CYBERSECURITY AND TRUSTWORTHY ARTIFICIAL INTELLIGENCE:

DETECTION, GOVERNANCE, AND RELIABILITY

EDITOR

PROF. DR. EYYÜP GÜLBANDILAR



BİDGE

BİDGE Yayınları

Cybersecurity and Trustworthy Artificial Intelligence: Detection, Governance, And Reliability

Editor: PROF. DR. EYYÜP GÜLBANDILAR

ISBN: 978-625-5057-84-6

1st Edition
Page Layout By: Gözde YÜCEL
Publication Date: 2026-10-25
BİDGE Yayınları

All rights reserved. No part of this work may be reproduced in any form or by any means, except for brief quotations for promotional purposes with proper source attribution, without the written permission of the publisher and the editor.

Certificate No: 71374
All rights reserved © BİDGE Yayınları
www.bidgeyayinlari.com.tr - bidgeyayinlari@gmail.com
Krc Bilişim Ticaret ve Organizasyon Ltd. Şti.
Güzeltepe Mahallesi Abidin Daver Sokak Sefer Apartmanı No: 7/9 Çankaya / Ankara

CONTENTS

Unsupervısed Learnıng ın Sıem Systems: Deployment Requırements and Lımıtatıons 1
ÖZGE TAŞ

GRAPH-BASED CYBER-ATTACK DETECTION: REPRESENTATION STRATEGIES, TOPOLOGICAL ANALYSIS, AND COMPUTATIONAL EFFICIENCY 20
ALİ MELİH KANCA, İLKER TÜRKER

ADVERSARIAL ARTIFICIAL INTELLIGENCE AND CROSS SECTOR CYBERSECURITY GOVERNANCE 64
ÖMER SEVİNÇ

AUTOMATED DETECTION OF PREDATORY JOURNALS: A SURVEY OF COMPUTATIONAL APPROACHES AND RELIABILITY CHALLENGES 73
DİDEM YANGEÇ

CHAPTER 1

UNSUPERVISED LEARNING IN SIEM SYSTEMS: DEPLOYMENT REQUIREMENTS AND LIMITATIONS

ÖZGE TAŞ[1]

Introduction

Security information and event management (SIEM) systems collect and correlate logs from network devices, applications, identity services, and endpoints within a common framework. Rule-based monitoring remains a basic means of identifying known attack patterns and violations of organizational policy. However, users, services, and workloads change, making normal behavior difficult to describe with fixed thresholds. Unsupervised learning can help examine deviations that have not been specified individually in advance (Bhatt, Manadhata & Zomlot, 2014; Vielberth et al., 2020).

Rare behaviour is not necessarily malicious. An overnight backup or a newly deployed service may also be statistically unusual. In security applications, anomaly scores therefore need to be interpreted alongside the context of the event. An algorithm may behave differently across organisations because normal activity,

[1] Lecturer, Kastamonu University, Çatalzeytin Vocational School, Department of Computer Technologies, ORCID: 0000-0001-7220-5054

logging coverage and attacks differ between environments (Chandola, Banerjee & Kumar, 2009; Sommer & Paxson, 2010).

This chapter compares unsupervised methods that may support SIEM workflows, focusing on their operating principles, data requirements, and limitations. The Isolation Forest (IF) and one-class support vector machine (OCSVM) are treated in detail alongside local-density, clustering, and reconstruction methods. PCA and CCA/RFCCA serve different purposes in representation and relationship analysis, and fuzzy inference is considered as decision support. This review does not attempt to catalogue the algorithms used by every commercial SIEM product. It proposes no new model and offers no experimental ranking of detection performance.

A SIEM system and a network intrusion detection system perform different functions. Moving an anomaly detection method from network traffic analysis into a SIEM workflow requires integration across data sources, a clear definition of the event being assessed and an output that analysts can investigate. Performance on a network dataset alone does not establish that a method will perform equally well in an enterprise SIEM environment (Sommer & Paxson, 2010; Vielberth et al., 2020).

Preparing SIEM Data for Learning Algorithms

Observation point and logging coverage

An algorithm can assess only the behavior captured in its input data. A passive network sensor may be unable to inspect the content of TLS-encrypted packets, although connection durations, directions, and packet sizes may remain available. TLS termination points, application servers, and endpoint logs provide other levels of detail. A server-side access log can contain request and response information, even when the connection is encrypted. The observation point thus determines when content inspection is

possible and when analysis must rely on flow statistics (Rescorla, 2018; MontazeriShatoori et al., 2020).

The unit of observation should be chosen before model development. An HTTP request, a completed network flow, a user session and host activity aggregated over a time window represent different aspects of behaviour. The total bytes per flow and hourly connections per user answer different questions. The deviation of interest should guide this choice. Identity logs, for example, may reveal departures from a user's usual login hours, whereas request frequency and status-code distributions may be more useful in web server logs. These examples illustrate data design choices, not measured detection results.

Features and data preparation

Raw logs typically contain timestamps, numerical fields, categories and free text. Most methods reviewed here require a numerical representation rather than the original log text. Candidate features include packet and byte counts, flow duration, packet-length distributions, intervals between requests and error rates. Context such as user role, service type or working hours can help explain groups of behaviour. Encoding an IP address or user identifier as an ordinary integer, however, may imply a meaningful numerical distance between values where none exists. The encoding should reflect what the field represents.

Records need to be aligned in time, and duplicate events and collection gaps should be identified. Missing-value treatment and record exclusions must also be clear. Scaling is particularly relevant to distance- and kernel-based methods, where a variable with a large numerical range may dominate the others. Suitable transformations can help with heavy-tailed variables such as traffic volume. Standard IF does not use the same kind of distance calculation, but irrelevant dimensions, encoding choices and the representation of outliers can

still affect its results (Liu, Ting & Zhou, 2012; Zimek, Schubert & Kriegel, 2012).

Preprocessing parameters must be learned from training data alone. Estimating scaling coefficients or selecting variables using the test period introduces information from the future. If labels guide feature selection, the overall process is no longer entirely label-free. Distributing records from the same session across data partitions can likewise make evaluation appear easier than it is. When public resources such as CIRA-CIC-DoHBrw-2020 are used, the dataset version, extracted fields and unit of observation should be specified. The dataset name does not supply these details (MontazeriShatoori et al., 2020; Sommer & Paxson, 2010).

Basic Conditions for Unsupervised Anomaly Detection

Unsupervised outlier detection examines the structure of a dataset that may contain anomalous observations. Novelty detection instead learns from a reference set assumed to represent normal behaviour and then assesses later observations. The absence of attack labels during training does not make reference-data quality unimportant. In OCSVM and autoencoder applications trained mainly on normal data, persistent malicious activity may become part of the learned pattern. The learning setting needs to be stated explicitly (Chandola, Banerjee & Kumar, 2009; Schölkopf et al., 2001; Scikit-learn developers, n.d.-b).

Attacks may also account for only a small share of all events. The base-rate problem discussed by Axelsson (2000) explains why high sensitivity alone does not ensure a useful alert queue. Let A denote an attack and T the generation of an alert. Bayes' theorem gives the probability that an alert corresponds to an attack:

$$P(A \mid T) = \frac{P(T \mid A)\,P(A)}{P(T \mid A)\,P(A) + P(T \mid A^{c})\,P(A^{c})} \quad (1)$$

Here, P(A) is the attack base rate, P(T | A) is sensitivity and P(T | A^c) is the false-positive rate. When attacks are rare, alerts on legitimate events may dominate the investigation queue. Evaluation should therefore consider alert precision and investigation workload alongside sensitivity. Security operations centre (SOC) analysts' accounts of alert quality also show that a correctly triggered rule may have a benign explanation. Benign alerts should not all be treated as the same kind of model error (Axelsson, 2000; Alahmadi, Axon & Martinovic, 2022).

How the Main Methods Work and Where They Are Limited

Isolation Forest

Isolation Forest (IF) measures how easily an observation can be separated through random partitions. Each tree is built from a subsample of ψ observations drawn from the training data. At each node, a feature and a split point are selected at random. Observations that differ markedly from the rest are expected to be isolated after relatively few splits. The path length of observation x in tree t is denoted by $h_t(x)$. If several observations remain in a leaf, the expected additional path length needed to separate them is included (Liu, Ting & Zhou, 2008, 2012).

$$\bar{h}(x) = \frac{1}{T}\sum_{t=1}^{T} h_t(x) \tag{2}$$

The expression above averages the adjusted path lengths over T trees. The normalisation term for a leaf containing m observations is given below. H_k denotes a harmonic sum, and c(m) is taken as zero when $m \leq 1$.

$$c(m) = 2H_{m-1} - \frac{2(m-1)}{m}, \quad m \geq 2$$
$$H_k = \sum_{j=1}^{k} \frac{1}{j} \tag{3}$$

For a subsample size $\psi \geq 2$, the original IF anomaly score is defined as:

$$s_{IF}(x;\ \psi)\ =\ 2^{-\frac{\bar{h}(x)}{c(\psi)}},\ \ \psi\ \geq\ 2 \tag{4}$$

Under this convention, shorter paths produce higher anomaly scores. A mean path length of c(ψ) gives a score of 0.5, which is not an attack probability. Software implementations may alter the sign or zero point of the score, so its direction must be checked. The number of trees T and subsample size ψ affect both stability and computational cost. The threshold used to generate alerts depends on the organisation's calibration data and investigation capacity (Liu, Ting & Zhou, 2008, 2012).

IF can provide a baseline for examining broad deviations in numerical features. Its training cost is approximately O(Tψ log ψ), with an expected scoring cost of O(T log ψ) per observation under balanced-tree assumptions. Irrelevant variables may nevertheless reduce the proportion of useful splits. Standard axis-aligned partitions can also produce undesirable score patterns when the data are rotated or relationships change. Extended Isolation Forest addresses this geometric issue. These limitations do not imply that standard IF cannot detect any relational anomalies. Its suitability depends on the geometry of the data (Liu, Ting & Zhou, 2012; Hariri, Carrasco Kind & Brunner, 2021).

One-Class Support Vector Machine

The one-class support vector machine (OCSVM) estimates the support of a distribution by mapping observations into a feature space through a kernel. When trained on examples that represent normal behaviour, it identifies new observations outside that region as candidates for investigation. With feature map Φ, weight vector w, slack variables ξ_i and offset ρ_{svm}, the optimisation problem is written as follows (Schölkopf et al., 2001):

$$\min_{w,\rho svm,\xi}\ \frac{1}{2}\|w\|^2\ +\ \frac{1}{\nu n}\sum_{i=1}^{n}\xi_i\ -\ \rho_{svm} \tag{5}$$

$$\langle w, \Phi(x_i) \rangle \geq \rho_{svm} - \xi_i$$
$$\xi_i \geq 0, \quad i = 1, \ldots, n \tag{6}$$

Under appropriate sampling conditions, $\nu \in (0,1]$ places an upper bound on the fraction of boundary violations and a lower bound on the fraction of support vectors. It does not directly specify the organisation's true attack rate. For an RBF kernel, γ determines the scale at which similarity between observations is assessed. Using α_i for the dual coefficients, the continuous decision value and its sign-based label are:

$$g(x) = \sum_{i=1}^{n} \alpha_i \exp[-\gamma \|x_i - x\|^2] - \rho_{svm}$$
$$f(x) = sign(g(x)) \tag{7}$$

Values of $g(x) < 0$ lie outside the learned support, while $g(x) > 0$ indicates its interior. The quantity $-g(x)$ can be used to rank observations for investigation. Retaining only the sign discards the finer ordering of the scores. Feature scaling and γ must be considered together: a kernel that is too narrow or too broad can obscure useful behavioural distinctions. The representativeness and contamination of the training data also affect the location of the boundary (Schölkopf et al., 2001).

Storing a dense kernel matrix may require $O(n^2)$ memory. Training time depends on the size of the dataset, the parameters and the solver, making frequent retraining of the conventional kernel model potentially costly for very large log collections. This does not rule out every form of online use. Approximate and incremental alternatives involve computational and accuracy trade-offs that differ from those of the conventional model. Training cost and the cost of scoring each new record should be measured separately in a SIEM deployment (Schölkopf et al., 2001; Zimek, Schubert & Kriegel, 2012).

Local Outlier Factor and local density

Local Outlier Factor (LOF) evaluates a record against its neighbours rather than against a global average. It compares local densities derived from reachability distances. An observation in a less dense region than its neighbours receives a higher outlier score (Breunig et al., 2000). For SIEM data, this allows user or server activity to be assessed within local groups of different densities. A rarely used service need not be considered anomalous simply because its behaviour differs from that of the system as a whole.

The usefulness of LOF depends on whether the neighbourhood is meaningful. Features such as request count, session duration and transferred bytes need compatible scales, and the contribution of categorical fields to distance must be defined. A very small neighbour count k can make scores sensitive to local fluctuations, whereas a large k may conceal small behavioural groups. The choice of k should therefore take dataset size into account. Density differences alone do not establish an attack: a new service or an infrequent legitimate transaction may also receive a high score. Detecting outliers within the training set and scoring records that arrive later are distinct uses. The conditions under which the selected LOF implementation supports new observations must be checked.

Clustering with K-means and DBSCAN

Clustering groups similar SIEM records to make them easier to investigate. K-means assigns observations to a specified number k of centres and seeks to reduce within-cluster distances. Records far from a centre may warrant attention, but distance alone is not evidence of an attack. Numerical features need scaling, the number of clusters needs justification and the stability of the groups should be examined across different initialisations. The preference for

similarly sized, roughly spherical groups can make differently shaped traffic patterns difficult to represent.

