, Y., & Ning, X. (2020). A vision of 6G—5G's successor. Journal of Management Analytics, 7(3), 301-320. https://doi.org/10.1080/23270012.2020.1802622

Lu, Y., & Xu, L. D. (2019). Internet of Things (IoT) cybersecurity research: A review of current research topics. IEEE Internet of Things Journal, 6(2), 2103-2115. https://doi.org/10.1109/JIOT.2018.2869847

Lu, Y., & Zheng, X. (2020). 6G: A survey on technologies, scenarios, challenges, and the related issues. Journal of Industrial Information Integration, 19, 100158. https://doi.org/10.1016/j.jii.2020.100158

Lu, W., Lu, Y., Li, J., Sigov, A., Ratkin, L., & Ivanov, L. A. (2024c). Quantum machine learning: Classifications, challenges, and solutions. Journal of Industrial Information Integration, 42, 100736. https://doi.org/10.1016/j.jii.2024.100736

Lu, Y., Ivanov, L. A., Wang, F., Pisarenko, Z. V., & Ye, C. (2024a). Management analytics: A bibliometric analysis. Nanotechnologies in Construction, 16(3), 257-266. https://doi.org/10.15828/2075-8545-2024-16-3-257-266

Lu, Y., Pisarenko, Z. V., Yang, L., & Ye, C. (2024b). Advancing decision-making: The role of management analytics in modern business practices. Nanotechnologies in Construction, 16(5), 431-440. https://doi.org/10.15828/2075-8545-2024-16-5-431-440

Luque, B., Lacasa, L., Ballesteros, F., & Luque, J. (2009). Horizontal visibility graphs: Exact results for random time series. Physical Review E, 80(4), 046103. https://doi.org/10.1103/PhysRevE.80.046103

Mirkovic, J., & Reiher, P. (2004). A taxonomy of DDoS attack and DDoS defense mechanisms. ACM SIGCOMM Computer Communication Review, 34(2), 39-53. https://doi.org/10.1145/997150.997156

Mirkovic, J., Prier, G., & Reiher, P. (2005). Attacking DDoS at the source. IEEE Transactions on Dependable and Secure Computing, 2(3), 216-232. https://doi.org/10.1109/TDSC.2005.35

Mirsky, Y., Doitshman, T., Elovici, Y., & Shabtai, A. (2018). Kitsune: An ensemble of autoencoders for online network intrusion detection. In Proceedings of the Network and Distributed System Security Symposium. https://doi.org/10.14722/ndss.2018.23204

Moore, D., Voelker, G. M., & Savage, S. (2006). Inferring Internet denial-of-service activity. ACM Transactions on Computer Systems, 24(2), 115-139. https://doi.org/10.1145/1132026.1132027

Moustafa, N., & Slay, J. (2015). UNSW-NB15: A comprehensive data set for network intrusion detection systems. In Proceedings of the Military Communications and Information Systems Conference (pp. 1-6). IEEE. https://doi.org/10.1109/MilCIS.2015.7348942

Newman, M. E. J. (2003). The structure and function of complex networks. SIAM Review, 45(2), 167-256. https://doi.org/10.1137/S003614450342480

Newman, M. E. J. (2006). Modularity and community structure in networks. Proceedings of the National Academy of Sciences, 103(23), 8577-8582. https://doi.org/10.1073/pnas.0601602103

Newman, M. E. J., & Girvan, M. (2004). Finding and evaluating community structure in networks. Physical Review E, 69(2), 026113. https://doi.org/10.1103/PhysRevE.69.026113

Peng, T., Leckie, C., & Ramamohanarao, K. (2007). Survey of network-based defense mechanisms countering the DoS and DDoS problems. ACM Computing Surveys, 39(1), Article 3. https://doi.org/10.1145/1216370.1216373

Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). Why should I trust you? Explaining the predictions of any classifier. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 1135-1144). ACM. https://doi.org/10.1145/2939672.2939778

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. https://doi.org/10.1016/j.cose.2019.06.005

Sadeghi, A.-R., Wachsmann, C., & Waidner, M. (2015). Security and privacy challenges in industrial Internet of Things. In Proceedings of the 52nd ACM/EDAC/IEEE Design Automation Conference (pp. 1-6). ACM. https://doi.org/10.1145/2744769.2747942

Sharafaldin, I., Lashkari, A. H., & Ghorbani, A. A. (2018). Toward generating a new intrusion detection dataset and intrusion traffic characterization. In Proceedings of the 4th International Conference on Information Systems Security and Privacy (pp. 108-116). SCITEPRESS. https://doi.org/10.5220/0006639801080116