DBSCAN does not require a predefined number of clusters. It forms groups using a neighbourhood radius eps and a density requirement MinPts, labelling observations that cannot be connected to a sufficiently dense group as noise (Ester et al., 1996). This can be useful for irregularly shaped behavioural groups. A single radius may, however, work poorly when densities vary, making scaling and distance selection especially consequential. A noise label is not an attack label, and a cluster identifier is not a continuous risk score. A deployment must also specify how new SIEM records are associated with existing groups. Standard batch DBSCAN does not provide online learning automatically.

Reconstruction with autoencoders

An autoencoder maps an input vector to an intermediate representation through an encoder and reconstructs it through a decoder. For anomaly analysis, the working assumption is that reference behaviour can be reconstructed reasonably well, while departures from it may produce larger errors. Kitsune is an example of a network intrusion detection system that uses ensembles of autoencoders with flow features. Its use and performance should not be generalised to all SIEM products (Mirsky et al., 2018).

For encoder $e\theta$, decoder $d\varphi$ and feature dimension d, the mean squared reconstruction error of a numerical input can be expressed as:

$$E(x) = \frac{1}{d}\sum_{j=1}^{d}\left(x_j - \hat{x}_j\right)^2, \quad \hat{x} = d_{\varphi}\left(e_{\theta}(x)\right) \tag{8}$$

Network width, representation size and training duration determine model capacity. Reference-data quality and control of overfitting matter: a highly flexible network may also reconstruct unusual observations with little error. Features on different scales do

not contribute equally to squared error. The loss function should suit the feature types, and the treatment of categorical fields needs to be explained. A score threshold should be assessed during a calibration period separate from training-loss estimation. Computing requirements depend on the architecture. An autoencoder does not necessarily require a GPU or support online learning by default (Mirsky et al., 2018; Chandola, Banerjee & Kumar, 2009).

Complementary Tools for Representation and Relationship Analysis

Dimensionality reduction with PCA

Principal component analysis (PCA) finds the linear directions that explain the greatest variance in centred data. It can provide a lower-dimensional representation when many network-log variables carry similar information. If their scales differ, variables with large numerical ranges may dominate the components. The number of retained components should be assessed both by explained variance and by whether the security-relevant behaviour is preserved (Jolliffe & Cadima, 2016).

PCA may precede another model or support outlier analysis through reconstruction residuals outside the retained subspace. High variance does not, however, imply normal behaviour, nor does low variance imply an attack. Security-relevant information lost in a linear representation cannot be recovered by the downstream algorithm. The transformation must be learned on the training data and then applied unchanged to later periods (Jolliffe & Cadima, 2016; Zimek, Schubert & Kriegel, 2012).

Context-dependent relationships with CCA and RFCCA

Canonical correlation analysis (CCA) maximises the correlation between linear combinations of two variable blocks measured on the same observations. One block might describe flow

volume and the other timing features. CCA summarises their relationship, but anomaly detection requires an additional decision criterion. Sample size, the suitability of the covariance structure and the risk of overfitting in high dimensions all need examination (Hotelling, 1936).

Random forest-based conditional canonical correlation analysis (RFCCA) uses random forests to estimate how canonical correlation between two blocks varies with contextual covariates. Such conditioning may be useful when a relationship differs across server roles or working hours. The method was developed for conditional correlation estimation. It should not be treated as a ready-made SIEM attack classifier or a temporal drift-monitoring algorithm. Deployment considerations include the reliability of context variables, local sample sizes and computational cost (Alakuş et al., 2021).

Fuzzy inference as decision support

Fuzzy sets and Mamdani inference allow different signals to be combined through linguistic rules defined by experts. An anomaly score and the organisational importance of an asset, for example, can be considered together when prioritising investigations. Input scales, membership functions and the rationale for the rules must be explicit. A membership degree cannot be interpreted as an attack probability (Zadeh, 1965; Mamdani & Assilian, 1975).

A system governed by fixed expert rules serves a different purpose from the unsupervised learning algorithms reviewed here. IF, OCSVM and LOF model deviations in the data, while fuzzy inference can use their outputs to support decisions. This distinction needs to be maintained in the comparison. A growing rule base and the resulting maintenance burden are further practical limitations.

Comparing Methods by Data and Deployment Conditions

Table 1 compares the main methods by the deviations they assess, their data and deployment requirements and their principal limitations. It sets out differences that can inform model selection. A numerical performance ranking would require the methods to be evaluated on the same data under the same preprocessing and evaluation conditions.

Table 1. Data and deployment requirements of the main methods

Method	Deviation assessed	Data and deployment requirements	Main limitations
IF	Short isolation path	Numerical features; tree count and subsample size	Irrelevant dimensions and partition geometry
OCSVM	Outside the estimated support	Representative normal reference; scaling; ν and γ	Contamination and kernel training cost
LOF	Lower density than neighbours	Meaningful distances; scaling; neighbour count	Neighbourhood choice and high dimensionality
K-means	Distance from a cluster centre	Numerical features; cluster count and initialisation	Cluster geometry; additional score and threshold
DBSCAN	No link to a dense group	Distance; eps and MinPts; policy for new records	Varying density; noise does not imply an attack
Autoencoder	Reconstruction error	Representative reference; architecture, loss and threshold	Overfitting; accurate reconstruction of anomalies

Source: Qualitative comparison based on Liu, Ting & Zhou (2008, 2012), Schölkopf et al. (2001), Breunig et al. (2000), Ester et al. (1996), Mirsky et al. (2018) and Scikit-learn developers .

These requirements should be considered together. LOF may be worth examining where local behavioural groups have different densities, and OCSVM where a normal reference can be defined in advance. Clustering suits the task of grouping similar records, while an autoencoder is a candidate when multivariate reconstruction is of

interest. IF offers a baseline for broader deviations. These are conditional reasons for considering a method, not validated performance findings for a particular organisation.

Computational demands cannot be inferred from algorithm names alone. Neighbour searches in LOF, kernel structure in OCSVM, distance calculations in clustering and network architecture in autoencoders all affect cost. Dataset size, indexing, batch or stream processing and software implementation can change that cost further. Training time, peak memory use and per-event scoring latency should be reported separately. Where dimensionality reduction is added, its information loss and computational overhead also belong in the comparison (Zimek, Schubert & Kriegel, 2012; Sommer & Paxson, 2010).

Integration into SIEM Workflows and Model Monitoring

Training, calibration and evaluation should use separate, successive periods. The model and transformations such as scaling are learned during training. A later calibration period is used to choose the threshold, and final evaluation uses a time interval excluded from both decisions. Data splitting must respect dependencies between observations, since records from the same session can leak information across periods. During live operation, scoring a new event is a different task from retraining the model. Low scoring latency therefore says little about retraining speed. Memory requirements, training time and scoring cost at the daily event volume should be assessed separately when selecting a model (Sommer & Paxson, 2010).

Scores from different models should not be compared directly until their direction is aligned. Under the convention used here, a higher IF score indicates greater abnormality, whereas the OCSVM continuous decision value has the opposite direction. Its negative can therefore be used for anomaly ranking. The definition

of the score returned by the software must also be checked. Threshold selection should account for analyst capacity and the acceptable cost of missed threats, as well as the score distribution. An alert budget can be established by tracking daily alert counts and the number of records that can be investigated during calibration. Where labels are available, the share of confirmed incidents within that budget and the threats missed should also be assessed. This makes the trade-off between investigation workload and security coverage visible (Alahmadi, Axon & Martinovic, 2022).

Production monitoring should cover alerts, score distributions and feature changes. Software updates, new user groups and seasonal workloads can alter behaviour. A shift in input distribution differs from a change in its security implications. Distinguishing them requires operational knowledge and confirmed incident records (Gama et al., 2014; Lu et al., 2019). Retraining automatically after every substantial change may incorporate malicious activity into the reference data. Before an update, data quality and event context need review, and the new version should be evaluated on a separate period. Retaining the data interval, feature schema, transformations, parameters, threshold and model version makes updates traceable and supports rollback.

Model outputs are enriched with identity, endpoint and application records for analyst review. Context should explain which behaviour was unusual and why, allowing the analyst to assess whether the event is malicious (Sommer & Paxson, 2010; Vielberth et al., 2020). These decisions can inform model revision, but labelling only high-scoring alerts introduces selection bias. Events outside the queue cannot be assumed safe. Reviewing samples across score ranges and recording the source of feedback help assess that bias. Alongside score statistics, evaluation should consider confirmed incidents, investigation time and operational workload (Alahmadi, Axon & Martinovic, 2022).

Conclusion

Unsupervised learning gives SIEM workflows a way to examine rare behaviour and deviations that are difficult to capture with known rules. Choosing a method requires more than an assessment of its mathematical formulation. The unit of observation, feature quality, representation of normal behaviour and daily log volume directly affect feasibility. IF isolates observations, while OCSVM estimates distributional support. Their different treatment of data geometry means that they need not respond equally to the same anomalies. LOF focuses on local density, clustering on behavioural groups and autoencoders on reconstruction error. Relationship-analysis tools such as CCA and RFCCA supply complementary information about dependence between variable blocks. A useful comparison must keep these functions distinct.

Model choice should reflect the organisation's data sources and investigation capacity. Separate calibration, temporal evaluation and an explicit alert budget help establish whether outputs are usable. Data quality, score distributions and operating conditions require monitoring, and updates should be recorded and reversible. Scores guide investigation, while security decisions require independent evidence and expert judgement. Unsupervised learning should therefore be assessed by its contribution to event visibility and investigation, beyond a single performance measure.

References

Alahmadi, B. A., Axon, L., & Martinovic, I. (2022). 99% false positives: A qualitative study of SOC analysts' perspectives on security alarms. *31st USENIX Security Symposium (USENIX Security 22), 2783*–2800. https://www.usenix.org/conference/usenixsecurity22/presentation/alahmadi

Alakuş, C., Larocque, D., Jacquemont, S., Barlaam, F., Martin, C.-O., Agbogba, K., … Labbe, A. (2021). Conditional canonical correlation estimation based on covariates with random forests. *Bioinformatics, 37*(17), 2714–2721. https://doi.org/10.1093/bioinformatics/btab158

Axelsson, S. (2000). The base-rate fallacy and the difficulty of intrusion detection. *ACM Transactions on Information and System Security, 3*(3), 186–205. https://doi.org/10.1145/357830.357849

Bhatt, S., Manadhata, P. K., & Zomlot, L. (2014). The operational role of security information and event management systems. *IEEE Security & Privacy, 12*(5), 35–41. https://doi.org/10.1109/MSP.2014.103

Breunig, M. M., Kriegel, H.-P., Ng, R. T., & Sander, J. (2000). LOF: Identifying density-based local outliers. *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, 93*–104. https://doi.org/10.1145/342009.335388

Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection: A survey. *ACM Computing Surveys, 41*(3), Article 15. https://doi.org/10.1145/1541880.1541882

Ester, M., Kriegel, H.-P., Sander, J., & Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, 226*–231. AAAI Press. https://cdn.aaai.org/KDD/1996/KDD96-037.pdf

Gama, J., Žliobaitė, 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

Hariri, S., Carrasco Kind, M., & Brunner, R. J. (2021). Extended isolation forest. *IEEE Transactions on Knowledge and Data Engineering, 33*(4), 1479–1489. https://doi.org/10.1109/TKDE.2019.2947676

Hotelling, H. (1936). Relations between two sets of variates. *Biometrika, 28*(3–4), 321–377. https://doi.org/10.1093/biomet/28.3-4.321

Jolliffe, I. T., & Cadima, J. (2016). Principal component analysis: A review and recent developments. *Philosophical Transactions of the Royal Society A, 374*(2065), Article 20150202. https://doi.org/10.1098/rsta.2015.0202

Liu, F. T., Ting, K. M., & Zhou, Z.-H. (2008). Isolation forest. *2008 Eighth IEEE International Conference on Data Mining, 413–*422. https://doi.org/10.1109/ICDM.2008.17

Liu, F. T., Ting, K. M., & Zhou, Z.-H. (2012). Isolation-based anomaly detection. *ACM Transactions on Knowledge Discovery from Data, 6*(1), Article 3. https://doi.org/10.1145/2133360.2133363

Lu, J., Liu, A., Dong, F., Gu, F., Gama, J., & Zhang, G. (2019). Learning under concept drift: A review. *IEEE Transactions on Knowledge and Data Engineering, 31*(12), 2346–2363. https://doi.org/10.1109/TKDE.2018.2876857

Mamdani, E. H., & Assilian, S. (1975). An experiment in linguistic synthesis with a fuzzy logic controller. *International Journal of Man-Machine Studies, 7*(1), 1–13. https://doi.org/10.1016/S0020-7373(75)80002-2

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

MontazeriShatoori, M., Davidson, L., Kaur, G., & Lashkari, A. H. (2020). Detection of DoH tunnels using time-series classification of encrypted traffic. *2020 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech), 63–*70. https://doi.org/10.1109/DASC-PICom-CBDCom-CyberSciTech49142.2020.00026

Rescorla, E. (2018). *The Transport Layer Security (TLS) protocol version 1.3 (RFC 8446)*. RFC Editor. https://doi.org/10.17487/RFC8446

Schölkopf, B., Platt, J. C., Shawe-Taylor, J., Smola, A. J., & Williamson, R. C. (2001). Estimating the support of a high-dimensional distribution. *Neural Computation, 13*(7), 1443–1471. https://doi.org/10.1162/089976601750264965

Scikit-learn developers. (n.d.-a). *Clustering*. Scikit-learn documentation. Retrieved October 17, 2026, from https://scikit-learn.org/stable/modules/clustering.html

Scikit-learn developers. (n.d.-b). *Novelty and outlier detection*. Scikit-learn documentation. Retrieved October 17, 2026, from https://scikit-learn.org/stable/modules/outlier_detection.html

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

Vielberth, M., Böhm, F., Fichtinger, I., & Pernul, G. (2020). Security operations center: A systematic study and open challenges. *IEEE Access, 8*, 227756–227779. https://doi.org/10.1109/ACCESS.2020.3045514

Zadeh, L. A. (1965). Fuzzy sets. *Information and Control, 8*(3), 338–353. https://doi.org/10.1016/S0019-9958(65)90241-X

Zimek, A., Schubert, E., & Kriegel, H.-P. (2012). A survey on unsupervised outlier detection in high-dimensional numerical data. *Statistical Analysis and Data Mining, 5*(5), 363–387. https://doi.org/10.1002/sam.11161

CHAPTER 2

GRAPH-BASED CYBER-ATTACK DETECTION: REPRESENTATION STRATEGIES, TOPOLOGICAL ANALYSIS, AND COMPUTATIONAL EFFICIENCY

ALİ MELİH KANCA[1]
İLKER TÜRKER[2]

Abstract

Graph-based approaches provide a flexible framework for cyber-attack detection by representing structural relationships among hosts, flows, events, system entities, and temporally ordered observations. This chapter presents an integrated methodological perspective on graph-based cyber-attack detection, focusing on the interplay among graph construction, representation strategy, topological characterization, interpretability, and computational efficiency. Two principal representation strategies are considered: handcrafted representations based on explicitly defined graph-theoretic descriptors and automated representation learning based on

[1] PhD, Karabük University Department of Computer Engineering, Orcid: 0000-0003-3271-7463
[2] Prof. Karabük University Department of Computer Engineering, Orcid: 0000-0001-7577-4658

graph embeddings and graph neural networks. Natural Visibility Graphs (NVGs) are examined as a focused example of how sequential network-traffic observations can be transformed into graph structures and characterized through interpretable topological properties. Evidence from published and publicly available studies is synthesized to demonstrate how graph topology can reveal attack-specific structural patterns and how the relevance of graph-derived descriptors can be systematically assessed. Feature-importance analysis, representation reduction, validation, and performance–efficiency trade-offs are subsequently discussed as complementary components of graph-based model design. Overall, the chapter provides an integrated framework that connects graph construction, structural representation, topological interpretation, and computational considerations for the development of predictive, interpretable, and computationally efficient cyber-attack detection systems.

Introduction

The increasing scale, complexity, and heterogeneity of modern networked systems have made the timely detection of malicious activities a fundamental requirement for cybersecurity. Network environments continuously generate large volumes of traffic containing interactions produced by legitimate users, applications, services, and potentially malicious actors. Within these environments, cyber-attacks may manifest through changes in communication behavior, packet and flow characteristics, temporal activity patterns, or interactions among network entities. Consequently, intrusion detection has evolved from predominantly rule- and signature-based mechanisms toward data-driven approaches capable of identifying complex patterns associated with both known and previously unseen malicious behavior.

Machine-learning (ML) and deep-learning (DL) techniques have become important components of this transition. Conventional ML algorithms can learn decision boundaries from engineered traffic features, whereas deep-learning architectures can model more complex nonlinear relationships directly from high-dimensional input representations. The increasing use of convolutional neural networks (CNNs), recurrent architectures, autoencoders, transformers, and related learning models has demonstrated that the predictive capability of an intrusion-detection system depends not only on the selected classification algorithm but also on the way in which the underlying security data are represented. A highly expressive classifier may still be constrained when the input representation does not adequately capture the behavioral or structural characteristics associated with malicious activity.

Traditional flow-based intrusion-detection approaches generally represent each network flow as a vector composed of statistical or protocol-related attributes. These attributes may include packet counts, byte counts, packet-length statistics, inter-arrival times, flow duration, forward and backward traffic characteristics, and other numerical descriptors. Such representations are computationally convenient and have been successfully employed in numerous machine-learning and deep-learning studies. However, a conventional feature vector primarily describes the attributes of an individual observation. Relationships among observations, communicating entities, temporally adjacent events, or interacting processes are not necessarily represented explicitly.

This limitation is particularly relevant because many cyber-attacks are inherently relational or structural. Distributed denial-of-service activity, for example, may involve coordinated communication patterns among multiple sources and a common target. Botnet behavior can be reflected through repeated or coordinated interactions among compromised devices. Advanced

threats may consist of sequences of causally related system events, while malware behavior may be characterized through relationships among functions, system calls, or application programming interface interactions. Similarly, temporal changes in individual network-traffic features may contain structural information that is difficult to express through isolated scalar values. These characteristics motivate representations capable of preserving relationships and structural organization in addition to conventional statistical information.

Graph-based modeling provides a natural framework for this purpose. A graph represents a system through a set of nodes and edges, where nodes correspond to entities or observations and edges describe relationships among them. Depending on the cybersecurity problem, nodes may represent hosts, Internet Protocol addresses, users, processes, functions, packets, flows, events, or temporally ordered observations. Edges can represent communication, dependency, invocation, causality, similarity, temporal association, or mathematically defined structural relationships. Graph construction therefore provides a flexible mechanism for transforming cybersecurity data into representations in which relational information becomes explicitly available for analysis (Caville et al., 2022; Daya et al., 2019; Islam et al., 2020; Protogerou et al., 2021; Zola et al., 2022).

An important characteristic of graph-based cybersecurity analysis is that there is no single universally applicable graph representation. Different graph construction strategies emphasize different properties of the underlying data. Communication graphs can model interactions among hosts or network endpoints, while flow graphs can characterize relationships associated with network flows. Function-call graphs represent behavioral relationships among software functions and can therefore support malware-oriented analysis. Provenance graphs provide a mechanism for

modeling causal relationships among system entities and events and are particularly relevant to multi-stage or advanced threats. Dynamic and temporal graphs extend these representations by explicitly incorporating changes in nodes, edges, or attributes over time.

Visibility graphs provide another perspective by transforming numerical sequences into graph structures (Iacovacci & Lacasa, 2016; Lacasa et al., 2008). Instead of defining nodes as communicating hosts or system entities, visibility-graph approaches treat individual observations within an ordered sequence as graph nodes and establish edges according to geometrically defined visibility relationships. Natural Visibility Graphs (NVGs), in particular, provide a mechanism for preserving aspects of the ordering and structural variation of a numerical sequence while enabling that sequence to be examined using graph-theoretic measures. This characteristic makes NVGs relevant to network-traffic analysis, where numerical flow features can be interpreted as sequential observations and subsequently transformed into graph structures (Kanca & Türker, 2026).

Once cybersecurity data have been represented as graphs, the resulting structures can be analyzed through different representation strategies. One approach is to calculate explicitly defined, or handcrafted, graph descriptors. Measures such as degree, graph density, clustering coefficient, shortest-path characteristics, betweenness centrality, closeness centrality, eigenvector centrality, modularity, graph diameter, and community-related measures summarize different structural properties of a graph. These descriptors can subsequently be supplied to conventional ML or DL classifiers and offer an important advantage in terms of structural interpretability because each feature corresponds to a defined graph-theoretic property.

A second strategy is automated representation learning, in which informative graph representations are learned directly from

the data rather than being completely specified through manually selected graph metrics. Graph embeddings, graph neural networks (GNNs), graph convolutional networks, graph attention mechanisms, and related architectures can learn latent representations by aggregating structural and attribute information. Handcrafted and automated representation strategies involve different trade-offs among predictive flexibility, computational complexity, feature engineering requirements, and interpretability.

The methodological diversity of graph-based cyber-attack detection is also reflected in the existing literature. Previous systematic analysis has shown that graph-based approaches have been investigated across multiple cybersecurity application domains using different graph construction, feature extraction, and learning strategies. These studies include both explicitly defined graph-based features and data-driven representation-learning approaches, highlighting that representation strategy constitutes an important methodological design choice in graph-based cyber-attack detection (Kanca & Türker, 2025).

The diversity of graph representations also introduces an additional methodological problem. Constructing a graph does not by itself determine which structural information is useful for cyber-attack detection. When handcrafted graph representations are employed, numerous topological metrics can potentially be extracted from the same graph. Although these metrics characterize complementary properties of graph structure, their contribution to attack classification may differ considerably. Furthermore, calculating a large number of graph metrics can substantially increase computational cost, particularly when graphs are generated repeatedly across multiple traffic features and temporal windows. Consequently, graph-based intrusion detection requires consideration not only of how the graph is constructed, but also of which structural descriptors should be retained.

Interpretability represents another important dimension of this problem. A graph-based representation may achieve high classification performance without necessarily explaining which structural characteristics distinguish malicious traffic from benign traffic. Examining attack-specific topological patterns can help bridge this gap. Instead of treating graph-derived features exclusively as inputs to a classifier, their distributions can be analyzed to determine whether particular attacks are associated with systematic changes in connectivity, clustering, centrality, path structure, or community organization. Such analysis provides a connection between predictive modeling and structural characterization of network behavior.

Previous NVG-based analysis provides evidence for this perspective. In the attack-specific topological study by Kanca et al.(Kanca and Türker, 2026) 76 numerical network-traffic features were independently transformed into NVGs, and ten graph-theoretic metrics were calculated from each representation, resulting in 760 topological descriptors per traffic frame. Beyond evaluating the discriminative capability of these representations, the study examined attack-versus-benign differences across traffic-feature–topology combinations. Of 10,640 comparisons, 7,777 remained statistically significant following false discovery rate correction, and 4,844 exhibited large effect sizes. The findings indicated that attack-related structural differences were distributed across multiple combinations of traffic characteristics and graph metrics rather than being represented by a single universal topological descriptor.

This observation naturally raises a further methodological question: if a large collection of graph descriptors contains potentially discriminative information, are all of those descriptors required for effective detection? Different topological metrics may capture overlapping structural characteristics, while their computation can increase the overall cost of graph-based analysis.

Feature-importance and feature-selection methods therefore provide a potential mechanism for identifying informative descriptors, reducing redundancy, and constructing more compact graph representations. This consideration establishes a direct connection between structural interpretability and computational efficiency.

Taken together, these observations suggest that graph-based cyber-attack detection should not be formulated solely as a classification problem. It can instead be viewed as a sequence of interrelated representation and design decisions. The raw security data must first be associated with an appropriate graph construction mechanism. The resulting graph must then be represented using handcrafted or automatically learned information. When handcrafted topology is employed, relevant structural characteristics must be identified and interpreted. Redundant or computationally expensive information may subsequently need to be reduced, and the resulting representation must ultimately be evaluated in terms of predictive performance, robustness, interpretability, and computational efficiency.

Accordingly, this chapter conceptualizes graph-based cyber-attack detection as an integrated design pipeline rather than as a single model-selection problem. The central question addressed throughout the chapter is: “How can cybersecurity data be transformed into graph representations, structurally characterized, interpreted, reduced, and evaluated to support predictive, interpretable, and computationally efficient cyber-attack detection?”

To address this question, the chapter connects three complementary levels of analysis. The first is the representation level, which considers graph construction alternatives and handcrafted and automated representation strategies. The second is the structural level, which examines how graph topology can characterize network traffic and reveal attack-specific patterns, with particular emphasis on NVG-based representations. The third is the

metric and efficiency level, which considers the relative importance of topological descriptors, representation reduction, predictive performance, and computational cost.

The contribution of the chapter therefore lies primarily in methodological integration and synthesis. It does not present previously reported experiments as newly generated evidence. Instead, selected findings from the authors' published and publicly available prior work are used to support the relevant methodological discussions. These findings are considered within a broader methodological framework that connects graph construction, representation learning, topological interpretation, feature importance, representation reduction, computational efficiency, and model evaluation. This organization provides a basis for understanding not only whether graph-based methods can classify cyber-attacks, but also what structural information they capture, how that information can be interpreted, which components may be informative, and what methodological considerations are required for practical deployment.

Finally, the chapter identifies several issues that remain central to the development of graph-based intrusion-detection systems, including cross-dataset generalization, class imbalance, graph-construction complexity, computational scalability, explainability, and dynamic graph modeling. By considering these issues within a common design framework, the chapter aims to provide a structured foundation for both interpreting existing graph-based cyber-attack detection research and designing future systems.

Graph-Based Representation of Cyber-Security Data

Why Graphs?

A graph G=(V,E) represents a set of nodes V and relationships E. In cyber-security, a node can represent a host, IP address, user, process, packet, flow, function, event, or time-indexed

observation. An edge can encode communication, similarity, dependency, temporal visibility, invocation, causality, or another domain-specific relation. This flexibility makes graph modeling particularly useful when malicious behavior is expressed through interactions rather than isolated feature values.

Graph construction is therefore not a neutral preprocessing step. The choice of nodes and edges determines which structural properties are available to downstream algorithms. A host-communication graph emphasizes connectivity and interaction patterns, whereas a function-call graph captures program behavior. A visibility graph instead maps a numerical sequence into a network and is therefore suited to analyzing the structural organization of sequential observations.

Major Graph Construction Strategies

Graph construction is one of the fundamental stages of graph-based cyber-security analysis because it determines how the entities and relationships contained in the original data are represented structurally. Cyber-security data can originate from heterogeneous sources, including network flows, packet sequences, communication records, system events, application behavior, and temporal observations. Consequently, the definition of nodes and edges may vary considerably depending on the characteristics of the data and the security problem being investigated. Different graph construction strategies therefore emphasize different aspects of malicious behavior.