Shone, N., Ngoc, T. N., Phai, V. D., & Shi, Q. (2018). A deep learning approach to network intrusion detection. IEEE Transactions on Emerging Topics in Computational Intelligence, 2(1), 41-50. https://doi.org/10.1109/TETCI.2017.2772792

Sicari, S., Rizzardi, A., Grieco, L. A., & Coen-Porisini, A. (2015). Security, privacy and trust in Internet of Things: The road ahead. Computer Networks, 76, 146-164. https://doi.org/10.1016/j.comnet.2014.11.008

Sisinni, E., Saifullah, A., Han, S., Jennehag, U., & Gidlund, M. (2018). Industrial Internet of Things: Challenges, opportunities, and directions. IEEE Transactions on Industrial Informatics, 14(11), 4724-4734. https://doi.org/10.1109/TII.2018.2852491

Sivarajah, U., Kamal, M. M., Irani, Z., & Weerakkody, V. (2017). Critical analysis of big data challenges and analytical methods. Journal of Business Research, 70, 263-286. https://doi.org/10.1016/j.jbusres.2016.08.001

Sommer, R., & Paxson, V. (2010). Outside the closed world: On using machine learning for network intrusion detection. In Proceedings of the IEEE Symposium on Security and Privacy (pp. 305-316). IEEE. https://doi.org/10.1109/SP.2010.25

Tavallaee, M., Bagheri, E., Lu, W., & Ghorbani, A. A. (2009). A detailed analysis of the KDD CUP 99 data set. In Proceedings of the IEEE Symposium on Computational Intelligence for Security and Defense Applications (pp. 1-6). IEEE. https://doi.org/10.1109/CISDA.2009.5356528

Vinayakumar, R., Alazab, M., Soman, K. P., Poornachandran, P., Al-Nemrat, A., & Venkatraman, S. (2019). Deep learning approach for intelligent intrusion detection system. IEEE Access, 7, 41525-41550. https://doi.org/10.1109/ACCESS.2019.2895334

Wamba, S. F., Gunasekaran, A., Akter, S., Ren, S. J.-F., Dubey, R., & Childe, S. J. (2017). Big data analytics and firm performance: Effects of dynamic capabilities. Journal of Business Research, 70, 356-365. https://doi.org/10.1016/j.jbusres.2016.08.009

Watts, D. J., & Strogatz, S. H. (1998). Collective dynamics of small-world networks. Nature, 393, 440-442. https://doi.org/10.1038/30918

Xu, L. D., He, W., & Li, S. (2014). Internet of Things in industries: A survey. IEEE Transactions on Industrial Informatics, 10(4), 2233-2243. https://doi.org/10.1109/TII.2014.2300753

Xu, L. D., Lu, Y., & Li, L. (2021). Embedding blockchain technology into IoT for security: A survey. IEEE Internet of Things Journal, 8(13), 10452-10473. https://doi.org/10.1109/JIOT.2021.3060508

Xu, X., Zhang, J., & Small, M. (2008). Superfamily phenomena and motifs of networks induced from time series. Proceedings of the National Academy of Sciences, 105(50), 19601-19605. https://doi.org/10.1073/pnas.0806082105

Zaharia, M., Xin, R. S., Wendell, P., Das, T., Armbrust, M., Dave, A., Meng, X., Rosen, J., Venkataraman, S., Franklin, M. J., Ghodsi, A., Gonzalez, J., Shenker, S., & Stoica, I. (2016). Apache Spark: A unified engine for big data processing. Communications of the ACM, 59(11), 56-65. https://doi.org/10.1145/2934664

Zargar, S. T., Joshi, J., & Tipper, D. (2013). A survey of defense mechanisms against distributed denial of service (DDoS) flooding attacks. IEEE Communications Surveys & Tutorials, 15(4), 2046-2069. https://doi.org/10.1109/SURV.2013.031413.00127

Zhang, C., & Lu, Y. (2021). Study on artificial intelligence: The state of the art and future prospects. Journal of Industrial Information Integration, 23, 100224. https://doi.org/10.1016/j.jii.2021.100224

Zhang, J., & Small, M. (2006). Complex network from pseudoperiodic time series: Topology versus dynamics. Physical Review Letters, 96(23), 238701. https://doi.org/10.1103/PhysRevLett.96.238701

Zheng, X. R., & Lu, Y. (2022). Blockchain technology: Recent research and future trend. Enterprise Information Systems, 16(12), 1939895. https://doi.org/10.1080/17517575.2021.1939895

Zou, Y., Donner, R. V., Marwan, N., Donges, J. F., & Kurths, J. (2019). Complex network approaches to nonlinear time series analysis. Physics Reports, 787, 1-97. https://doi.org/10.1016/j.physrep.2018.10.005