In network-oriented security applications, communication graphs provide one of the most intuitive representations. In this approach, nodes typically correspond to communicating entities such as hosts or IP addresses, while edges represent communication between these entities. Such a representation shifts the analysis from individual traffic records toward the interaction structure of the

network. For example, unusual connectivity patterns, repeated interactions, or changes in communication behavior may provide useful structural information for identifying intrusion, botnet activity, or lateral movement (Yao et al., 2019).

A related approach is the flow graph, which focuses more directly on network-flow information. Depending on the particular design, nodes may represent flows, communicating endpoints, or other flow-related entities, while edges describe relationships among them. Flow graphs can incorporate information derived from source and destination addresses, ports, protocols, traffic direction, or other flow characteristics. They are therefore particularly relevant to network intrusion and denial-of-service analysis, where malicious activity may produce characteristic communication and traffic patterns (Yao et al., 2018).

Visibility graphs follow a substantially different construction principle. Rather than directly representing communicating entities, visibility-based methods transform an ordered numerical sequence into a graph. Individual observations are represented as nodes, and edges are established according to predefined visibility criteria. Variants such as the Visibility Graph (VG), Horizontal Visibility Graph (HVG), and Natural Visibility Graph (NVG) provide different mechanisms for determining these relationships. In cyber-security applications, numerical network-traffic features can be considered as sequential observations and transformed into visibility graphs. The resulting graph structure can then be characterized through topological metrics, allowing structural properties of changing traffic behavior to be examined.

Outside conventional network-traffic analysis, function-call graphs provide a useful representation of program behavior. Nodes typically correspond to functions, procedures, or API calls, while edges indicate invocation relationships. Instead of describing network communication, these graphs characterize how software

components interact during execution or within program structure. Consequently, function-call graphs are particularly relevant to malware analysis, where malicious and benign programs may exhibit different patterns of function or API usage (Z. Zhao et al., 2013).

Another important representation is the provenance graph. Provenance-oriented approaches model relationships among system entities and events, with edges representing dependencies, information flows, or causal relationships. Such graphs can provide a broader view of how activities propagate through a system. This is particularly useful when an attack cannot be sufficiently characterized by a single isolated event. Multi-stage attacks and advanced persistent threats may involve sequences of related actions, and provenance graphs can support the reconstruction and analysis of these attack chains (Li et al., 2022).

Cyber-security behavior is also inherently time-dependent. For this reason, dynamic or temporal graphs extend conventional static graph representations by incorporating changes in nodes, edges, attributes, or interactions over time. Instead of representing network behavior through a single fixed structure, these approaches can capture how relationships evolve. Temporal graph representations may therefore be useful for detecting attacks characterized by gradual behavioral changes, recurring interactions, or sequences of malicious activities (J. Zhao et al., 2026).

These graph construction strategies are summarized in Table 1.

Table 1. Major graph construction strategies in cyber-security analysis.

Graph type	Typical unit/relationship	Potential cyber-security use
Communication graph	Hosts/IPs and communication links	Intrusion, botnet, lateral-movement analysis
Flow graph	Flows or communicating endpoints	Network intrusion and DDoS characterization
Visibility graph (VG/HVG/NVG)	Sequential observations linked by visibility	Structural analysis of time-varying traffic features
Function-call graph	Functions/API calls and invocation edges	Malware and program-behavior analysis
Provenance graph	Entities/events connected by causal provenance	Advanced persistent threat and attack-chain analysis
Dynamic/temporal graph	Time-evolving nodes and edges	Changing network behavior and temporal attacks

As shown in Table 1, graph construction strategies differ primarily in what is represented as a node and what type of relationship is represented as an edge. This distinction is methodologically important because different graph constructions preserve different forms of information. A communication graph emphasizes interactions among network entities, whereas a function-call graph emphasizes relationships within software behavior. Provenance and temporal graphs provide information about causal or time-dependent relationships, while visibility graphs reveal structural characteristics embedded in sequential numerical observations.

Therefore, the selection of a graph construction strategy should be aligned with both the characteristics of the available security data and the objective of the detection task. There is no single graph structure that is inherently appropriate for every cyber-

security problem. Instead, graph construction determines which structural characteristics become available for subsequent analysis and learning. After the graph has been constructed, a second methodological decision is required: how should the information contained in that graph be represented for machine-learning or deep-learning models? This question leads to handcrafted representation and automated representation learning, which are discussed in the following section.

Representation and Learning Strategies

Once cyber-security data have been transformed into a graph, the next important question concerns how the information contained in the graph should be represented for subsequent analysis and classification. Graph construction determines the entities and relationships preserved from the original data, whereas the representation strategy determines how this structural information is provided to a machine-learning or deep-learning model.

Graph-based cyber-security approaches can broadly be considered from two representation perspectives: handcrafted representation and automated representation learning. These strategies differ in the extent to which structural information is explicitly defined by the researcher or automatically learned from data. Each strategy offers different advantages in terms of interpretability, flexibility, computational requirements, and dependence on feature engineering.

Handcrafted Representation

Handcrafted graph representation describes a graph through a predefined set of structural or topological measures. Instead of directly providing the complete graph structure to a learning algorithm, relevant characteristics of the graph are calculated and organized as a feature vector. Common examples include degree-based measures, graph density, clustering coefficient, shortest-path

characteristics, graph diameter, centrality measures, modularity, and community-related properties.

These metrics characterize different aspects of graph structure. Degree-related measures describe node connectivity, whereas density provides information about the overall level of connectivity in the graph. Clustering coefficients characterize the tendency of nodes to form locally interconnected structures, while path-based measures describe aspects of network-wide organization(Watts & Strogatz, 1998). Betweenness and closeness centrality characterize different aspects of the structural position of nodes within a network(Freeman, 1978). Eigenvector-based centrality additionally considers the importance of the nodes to which a node is connected(Bonacich, 1972). Modularity and community-related measures characterize the organization of networks into groups or densely connected substructures(Newman & Girvan, 2004).

In cyber-security applications, these descriptors can transform complex relational structures into numerical representations that can subsequently be processed by conventional machine-learning or deep-learning models. One of the principal advantages of this approach is interpretability. Because each feature has a graph-theoretic definition, differences observed between benign and malicious behavior can potentially be associated with specific structural characteristics.

However, handcrafted representations also introduce important design decisions. The researcher must determine which topological measures should be calculated, and a predefined metric set may not capture every task-specific pattern contained in the original graph. Furthermore, extracting a large number of graph metrics can introduce redundant information and increase computational cost. Therefore, the selection and importance of handcrafted topological descriptors become important

considerations in the design of graph-based cyber-attack detection systems.

Automated Representation Learning

Automated representation learning follows a different principle. Rather than describing the graph exclusively through manually selected topological measures, the model learns informative representations directly from graph structure and, where available, node or edge attributes.

Graph embeddings and Graph Neural Networks (GNNs) are representative examples of this approach. Graph-learning architectures can aggregate information from neighboring nodes and learn latent representations that reflect both local and broader structural relationships. Depending on the task, these learned representations can subsequently be used for node classification, edge prediction, graph classification, anomaly detection, or other security-related objectives (Lo et al., 2022).

The principal advantage of automated representation learning is its ability to reduce dependence on manually specified graph descriptors (Caville et al., 2022; Lo et al., 2022). Potentially complex and nonlinear structural patterns can be learned directly from data, which may be particularly valuable when the relevant graph characteristics are difficult to define in advance.

However, automated representation learning also presents challenges. Learned latent features generally do not have the same direct semantic interpretation as explicitly calculated topological measures. Consequently, understanding why a particular graph or observation has been classified as malicious can become more difficult. Model complexity, computational requirements, training-data requirements, and scalability to large graphs must also be considered.

Thus, automated representation learning should not simply be interpreted as a replacement for handcrafted representation. Rather, it represents a different mechanism for extracting useful information from graph-structured cyber-security data.

Evidence from Literature

The diversity of graph representation strategies described above is also reflected in the existing cyber-security literature. Graph-based approaches have been investigated across a range of application domains, including intrusion and network intrusion detection, botnet and distributed denial-of-service detection, IoT security, malware analysis, and advanced threat detection. Across these domains, graph construction and representation are closely related to the type of security data being analyzed and the structural relationships that the detection task seeks to preserve.

Handcrafted approaches characterize graph structure through explicitly defined descriptors such as centrality, connectivity, clustering, path-based measures, and other structural properties. These representations provide a direct connection between the original graph and the numerical features supplied to a learning algorithm, which can support structural interpretation of malicious behavior. In contrast, automated representation-learning approaches, including graph embeddings and GNN-based architectures, learn latent representations directly from graph structure and associated attributes (Caville et al., 2022; Lo et al., 2022). Such approaches reduce dependence on manually specified descriptors and enable task-oriented structural patterns to be learned from data.

The literature also demonstrates that the suitability of a representation strategy depends on factors such as graph construction, data characteristics, detection objective, computational requirements, and the desired level of interpretability. Handcrafted and automated approaches therefore represent different

methodological choices for extracting and utilizing structural information from graph-based cyber-security data. Consequently, representation strategy is better understood as a methodological design choice than as a fixed component of graph-based cyber-attack detection.

From this perspective, an important distinction emerges between constructing a graph and representing the information contained in that graph. Graph construction determines which entities and relationships are preserved, whereas representation determines how this structural information becomes available to subsequent learning and analysis. This distinction provides the basis for the next part of the chapter, which focuses on Natural Visibility Graphs as a specific mechanism for transforming sequential network-traffic observations into graph structures.

Natural Visibility Graphs as a Structural Representation

Among graph construction approaches, visibility graphs provide a distinctive mechanism for converting sequential numerical data into graph structures. Unlike communication, flow, or provenance graphs, where nodes and edges generally correspond to explicitly defined entities and interactions, visibility graphs derive their connectivity directly from the geometric relationships among observations in an ordered sequence. This property makes them particularly relevant when the original data consist of numerical measurements whose temporal or sequential organization may contain discriminative information.

In network-traffic analysis, many flow-based features can be arranged as sequences of consecutive observations. Packet statistics, flow duration, inter-arrival-time characteristics, packet-length measures, and forward or backward traffic properties may exhibit different local patterns under benign and malicious conditions. Rather than analyzing these observations only as conventional

numerical values, visibility-graph transformation provides an alternative structural representation. Once the sequence has been converted into a graph, its organization can be characterized using graph-theoretic measures such as connectivity, clustering, path structure, centrality, and community organization.

From a Numerical Sequence to an NVG

Natural Visibility Graph (NVG) transformation maps an ordered numerical sequence into a graph while preserving information about the relative magnitude and organization of its observations (Lacasa et al., 2008). Consider a numerical sequence

$$\mathrm{X} = \{\mathrm{x}_1, \mathrm{x}_2, \ldots\ldots \mathrm{x}_n\}$$

where each observation x_i corresponds to a position t_i in the ordered sequence. During NVG construction, each observation is represented as a node. Therefore, a sequence containing n observations produces a graph containing n nodes(Lacasa et al., 2008).

The edges of the graph are determined according to the natural visibility criterion. Two observations x_i and x_j, where $t_i < t_j$, are connected when every intermediate observation x_k satisfies the visibility condition

$$x_k < x_i + \left(x_j - x_i\right)\frac{t_k - t_i}{t_j - t_i}, \qquad t_i < t_k < t_j.$$

Geometrically, this condition means that a straight line drawn between two observations must remain above all intermediate observations. If no intermediate observation blocks this line of sight, the corresponding nodes are connected by an edge. Adjacent observations are therefore naturally connected, while additional long-range edges are determined by the relative values of the observations located between them (Lacasa et al., 2008).

This construction mechanism converts variations in the original numerical sequence into variations in graph connectivity. Peaks, valleys, local fluctuations, and broader changes in the sequence can influence which observations are mutually visible. Consequently, two numerical sequences containing different structural patterns may generate graphs with different degree distributions, clustering behavior, path characteristics, centrality patterns, or community structures.

The important point is that NVG transformation does not merely calculate another statistical feature from the original sequence. Instead, it changes the representation domain of the data:

Numerical sequence → visibility relationships → graph structure → topological descriptors

This transformation makes it possible to examine a numerical traffic feature from a graph-theoretic perspective. Conventional statistics may summarize characteristics such as mean, variance, minimum, or maximum values, whereas an NVG-based representation enables the structural organization of observations within a local sequence to be characterized.

For network-traffic analysis, the transformation can be applied independently to multiple numerical traffic features. Suppose that a traffic dataset contains m numerical features and that a local frame contains n consecutive flow observations. For each feature, the n values within the frame can be interpreted as an independent numerical sequence and transformed into a separate NVG. A single traffic frame can therefore produce multiple graphs, each representing the local structural behavior of a different traffic characteristic (Kanca & Türker, 2026).

For example, a frame may contain sequential observations for packet-length statistics, flow duration, inter-arrival times, forward packet characteristics, and backward packet characteristics.

Rather than combining all of these values into a single graph, the feature-wise NVG approach constructs a separate graph for each feature. The resulting collection of graphs represents different structural dimensions of the same traffic interval.

This distinction is important because different traffic features may respond differently to malicious behavior. An attack may produce substantial structural changes in packet timing while having a smaller effect on another traffic characteristic. Feature-wise graph construction allows these differences to be retained instead of immediately aggregating them into a single representation.

After graph construction, topological measures can be extracted from each NVG. These measures convert the graph structure into numerical descriptors that can subsequently be used for statistical analysis, machine-learning classification, deep-learning models, or interpretability-oriented analysis. In this sense, NVG provides a bridge between sequential network-traffic data and graph-based structural analysis.

The resulting processing chain can be expressed as:

Network Traffic → Numerical Feature → Local Sequence → NVG Construction → Topological Metrics → Learning/Statistical Analysis

This framework is particularly relevant when the research objective extends beyond predictive performance. Because each traffic feature produces an explicit graph and each graph can be characterized through interpretable topological measures, it becomes possible to investigate not only whether malicious traffic can be classified, but also which traffic characteristics and which structural graph properties distinguish attack behavior from benign behavior.

Framing and Labeling Considerations

Before NVGs can be constructed from network-traffic data, the continuous or ordered sequence of flow observations must generally be divided into manageable local segments. A common approach is sliding-window framing, in which consecutive observations are grouped into fixed-length frames.

Two parameters are particularly important in this process: window length (WL) and step size. Window length determines the number of consecutive observations included in each frame and, consequently, the number of nodes in each feature-specific NVG. Step size determines how far the window moves before the next frame is generated.

For example, if a window contains 100 observations and the step size is 50, two consecutive frames overlap by 50 observations, corresponding to 50% overlap. Overlapping windows can increase the number of available frames and allow neighboring segments of traffic behavior to be represented with greater continuity.

Window length represents an important design parameter because it determines the amount of local context represented by each graph. A relatively short window generates smaller graphs and may better preserve highly localized changes in traffic behavior. It can also reduce graph-construction and topological-feature extraction costs. However, a very short window may provide insufficient context for representing more complex structural patterns.

Conversely, larger windows contain more observations and can capture broader patterns of traffic behavior. At the same time, they produce larger graphs, potentially increasing the computational cost of graph construction and topological analysis. Large windows may also combine multiple local behaviors within the same frame. Window selection therefore involves a trade-off among

temporal/local resolution, structural information, sample availability, and computational complexity.

A second issue arises from frame labeling. In flow-based intrusion-detection datasets, individual observations typically have their own benign or attack labels. After multiple observations are grouped into a frame, however, a single label must generally be assigned to the entire frame for supervised classification.

If every observation within the frame belongs to the same class, frame labeling is straightforward. Mixed frames are more challenging because they may contain observations from multiple classes or a combination of benign and malicious traffic. Several labeling strategies can be considered. Majority voting assigns the most frequent class within the frame. An attack-prioritized strategy can assign an attack label whenever malicious observations are present, depending on the detection objective. Other application-specific rules may also be defined according to attack prevalence or temporal position.

There is no universally appropriate labeling strategy for every intrusion-detection task. The selected rule changes the semantic meaning of the resulting frame and may influence class distributions and classification results. Therefore, the labeling mechanism should be explicitly defined rather than treated as an incidental preprocessing detail.

Framing and labeling should consequently be considered integral components of NVG-based representation design. The graph presented to the learning algorithm is determined not only by the NVG visibility criterion, but also by which observations are grouped together, how much consecutive frames overlap, and how the resulting graph is labeled. These choices influence the structural information contained in the graph and the characteristics of the resulting representation.

Topological Characterization

After a numerical network-traffic sequence has been transformed into an NVG, the resulting graph must be characterized in a form suitable for statistical analysis or learning algorithms. Although the graph itself preserves structural relationships among observations, its complete adjacency structure may be difficult to interpret directly. Topological metrics provide a mechanism for summarizing different properties of this structure through numerical descriptors.

A single topological metric is generally insufficient to characterize all aspects of an NVG. Graph structure can be examined from several complementary perspectives, including overall connectivity, local organization, node centrality, path structure, and community organization. For this reason, multiple graph-theoretic measures can be extracted from each NVG to obtain a more comprehensive description of the underlying sequence.

Connectivity measures, such as average degree and graph density, describe how strongly the nodes of an NVG are connected. Average degree summarizes the typical number of connections associated with nodes, whereas density represents the proportion of possible edges that are actually present in the graph. In the context of visibility graphs, changes in these measures indicate that the visibility relationships among observations have changed. Therefore, traffic sequences with different numerical organization may produce different levels of graph connectivity.

Local organization can be characterized using measures such as the clustering coefficient. The clustering coefficient describes the extent to which neighboring nodes are also connected to one another. It therefore provides information about the formation of locally interconnected structures within the graph. In an NVG representation, changes in local variation or repeated structural

patterns within a traffic feature may affect the formation of these local connections.

Centrality measures provide another perspective by characterizing the structural position or importance of nodes within the graph. Degree centrality focuses on the number of direct connections associated with a node. Betweenness centrality reflects the extent to which a node lies on shortest paths between other nodes, while closeness centrality describes how close a node is to the remaining nodes in terms of graph distance. Eigenvector centrality additionally considers the importance of the nodes to which a particular node is connected. When these measures are summarized across an NVG, they can provide information about how structural influence or connectivity is distributed throughout the sequence.

Path-based measures characterize the global organization of the graph. Average shortest-path length represents the typical number of edges required to connect pairs of nodes through shortest paths, whereas graph diameter represents the greatest shortest-path distance within the graph. These measures provide information about the overall extent and reachability of the graph. Changes in the sequential organization of network-traffic observations may alter long-range visibility relationships and consequently influence these path characteristics.

Finally, community-related measures examine whether the graph can be separated into internally cohesive groups. Modularity evaluates the strength of community organization, while the number of detected communities indicates how many such groups are identified under a particular community-detection procedure. These measures provide a mesoscopic perspective between local node-level characteristics and global graph-level properties.

The principal metric families and their structural interpretations are summarized in Table 2.

Table 2. Principal topological metric families and their structural interpretations.

Metric family	Examples	Structural interpretation
Connectivity	Average degree, density	Overall level of graph connectivity
Local organization	Clustering coefficient	Tendency of neighboring nodes to form locally connected structures
Centrality	Betweenness, closeness, eigenvector, degree centrality	Relative structural importance or position of nodes
Path structure	Average shortest path, diameter	Global reachability and graph extent
Community structure	Modularity, number of communities	Mesoscopic organization and separation into groups

Taken together, the metric families summarized in Table 2 provide complementary descriptions of an NVG. Connectivity measures indicate how strongly the graph is connected; clustering describes local organization; centrality measures characterize the structural roles of nodes; path-based measures describe global reachability; and community measures characterize intermediate-scale organization. Consequently, the original numerical sequence is no longer represented only by its individual values but also by a set of descriptors reflecting its structural organization.

For cyber-attack detection, this transformation is particularly relevant because malicious behavior may modify the organization of network-traffic features in different ways. An attack does not necessarily produce a uniform increase or decrease in every traffic variable. Instead, it may alter packet timing, traffic direction, packet-length characteristics, flow rates, or other attributes in different patterns. When these feature sequences are transformed into NVGs, such changes may be reflected through differences in connectivity, clustering, centrality, path structure, or community organization.

Therefore, topological characterization provides two related benefits. First, the extracted metrics can serve as numerical inputs to machine-learning or deep-learning models for attack classification. Second, the same metrics can be analyzed statistically to investigate which structural properties differ between benign and malicious traffic. The latter is especially important for interpretability because it shifts the analysis from simply asking whether a model can distinguish attacks toward examining how attack behavior is reflected in graph topology.

However, extracting a large number of topological metrics also introduces an important methodological question. Different metrics may capture overlapping structural information, and their relevance may vary according to the traffic feature and attack class being examined. Thus, the availability of many graph descriptors does not imply that all of them are equally informative.

This observation motivates two subsequent questions addressed in the following sections: Do different cyber-attack classes exhibit distinguishable topological characteristics, and which topological metrics contribute most strongly to their discrimination?

Topology as an Indicator of Attack Behavior

Graph-based cyber-attack detection becomes more informative when topological representations are used not only for classification but also for understanding how malicious traffic differs structurally from benign behavior. From this perspective, attack-specific topological analysis examines whether changes in network-traffic characteristics are reflected in NVG properties such as connectivity, clustering, centrality, and path structure.

In an attack-specific NVG analysis by Kanca and Türker (Kanca and Türker, 2026), 76 numerical network-traffic features were independently transformed into NVGs, and ten graph-theoretic metrics were extracted from each feature, resulting in 760

topological descriptors per traffic frame. A multi-branch CNN achieved an average accuracy of 96.20% and an MCC of 0.9566 under stratified five-fold cross-validation. Beyond classification performance, the analysis investigated whether individual cyber-attack classes exhibited distinguishable topological patterns relative to benign traffic.

For this purpose, 14 attack classes were compared with benign traffic across 10,640 traffic-feature–topology combinations. After false-discovery-rate correction, 7,777 comparisons (73.1%) remained statistically significant, while 4,844 exhibited large Cliff's delta effect sizes. These results provide empirical evidence that malicious traffic can be associated with substantial structural differences in NVG representations.

Importantly, the strongest differences were not associated with a single universal topological metric. Instead, they emerged from specific combinations of traffic characteristics and graph properties. Features related to backward traffic, packet length, and inter-arrival behavior frequently showed strong associations, while degree, clustering, and eigenvector-centrality measures captured different forms of attack-related structural variation. These findings indicate that topological interpretation should consider the interaction between the underlying traffic feature, the selected graph descriptor, and the attack class rather than relying on a single universal metric. Representative examples are presented in Table 3.

Table 3. Representative attack-specific traffic-feature–topology signatures in NVG representations.

Attack example	Traffic feature	Topological metric	Cliff's δ
Infiltration	Fwd Pkts/s	Average Degree	-0.9910
Bot	Fwd Act Data Pkts	Average Clustering	+0.8474
FTP-BruteForce	Pkt Size Avg	Eigenvector Centrality	+0.9946
Brute Force-Web	Flow IAT Std	Average Degree	-0.9994
SQL Injection	Fwd IAT Tot	Average Degree	-0.9999
SSH-Bruteforce	Tot Bwd Pkts	Eigenvector Centrality	-0.9936

As shown in Table 3, different attacks may be associated with different traffic-feature–topology signatures. However, these findings were obtained from a single benchmark dataset, and some attack classes contained substantially fewer observations than others. Therefore, the identified signatures should be validated across additional datasets and network environments before being considered general attack-specific patterns.

It should also be noted that statistical separation between attack and benign distributions does not necessarily imply predictive importance within a classification model; these represent related but distinct analytical perspectives.

Topological Metric Importance and Explainability

Why Metric Importance Matters

Graph-based representations can produce a large number of topological descriptors, particularly when multiple graph metrics are extracted from several traffic features. Although these descriptors characterize different structural properties, they may not contribute equally to cyber-attack classification. Some metrics may contain

highly discriminative information, whereas others may provide redundant or limited additional information.

Computing a large metric set can also increase feature-extraction time, memory requirements, and model dimensionality. Therefore, metric-importance analysis has two complementary objectives. First, it improves interpretability by identifying which structural graph properties contribute most strongly to the detection task. Second, it supports representation reduction by identifying smaller subsets of potentially informative metrics. This creates a direct connection between explainability and computational efficiency.

Complementary Importance Methods

Feature importance can be evaluated from different methodological perspectives. A metric considered important by one method may receive a different ranking from another because the methods measure importance according to different principles. For this reason, combining complementary approaches can provide a broader assessment than relying on a single importance method.

Table 4. Complementary feature-importance and feature-selection methods for evaluating topological metrics.

Method	Core idea	Interpretive role
SHAP	Attributes model predictions to input features using Shapley-value principles	Model-oriented contribution
Grouped Permutation Feature Importance (GPFI)	Jointly permutes features within a predefined group and measures the resulting change in model performance	Group-level predictive dependency
Boruta	Compares real variables with randomized shadow variables	Relevance screening
Recursive Feature Elimination (RFE)	Iteratively removes less important variables	Relative ranking / subset selection

These methods answer related but different questions. SHAP focuses on the contribution of features to model predictions (Lundberg & Lee, 2017), while Grouped Permutation Feature Importance evaluates how model performance changes when the information associated with a predefined feature group is jointly disrupted (Au et al., 2022). Boruta focuses on determining whether features contain relevant information compared with randomized alternatives (Kursa & Rudnicki, 2010), whereas RFE progressively removes less important features to obtain a relative ranking or reduced subset (Guyon et al., 2002).

Integrating Importance Evidence

Because feature-importance methods evaluate relevance according to different principles, their outputs should not necessarily be expected to produce identical rankings. A topological descriptor may appear important because it strongly influences model predictions, because disrupting its information reduces predictive performance, or because it consistently provides information beyond randomized alternatives. These perspectives are related but not equivalent.

For this reason, importance analysis can benefit from considering evidence from multiple complementary methods rather than relying exclusively on a single ranking. An integrated importance analysis can consequently be used to identify candidate descriptors for subsequent representation reduction. However, importance ranking alone is not sufficient evidence that lower-ranked metrics can be removed without affecting detection capability. Any reduced representation should be evaluated again using the same predictive criteria applied to the original representation, while also considering computational cost and the stability of the selected metrics across validation partitions.

This distinction is important for efficient graph-based cyber-attack detection. Feature importance identifies potentially informative structural properties, whereas feature reduction determines whether a smaller representation can preserve sufficient information for the detection task. The transition from importance analysis to representation reduction therefore requires explicit empirical validation rather than assuming that the highest-ranked descriptors automatically constitute an optimal subset.

Performance–Efficiency Trade-Offs

The extraction of graph-theoretic descriptors introduces an additional computational stage into graph-based cyber-attack detection. As the number and complexity of topological metrics increase, the resulting representation may provide a richer structural description of network traffic, but it may also increase feature-extraction time, memory requirements, and overall processing cost. Efficient model design therefore requires considering predictive capability and computational requirements jointly rather than evaluating classification performance in isolation.

Feature-importance analysis provides a natural starting point for representation reduction. After candidate topological descriptors have been ranked or evaluated, progressively smaller metric subsets can be constructed and compared with the full representation. Such subsets may be defined according to individual importance rankings, consensus across multiple importance methods, redundancy analysis, or combinations of these criteria. The objective is not necessarily to identify the smallest possible representation, but to determine whether redundant or weakly informative descriptors can be removed without introducing an unacceptable loss of predictive information.

The evaluation of reduced representations should consider multiple dimensions. Predictive performance can be assessed using

metrics such as accuracy, weighted or macro-averaged F1-score, and Matthews correlation coefficient, while computational evaluation can include graph-construction time, topological feature-extraction time, model training time, inference time, and memory consumption. Reporting these components separately is particularly useful because the computational bottleneck in a graph-based pipeline may arise before the classification model itself is executed.

A reduced representation should also be evaluated under an experimental design that separates feature selection from final performance estimation. If the same validation data are repeatedly used both to select the most informative descriptors and to report the final performance of the reduced representation, the resulting estimate may be optimistic. Independent holdout evaluation or nested cross-validation can provide a more rigorous assessment when the available data and computational resources permit.

From this perspective, efficiency should be interpreted as a performance–cost trade-off rather than simply as dimensionality reduction. A compact representation is useful only when its computational advantages are accompanied by acceptable predictive capability and sufficient stability across validation partitions, datasets, or network environments. Consequently, the preferred representation may differ according to the operational context: offline forensic analysis may tolerate computationally expensive structural descriptors, whereas near-real-time intrusion detection may place greater emphasis on extraction and inference latency.

An Integrated Design Framework for Graph-Based Cyber-Attack Detection

The previous sections demonstrate that graph-based cyber-attack detection involves several interconnected design decisions. Detection performance depends not only on the selected classifier but also on how security data are transformed into graphs, how

structural information is represented, which topological properties are retained, and how computational efficiency and interpretability are evaluated. Based on these considerations, the main stages can be organized into an integrated design framework.

The proposed framework is not restricted to NVG-based representations. Instead, NVG serves as a concrete example of how graph construction, topological characterization, metric importance, and representation reduction can be connected within a broader graph-based cyber-attack detection methodology.

Define the security observation unit.

The first step is to determine what will be represented in the analysis, such as flows, packets, hosts, processes, system events, function calls, or sequential traffic-feature values. This decision establishes the basic unit from which the graph will be constructed.

Select the graph construction strategy.

Nodes and edges should be defined according to the security problem and available data. Communication, flow, provenance, function-call, temporal, or visibility graphs may capture different aspects of malicious behavior.

Choose the representation strategy.

The constructed graph can be represented through handcrafted topological descriptors or automatically learned representations. The selection should consider the objectives of the analysis, particularly predictive capability and interpretability.

Characterize structural information.

Relevant graph properties should be extracted or learned. Depending on the problem, these may include connectivity, clustering, centrality, path structure, community organization, or temporal characteristics.

Evaluate predictive and structural information.

Evaluation should not be limited to classification metrics. When interpretability is an objective, benign and malicious traffic can also be compared in terms of their structural graph characteristics.

Assess metric importance and redundancy.

Evaluate the contribution of topological descriptors using complementary importance or feature-selection methods, while considering redundancy, feature interactions, and stability across validation partitions.

Reduce the representation where appropriate.

Define progressively reduced metric subsets based on importance and redundancy evidence, and compare them with the full representation in terms of predictive capability and computational requirements.

Validate the resulting model.

Evaluate the selected representation using an appropriate validation strategy and, where feasible, independent holdout data or nested cross-validation to reduce selection bias and assess generalizability.

Evaluate computational requirements.

Graph construction, topological metric extraction, model training, inference time, memory requirements, and scalability should be considered when assessing practical applicability.

Preserve interpretability.

Where possible, detection results should be related back to the original security data and graph structure. This can help explain

not only whether malicious behavior is detected but also which structural characteristics contribute to the decision.

Overall, the framework connects the major stages of graph-based cyber-attack detection as follows:

Security Data → Graph Construction → Graph Representation → Structural Characterization → Classification and Structural Analysis → Importance Assessment → Representation Reduction → Validation → Efficiency and Interpretability

This integrated perspective emphasizes that graph-based cyber-attack detection should be approached as an end-to-end representation and analysis problem, rather than solely as a model-selection problem. The effectiveness of the final system depends on the interaction among graph construction, representation, structural information, learning, validation, and computational requirements.

Challenges and Open Research Directions

Despite the potential of graph-based representations for cyber-attack detection, several methodological and practical challenges remain. High classification performance on a benchmark dataset does not necessarily guarantee that the same graph representation, topological characteristics, or selected metrics will remain effective in different network environments. Future research should therefore consider not only predictive performance but also generalization, evaluation reliability, computational scalability, and interpretability.

Generalization Across Datasets and Environments

Topological patterns identified in a particular dataset may partly reflect its traffic-generation process, feature distributions, attack composition, or network environment. Therefore, attack-specific signatures and important graph metrics should be validated across independent datasets before being considered generalizable.

Future studies should investigate whether similar structural patterns remain stable across enterprise, IoT, cloud, and other operational network environments. Cross-dataset experiments would also help determine whether the importance rankings of topological metrics remain consistent under different traffic conditions.

Class Imbalance and Rare Attacks

Cyber-attack datasets often contain substantial differences in the number of observations available for different traffic classes (Chang & Branco, 2021). Consequently, overall accuracy alone may provide an incomplete description of model behavior, particularly for rare attack categories. Evaluation should therefore include class-sensitive measures such as macro-averaged F1-score, per-class precision and recall, and confusion matrices, together with more global measures such as weighted F1-score and Matthews correlation coefficient.

Computational Scalability

Graph construction and topological characterization can introduce substantial computational overhead, particularly when large numbers of traffic features, frames, or graph descriptors are processed (Kanca & Türker, 2025). Scalability should therefore be evaluated across the complete analysis pipeline rather than only at the classifier level. Future work should investigate more efficient graph-construction procedures, selective computation of topological descriptors, parallel processing, and representation-reduction strategies that balance structural information with computational requirements.

Explainability Beyond Feature Importance

Feature-importance methods can indicate which topological descriptors contribute to a model's predictions, but importance alone

does not fully explain why a particular traffic pattern is classified as malicious (Kanca & Türker, 2025). Future research should connect model-level explanations with graph-level structural changes and the underlying network-traffic characteristics from which those graphs are constructed. Such analysis may help distinguish between predictive importance and security-relevant structural interpretation.

Dynamic Graphs and GNN-Based Models

Static graph representations capture structural relationships within a defined observation interval but may not fully represent how those relationships evolve over time. Dynamic and temporal graph models provide a potential direction for incorporating structural evolution directly into cyber-attack detection (Kanca & Türker, 2025; J. Zhao et al., 2026; Zola et al., 2022). Future studies may also investigate advanced GNN-based architectures that capture evolving structural relationships while addressing the computational and interpretability challenges associated with automated graph representation learning.

Conclusion

Graph-based cyber-attack detection should be approached as an end-to-end representation and analysis problem rather than solely as a model-selection problem. The effectiveness of a graph-based detection system depends not only on the learning algorithm but also on how security data are mapped to nodes and edges, how the resulting structural information is represented, and which graph-derived characteristics are retained for subsequent analysis. Graph construction and representation should therefore be treated as central methodological design decisions that directly influence predictive capability, interpretability, and computational requirements.

Within this broader perspective, Natural Visibility Graphs provide a useful mechanism for transforming sequential network-traffic observations into graph structures by encoding relationships

determined by their temporal ordering and relative magnitudes. Evidence from attack-specific NVG analysis indicates that malicious behavior is not necessarily characterized by a single universal topological property; rather, different cyber-attack classes may exhibit distinct traffic-feature–topology signatures. This observation highlights the value of topological analysis not only as a source of predictive features but also as a means of structurally interpreting how malicious traffic differs from benign behavior.

At the same time, richer graph representations are not necessarily better representations. Extracting large collections of topological descriptors can introduce redundancy and computational overhead, making metric relevance and representation reduction important components of graph-based model design. Feature-importance and feature-selection methods can help identify potentially informative structural properties, but importance rankings should not be interpreted as direct evidence that a reduced representation is sufficient. Reduced metric sets should instead be validated explicitly in terms of predictive performance, stability, and computational cost. The central design objective is therefore not simply to maximize classification performance or minimize the number of descriptors, but to achieve an appropriate balance among predictive capability, structural interpretability, and computational efficiency.

Future research should extend this perspective through cross-dataset and cross-environment validation, improved treatment of class imbalance and rare attacks, and more systematic assessment of end-to-end computational scalability. Further research on dynamic graph models and graph neural networks may also improve the ability of automated representation-learning approaches to capture evolving structural patterns while maintaining computational scalability and interpretability. Progress in these directions could support graph-based cyber-attack detection systems that are not only

accurate, but also explainable, computationally practical, robustly evaluated, and transferable across heterogeneous network environments.

Kaynakça

Au, Q., Herbinger, J., Stachl, C., Bischl, B., Casalicchio, G., Atzmueller, M., Fürnkranz, J., Kliegr, T., & Schmid, U. (2022). Grouped feature importance and combined features effect plot. *Data Mining and Knowledge Discovery 2022 36:4*, *36*(4), 1401–1450. https://doi.org/10.1007/S10618-022-00840-5

Bonacich, P. (1972). Factoring and weighting approaches to status scores and clique identification. *Journal of Mathematical Sociology*, *2*(1), 113–120. https://doi.org/10.1080/0022250X.1972.9989806

Caville, E., Lo, W. W., Layeghy, S., & Portmann, M. (2022). Anomal-E: A self-supervised network intrusion detection system based on graph neural networks. *Knowledge-Based Systems*, *258*. https://doi.org/10.1016/j.knosys.2022.110030

Chang, L., & Branco, P. (2021). *Graph-based Solutions with Residuals for Intrusion Detection: the Modified E-GraphSAGE and E-ResGAT Algorithms*. http://arxiv.org/abs/2111.13597

Daya, A. A., Salahuddin, M. A., Limam, N., & Boutaba, R. (2019). A Graph-Based Machine Learning Approach for Bot Detection. *2019 IFIP/IEEE Symposium on Integrated Network and Service Management, IM 2019*, 144–152. https://arxiv.org/abs/1902.08538v1

Freeman, L. C. (1978). Centrality in social networks conceptual clarification. *Social Networks*, *1*(3), 215–239. https://doi.org/10.1016/0378-8733(78)90021-7

Guyon, I., Weston, J., Barnhill, S., & Vapnik, V. (2002). Gene selection for cancer classification using support vector machines. *Machine Learning*, *46*(1–3), 389–422. https://doi.org/10.1023/A:1012487302797/METRICS

Iacovacci, J., & Lacasa, L. (2016). Sequential visibility-graph motifs. *Physical Review E*, *93*(4). https://doi.org/10.1103/PHYSREVE.93.042309

Islam, R., Refat, R. U. D., Yerram, S. M., & Malik, H. (2020). Graph-Based Intrusion Detection System for Controller Area Networks. *IEEE Transactions on Intelligent Transportation Systems*, *23*(3), 1727–1736. https://doi.org/10.1109/TITS.2020.3025685

Kanca, A. M., & Türker, İ. (2025). A Systematic Review of Graph-Based Representation Techniques for Cyber-Attack Detection Across Application Domains. *Concurrency and Computation: Practice and Experience*, *37*(27–28). https://doi.org/10.1002/CPE.70389

Kanca, A. M., & Türker, İ. (2026). GraphDL: A visibility graph-based structural representation framework for flow-based cyber-attack detection. *Scientific Reports 2026*. https://doi.org/10.1038/S41598-026-55073-Z

Kursa, M. B., & Rudnicki, W. R. (2010). Feature Selection with the Boruta Package. *Journal of Statistical Software*, *36*(11), 1–13. https://doi.org/10.18637/JSS.V036.I11

Lacasa, L., Luque, B., Ballesteros, F., Luque, J., & Nuno, J. C. (2008). From time series to complex networks: the visibility graph. *Proceedings of the National Academy of Sciences of the United States of America*, *105*(13), 4972–4975. https://doi.org/10.1073/pnas.0709247105

Li, J., Zhang, R., Liu, J., & Liu, G. (2022). LogKernel: A Threat Hunting Approach Based on Behaviour Provenance Graph and Graph Kernel Clustering. *Security and Communication Networks*, *2022*(1), 4577141. https://doi.org/10.1155/2022/4577141

Lo, W. W., Layeghy, S., Sarhan, M., Gallagher, M., & Portmann, M. (2022). E-GraphSAGE: A Graph Neural Network based Intrusion Detection System for IoT. *Proceedings of the IEEE/IFIP Network Operations and Management Symposium 2022: Network and Service Management in the Era of Cloudification, Softwarization and Artificial Intelligence, NOMS 2022*. https://doi.org/10.1109/NOMS54207.2022.9789878

Lundberg, S. M., & Lee, S. I. (2017). A Unified Approach to Interpreting Model Predictions. *Advances in Neural Information Processing Systems*, *2017-December*, 4766–4775. https://arxiv.org/pdf/1705.07874

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

(PDF) Topological Signatures of Cyber-Attack Classes in Natural Visibility Graph Representations of Network Traffic. (n.d.). Retrieved October 1, 2026, from https://www.researchgate.net/publication/414686771_Topological_Signatures_of_Cyber-Attack_Classes_in_Natural_Visibility_Graph_Representations_of_Network_Traffic

Protogerou, A., Papadopoulos, S., Drosou, A., Tzovaras, D., & Refanidis, I. (2021). A graph neural network method for distributed anomaly detection in IoT. *Evolving Systems*, *12*(1), 19–36. https://doi.org/10.1007/s12530-020-09347-0

Watts, D. J., & Strogatz, S. H. (1998). Collective dynamics of 'small-world' networks. *Nature 1998 393:6684*, *393*(6684), 440–442. https://doi.org/10.1038/30918

Yao, Y., Su, L., & Lu, Z. (2018). DeepGFL: Deep Feature Learning via Graph for Attack Detection on Flow-Based Network Traffic. *Proceedings - IEEE Military Communications Conference MILCOM*, *2019-October*, 579–584. https://doi.org/10.1109/MILCOM.2018.8599821

Yao, Y., Su, L., Lu, Z., & Liu, B. (2019). STDeepGraph: Spatial-temporal deep learning on communication graphs for long-term network attack detection. *Proceedings - 2019 18th IEEE International Conference on Trust, Security and Privacy in Computing and Communications/13th IEEE International Conference on Big Data Science and Engineering, TrustCom/BigDataSE 2019*, 120–127. https://doi.org/10.1109/TrustCom/BigDataSE.2019.00025

Zhao, J., Zhang, D., He, Q., Lin, M., & Yang, Y. (2026). DMSTG-AD: an SDN intrusion detection method based on dynamic multi-scale spatio-temporal graph neural network. *Scientific Reports 2026 16:1*, *16*(1), 14528-. https://doi.org/10.1038/s41598-026-44360-4

Zhao, Z., Wang, J., & Wang, C. (2013). An unknown malware detection scheme based on the features of graph. *Security and Communication Networks*, *6*(2), 239–246. https://doi.org/10.1002/SEC.524;SUBPAGE:STRING:FULL

Zola, F., Segurola-Gil, L., Bruse, J. L., Galar, M., & Orduna-Urrutia, R. (2022). Network traffic analysis through node behaviour classification: a graph-based approach with temporal dissection and data-level preprocessing. *Computers and Security*, *115*. https://doi.org/10.1016/j.cose.2022.102632

CHAPTER 3

ADVERSARIAL ARTIFICIAL INTELLIGENCE AND CROSS SECTOR CYBERSECURITY GOVERNANCE

ÖMER SEVİNÇ[1]

Abstract Education, healthcare, and government use AI for different missions, yet their AI systems face common technical attacks and governance failures. This chapter develops a cross-sector model for adversarial machine learning and trustworthy cybersecurity. It distinguishes attacks during data collection, model development, deployment, and operation; examines poisoning, evasion, extraction, privacy leakage, prompt injection, and agent misuse; and maps controls to the AI lifecycle. A comparative analysis shows why control intensity must vary with consequences: an incorrect tutoring response, a corrupted clinical recommendation, and a distorted public-benefit decision do not produce equivalent harm. The proposed governance model combines risk classification, secure development, independent evaluation, continuous monitoring, incident disclosure, and meaningful human control.

1 Dr., Ondokuz Mayıs University, Vezirköprü Vocational School, Department of Computer Programming.

Keywords adversarial machine learning, AI governance, cybersecurity, model security, prompt injection, risk management

Introduction

Organizations increasingly use AI inside security operations and expose AI capabilities to students, clinicians, civil servants, and citizens. This creates a coupled problem: AI can defend digital systems, but AI components must also be defended. Conventional cybersecurity protects accounts, devices, applications, and networks. AI security adds the behavior of data-driven models, whose performance can change when inputs, training data, context, or dependencies change.

A cross-sector approach is useful because many technical attacks are shared, while acceptable risk is not. A chatbot that gives an incorrect study suggestion can usually be corrected. A clinical model that suppresses a warning may cause physical harm. A public system that incorrectly flags fraud may interrupt a legal entitlement. Governance must therefore combine reusable technical controls with sector-specific impact analysis (NIST, 2023; Vassilev et al., 2024).

Adversarial AI Across the Lifecycle

During data acquisition, an attacker may poison labels, insert backdoors, manipulate feedback, or compromise a data supplier. Provenance records, source authentication, statistical screening, and separation of duties reduce this risk. Data quality checks alone are insufficient because malicious samples may appear statistically ordinary while creating a targeted behavior.

During model development, risks include vulnerable packages, exposed credentials, unauthorized changes, and weak evaluation. Reproducible builds, signed artifacts, dependency inventories, isolated training environments, and controlled experiment records provide evidence about what was produced.

Red-team evaluation should test expected misuse and high-impact failure modes before release.

At deployment, attackers may craft evasion inputs, extract the model through repeated queries, infer sensitive training information, or exploit an application around the model. Rate limits, authentication, input validation, output filtering, privacy protections, and least-privilege tool access reduce exposure. Generative systems require special attention to prompt injection because instructions and untrusted content may share the same context.

During operation, model drift, supplier updates, and changing attacker behavior can invalidate earlier tests. Monitoring should cover both system security and model behavior. Organizations need thresholds for warning, restricting, rolling back, or disabling a model, and they must rehearse those actions before an incident.

Comparative Risk in Three Sectors

Education places particular weight on children's privacy, equal access, academic freedom, and due process. Monitoring designed to detect account compromise should not become an unrestricted behavioral profile. Automated restrictions need timely review because they may block learning or assessment.

Healthcare places greater weight on immediate safety and continuity. A containment action can itself create harm if it interrupts a therapeutic device or removes clinical information. Model security must be tested with clinical workflow, and humans need authority and time to evaluate consequential outputs.

Government carries duties of legality, equality, transparency, and continuity. An AI system may affect benefits, enforcement, or public communication. Security controls must preserve evidence for oversight and provide a route to challenge outcomes. Some technical

details may need protection, but secrecy cannot replace accountability.

Sector	Highest consequence	Required emphasis
Education	Loss of access or unfair discipline	Privacy, due process, and age-appropriate safeguards
Healthcare	Patient harm or unsafe delay	Clinical validation, continuity, and authorized containment
Government	Loss of rights or essential service	Legality, auditability, and public accountability

Integrated Governance Model

The first step is risk classification based on the decision supported, affected population, reversibility, data sensitivity, scale, and dependency of essential services. Classification determines testing depth, approval authority, monitoring, and incident reporting. A low-impact drafting assistant should not receive the same process as a diagnostic or eligibility system.

The second step is secure lifecycle management. Organizations should maintain an AI inventory, data lineage, dependency records, model and prompt versions, evaluation results, access policies, and named owners. Changes should move through controlled review. Independent evaluation is appropriate where developers have incentives or blind spots that could weaken assurance.

The third step is operational governance. Security and domain teams should share metrics, conduct scenario exercises, and define decision rights during incidents. Reports should separate model error, malicious manipulation, data failure, and infrastructure failure because each requires a different response. Lessons should update requirements, tests, and procurement terms.

Meaningful human control depends on competence and authority. A nominal reviewer who cannot understand uncertainty, access supporting evidence, or reverse an action does not provide

effective oversight. Training must therefore be role-specific and supported by interfaces that present evidence at the time of decision.

Taxonomy of Attacker Goals and Capabilities

Adversarial AI analysis should identify the attacker's objective, knowledge, access, and stage of intervention. Objectives may include causing any error, forcing a specific output, degrading service, extracting information, evading detection, or obtaining unauthorized tool execution. An attacker may know only that a model exists, may observe confidence scores, may query it repeatedly, or may have access to training data and parameters. These distinctions determine which tests and mitigations are relevant.

Capability also includes influence over surrounding systems. A user who cannot alter a model may still manipulate a document retrieved by it, compromise an API key, or exploit an overprivileged plug-in. Security assessment should therefore cover the complete AI system, including data pipelines, orchestration code, external tools, human review, monitoring, and update mechanisms. Model robustness alone cannot compensate for weak application security.

Testing Red Teaming and Independent Evaluation

Evaluation should begin with ordinary performance and extend to malicious conditions. Testers can introduce label errors, hidden triggers, distribution shifts, malformed inputs, prompt injections, excessive queries, and attempts to reveal restricted information. Results should be reported against predefined acceptance criteria, with separate measures for successful attacks, false alarms, performance degradation, and operational impact. A single benchmark score does not establish security across changing contexts.

Red teams should include domain specialists who understand realistic consequences. An education exercise may test exposure of examination material; a healthcare exercise may test corrupted clinical context; a government exercise may test unauthorized access to citizen records or public communication tools. Independent evaluation is valuable for high-impact systems because developers may know the intended behavior but overlook misuse paths or organizational incentives that weaken controls.

Security of Large Language Models and AI Agents

Large language models blur the boundary between data and instructions. Prompt injection exploits this property by placing adversarial instructions in user input, retrieved documents, web pages, or tool output. Current defenses can reduce but not eliminate the problem. Systems should treat external content as untrusted, constrain tools through explicit schemas and permissions, separate security decisions from free-form generation, and require confirmation for irreversible or high-impact actions.

AI agents magnify risk because they can plan sequences, call services, and act with delegated credentials. Each agent should have a defined purpose, limited environment, spending and action limits, complete audit trail, and a reliable stop mechanism. Credentials should be short-lived and scoped to the specific task. Simulated environments and staged deployment can reveal unsafe loops, unexpected tool combinations, and attempts to bypass approval before the agent reaches operational systems.

Assurance Evidence and Control Mapping

Organizations need an assurance case that connects risks to controls and evidence. The case may include data provenance, secure-development records, threat models, test results, access reviews, incident exercises, monitoring performance, and approval decisions. Evidence should be versioned because a model update,

new retrieval source, or infrastructure change may invalidate an earlier conclusion. Control mapping can align AI-specific safeguards with established cybersecurity, privacy, safety, and quality-management processes.

Residual risk should be stated explicitly. Adversarial training may improve resistance to tested inputs without covering unknown attacks. Output filters may reduce harmful disclosure while introducing false positives. Differential privacy may limit memorization while reducing utility. Decision-makers need these tradeoffs in concrete terms, along with conditions for restricting or ending use. Assurance is credible when it describes limitations and operational response, not when it presents a control as complete protection.

Incident Disclosure and Shared Learning

AI incidents can be difficult to classify because poor output may result from model error, malicious input, compromised data, software failure, or misuse. Incident processes should preserve the interaction, context, model and prompt versions, connected tools, identity events, and relevant system logs. Investigators should assess both immediate technical effects and downstream decisions made by people or automated services.

Responsible disclosure and sector information sharing can prevent repeated harm. Reports should describe the affected component, attack path, impact, containment, and durable corrective action without exposing personal information or providing unnecessary exploitation detail. Cross-sector learning is especially valuable for prompt injection, model supply-chain compromise, and privacy leakage because the same mechanisms recur in education, healthcare, and government even when the consequences differ.

Conclusion

Adversarial AI is not a single vulnerability that can be removed through one test. It is a lifecycle risk shaped by data, software, people, suppliers, and institutional purpose. Education, healthcare, and government can reuse core controls, but they should scale those controls to the harm that a failure or attack could create. Trustworthy deployment requires evidence that the system is secure enough for its specific use, can be monitored in practice, and can be stopped or recovered without abandoning the service it supports.

References

European Union Agency for Cybersecurity. (2021). Securing machine learning algorithms. https://www.enisa.europa.eu/publications/securing-machine-learning-algorithms

European Parliament and Council of the European Union. (2024). Regulation (EU) 2024/1689 laying down harmonised rules on artificial intelligence. Official Journal of the European Union.

National Institute of Standards and Technology. (2023). Artificial Intelligence Risk Management Framework 1.0 (NIST AI 100-1). https://doi.org/10.6028/NIST.AI.100-1

National Institute of Standards and Technology. (2024). Artificial Intelligence Risk Management Framework Generative Artificial Intelligence Profile (NIST AI 600-1). https://doi.org/10.6028/NIST.AI.600-1

Organisation for Economic Co-operation and Development. (2025). Governing with artificial intelligence. https://doi.org/10.1787/795de142-en

Vassilev, A., Oprea, A., Fordyce, A., & Anderson, H. (2024). Adversarial Machine Learning: A Taxonomy and Terminology of Attacks and Mitigations. NIST. https://doi.org/10.6028/NIST.AI.100-2e2023

World Health Organization. (2024). Ethics and governance of artificial intelligence for health: Guidance on large multi-modal models. https://www.who.int/publications/i/item/9789240084759

CHAPTER 4

AUTOMATED DETECTION OF PREDATORY JOURNALS: A SURVEY OF COMPUTATIONAL APPROACHES AND RELIABILITY CHALLENGES

Didem Yangeç[1]

Abstract

Predatory journals are a major publishing problem that threatens the reliability of scholarly communication. They have become more visible with the growth of open access publishing. Recent approaches to their automatic detection draw on machine learning, natural language processing, web and visual analysis, explainable artificial intelligence, and large language models. However, differences in datasets, ground truth labels, feature representations, and validation strategies make it difficult to compare reported performance directly. This structured narrative review compares studies on automatic predatory journal detection in terms of their datasets, labeling strategies, computational methods, validation approaches, and reliability concerns. The synthesis identifies several key methodological challenges. These include inconsistent ground truth labels, class imbalance, false positives and false negatives, changes over time, external validation, and

[1] Öğr. Gör. Dr., Muğla Sıtkı Koçman Üniversitesi, Bilgi İşlem Daire Başkanlığı, Orcid: 0009-0003-4882-9934

reproducibility. The generalizability of these systems across disciplines, geographical regions, and linguistic contexts also requires further assessment. The findings suggest that future research should look beyond classification performance. Greater attention should be given to datasets that combine multiple sources and include timestamps, evidence fusion, explainable models, and decision support with human oversight. Large language models are therefore recommended for extracting evidence, structuring information, and generating explanations, alongside independent validation and human oversight.

Keywords: Predatory journals, automatic detection, machine learning, explainable artificial intelligence, large language models

Introduction

Background and Motivation

The digitalization of scholarly communication and the growth of open access publishing have made scientific knowledge more accessible to researchers. These changes have also made the reliability of publishing practices an important concern. Predatory publishing is associated with misleading or opaque practices, inadequate editorial and peer review processes, and financial interests taking priority over scientific quality assurance. However, the literature offers no universally accepted operational definition of predatory publishing or agreement on the criteria that define it (Kakamad et al., 2024, 2026; Perera, 2025). Common indicators include opaque or inadequate peer review, inconsistencies in editorial board information, unverifiable indexing and impact metric claims, unclear or misleading article processing charges, aggressive publication invitations, and difficulties in verifying publisher information (Fung et al., 2026; Kakamad et al., 2024; Thallinger et al., 2026). Some of these features can be extracted automatically from journal websites, bibliographic sources, and publisher

communications. This has made predatory journal detection a subject of computational research using natural language processing, machine learning, and web mining.

Researchers' decisions to publish in predatory or questionable journals cannot be explained solely by a lack of individual awareness. In Türkiye, research has examined the relationship between performance evaluation systems based on metrics and questionable publishing choices (Öztürk & Taşkın, 2024). In Ghana, the role of academic promotion criteria in discouraging publication in questionable journals has also been assessed (Frandsen et al., 2024). A scoping review of health researchers in South Asia identified publication pressure, inadequate research support, and perceived immediate benefits as major factors (Kashyap et al., 2024). Awareness remains another concern. A rapid scoping review of medical students found limited awareness (Tomlinson, 2024), while a study involving 558 researchers in Jordan identified a need for education (Khabour et al., 2024). A survey in Brazil that received 3,067 responses also reported a significant association between publication pressure and experience with predatory publishing (do Carmo et al., 2026). The potential inclusion of studies published in predatory journals in evidence syntheses extends the problem beyond individual publication decisions (Martinino et al., 2025).

Labeling a journal as "predatory" is more than a technical classification task. Lists of predatory journals differ in their identification criteria, update frequency, transparency, and removal procedures (Kakamad et al., 2024, 2026). Ground truth labels are therefore a key methodological consideration in automatic detection studies. They directly affect how model results should be interpreted. Blacklists such as Beall's List are used to define the positive class. Sources such as DOAJ, Web of Science, Scopus, and PubMed may serve as comparison sources or help define the negative class in

some studies (Richtig et al., 2023). However, overlaps between blacklists and bibliographic indexes have been documented (Richtig et al., 2023). The “predatory” label may also unfairly discredit emerging journals (Perera, 2025). These concerns suggest that such sources should not be treated as fixed or universal ground truth. The reliability of automatic detection systems thus depends on both the algorithms used and the sources and criteria used to assign class labels.

Need for Automated Detection

Traditional journal assessment relies largely on manual review by researchers, librarians, or subject specialists. However, this approach is difficult to scale across large numbers of journals whose websites, editorial teams, indexing claims, and publishing policies change over time. The lack of standardized lists and criteria for identifying predatory journals can also make assessments sensitive to the criteria applied and the publishing context (Kakamad et al., 2024; Perera, 2025). Researchers, editors, libraries, academic institutions, funders, and publishers all continue to play a role in addressing predatory publishing (Laine et al., 2025). Automatic systems should therefore serve as scalable decision support tools that assist human assessment. They can screen large numbers of journals, identify risk indicators, and prioritize cases that require closer review.

One computational example is the Academic Journal Predatory Checking (AJPC) system developed by Chen et al. (2023). It combines web data collection, feature extraction, and classification using machine learning. More recently, Zhuang et al. (2025) assessed website content and design alongside publication metadata. Their study showed how automatic screening could draw on multiple data sources. Automatic detection is thus moving toward

computational screening that combines several sources of evidence and supports expert review.

Advancement of Computational Approaches

Computational methods for automatic predatory journal detection now include machine learning using metadata and text, natural language processing, website and visual analysis, explainable machine learning, large language models, and behavioral evidence analysis. Chen et al. (2023) evaluated textual features from journal websites using several classification algorithms. Zhuang et al. (2025) combined website content and design with bibliometric indicators to develop an approach drawing on multiple data sources. Another area of development concerns the explainability of model outputs and the use of large language models in journal assessment. Wu et al. (2024) used interpretable machine learning and SHAP feature contributions to identify indicators associated with model predictions. Al-Moghrabi et al. (2024) empirically examined ChatGPT's performance in assessing biomedical and dental journals. These studies show growing interest in the evidence behind predictions and the reliability of new artificial intelligence tools, alongside classification performance.

The evidence available for computational assessment extends beyond static web content. Patterns in publication invitations and indexing claims (Fung et al., 2026; Thallinger et al., 2026), unusual editorial board structures (Downes, 2024), and active tests of peer review, fees, communication practices, and observable publisher behavior (Burgiss et al., 2024) provide additional sources of evidence. The field is therefore moving beyond binary classification based on a single data type toward a broader computational framework that can assess evidence from multiple sources.

Scope and Contributions of This Review

Studies on automatic predatory journal detection vary considerably in their datasets, labeling strategies, feature representations, algorithms, and evaluation protocols. The positive class may be defined using blacklists, delisting status, or expert assessment (Chen et al., 2023; Kakamad et al., 2024; Zhuang et al., 2025). The negative class is also defined using different sources. These differences make reported accuracy, precision, recall, and F1 scores difficult to compare directly. Strong performance on internal tests does not guarantee the same reliability in practice. Independent evaluations of AJPC have shown that classification results can vary across journal groups and assessment times (Teixeira da Silva & Kendall, 2023; Teixeira da Silva & Scelles, 2024; Teixeira da Silva et al., 2025). Using different web addresses for the same journal can also produce different outputs (Teixeira da Silva, 2025). These findings show that automatic detection systems require assessment beyond algorithmic performance. Ground truth labels, data sources, external validation, stability over time, input sensitivity, and reproducibility must also be considered.

Some studies of automatic journal assessment focus on specific disciplines. For example, Al-Moghrabi et al. (2024) examined assessment using a large language model in biomedical and dental journals. Other approaches use broader journal collections (Chen et al., 2023; Zhuang et al., 2025). However, overall performance on a broad dataset does not establish comparable reliability across health sciences, engineering, social sciences, and other disciplines. An important research need is therefore to develop automatic assessment systems that can be used across disciplines and validate their performance within each discipline using independent datasets.

This review examines these methodological differences and the need for generalizability across disciplines within a common framework. Its main contributions are to compare datasets and strategies for establishing ground truth; classify machine learning, natural language processing, web and visual analysis, explainable artificial intelligence, large language models, and behavioral approaches within this framework; and assess reported performance in relation to differences in data and labels. It also identifies key challenges involving external validation, changes over time, reproducibility, and misclassification. Finally, it discusses explainable decision support approaches that draw on evidence and retain human oversight. The review addresses two main research questions:

RQ1: How do ground truth definitions, data sources, feature representations, machine learning methods, and validation strategies affect the reliability of automatic predatory journal detection?

RQ2: What findings and limitations does the existing empirical literature report regarding the use of large language models in automatic journal assessment?

Guided by these questions, the review evaluates methods in relation to both their reported performance and the data, labels, evidence, and validation conditions under which their results were obtained.

Methodology

Review Design and Scope

This study is a structured narrative review of the computational literature on automatic predatory journal detection. It examines datasets, labeling strategies, feature representations, methods, and validation approaches. The aim is to identify, classify, and critically synthesize the main computational approaches in this

methodologically diverse field. The primary scope covers studies that automatically or semi automatically identify predatory or questionable publishing at the journal or publication venue level. Approaches using machine learning, deep learning, natural language processing, explainable artificial intelligence, large language models, web and visual analysis, and multiple data sources were considered.

Predatory journals were treated as conceptually distinct from hijacked or cloned journals because they represent different phenomena. However, studies on the computational detection of hijacked or cloned journals were included as related literature because of their methodological similarities. Studies providing empirical evidence on ground truth labels, external validation, stability over time, input sensitivity, and behavioral or structural indicators were considered supporting literature. The evidence base therefore comprised three groups: (i) core studies that directly develop automatic detection or classification methods, (ii) related studies on the computational detection of hijacked or cloned journals, and (iii) supporting studies that provide evidence on ground truth, external validation, or detection features.

Information Sources and Search Strategy

The literature search was conducted using Scopus, Web of Science, Google Scholar, and IEEE Xplore. It focused on automatic detection of predatory and questionable journals, alongside methodologically related studies on hijacked and cloned journal detection. Topic terms were combined with computational method terms. Topic terms included “predatory journal”, “predatory publishing”, “predatory publishing venue”, “questionable journal”, “hijacked journal”, “clone journal”, and “journal hijacking”. Method terms included “automatic detection”, “classification”, “machine learning”, “deep learning”, “artificial intelligence”, “natural

language processing", "NLP", "large language model", "ChatGPT", "BERT", "explainable artificial intelligence", "web analysis", "domain analysis", and "network analysis". Boolean operators were used to combine these terms. Separate sets of search queries were developed for predatory or questionable journals and for hijacked or cloned journals.

An example of a general search string was: ("predatory journal" OR "questionable journal" OR "predatory publishing venue") AND ("automatic detection" OR "classification" OR "machine learning" OR "deep learning" OR "natural language processing" OR "artificial intelligence" OR "large language model" OR "ChatGPT" OR "BERT"). Searches for hijacked or cloned journals combined ("hijacked journal" OR "clone journal" OR "journal hijacking") with method terms such as "detection", "classification", "machine learning", "artificial intelligence", "large language model", "web", "domain", and "network analysis". The electronic search was supplemented by backward and forward citation tracking. Reference lists of key included studies and relevant publications citing them were examined. This helped identify studies that used different terminology but remained relevant to the research questions.

Inclusion and Exclusion Criteria

Studies were assessed according to their relevance to the research questions and the type of evidence they provided for automatic journal assessment. Eligibility criteria were applied consistently with the core, related, and supporting literature groups defined in the Review Design and Scope section. A study was included if it met at least one of the following conditions:

1. It presented a computational method for automatically or semi automatically identifying or assessing predatory or questionable journals or publication venues. Eligible methods

included machine learning, deep learning, natural language processing, large language models, explainable artificial intelligence, web or visual analysis, bibliometric analysis, or combinations of these approaches.

2. It developed a computational or systematic digital method for detecting hijacked or cloned journals.

3. It empirically assessed the external validity, stability over time, or input sensitivity of an existing automatic predatory journal detection system.

4. It provided empirical findings that could directly inform ground truth construction, feature extraction, or evidence validation in automatic detection systems.

Studies concerned only with researcher awareness, publication pressure, institutional policy, or the general causes and consequences of predatory publishing were excluded from the comparative methodological synthesis. Where needed, they were used only to explain the motivation for the study or its sociotechnical context. General studies on fabricated articles and paper mills were also excluded from the core review unless they directly addressed detection at the journal or publisher level.

Duplicate records were removed. Records lacking sufficient bibliographic information to assess their methods or findings were excluded from the main comparative synthesis. For studies available only as titles or abstracts, methodological details that could not be verified were omitted from the comparison tables.

Study Selection

Study selection proceeded in stages. Titles and abstracts were first assessed against the research questions and inclusion criteria. Records clearly outside the scope were excluded. Full texts or

detailed publication records of potentially eligible studies were then examined. The final selection followed the inclusion and exclusion criteria defined in the Inclusion and Exclusion Criteria section.

The included studies were organized into the three evidence groups defined in the Review Design and Scope section. Core computational studies directly examined predatory or questionable journal detection. Related studies addressed the computational detection of hijacked or cloned journals. Supporting studies provided evidence on ground truth, external validity, and behavioral, editorial, or bibliometric indicators. However, the review records did not systematically document the total numbers of retrieved records, duplicates, records screened by title and abstract, or publications assessed in full text. These counts were therefore not estimated retrospectively, and no study selection flow diagram was presented.

Data Extraction and Study Classification

For core computational studies, information used in the comparative synthesis was limited to data explicitly reported in the source publications. Where available, the extracted information covered the unit of analysis, dataset size and class distribution, label sources and meanings, data and feature types, algorithms or models, training and validation approaches, and reported performance metrics. Values not explicitly provided in the source were not estimated. Accuracy, precision, recall or sensitivity, F1, specificity, ROC AUC, and PR AUC were recorded only as reported in each study. Because the studies used different datasets and target class definitions, these values were not used to rank methods directly. Label meaning was also treated as a separate comparison dimension. Labels based on blacklist membership, removal from an index, expert assessment, or another protocol were not assumed to represent the same target variable.

For related and supporting studies, extraction focused on the sample or unit of analysis, evidence type, assessment approach, main findings, and methodological contribution to automatic detection. These studies were assessed separately to clarify methodological challenges and additional sources of evidence. They were not ranked alongside the performance results of core classifiers.

Synthesis Approach

A quantitative meta analysis was not conducted because the datasets, target class definitions, feature representations, and evaluation protocols could not be directly combined. A structured comparative and narrative synthesis was used instead. Reported performance values were therefore neither pooled as a common effect size nor used to rank algorithms across studies. The synthesis proceeded at two levels. First, studies were described by their data and method types. These categories covered metadata and text analysis using natural language processing, web and visual analysis, conventional or explainable machine learning, large language models, and behavioral or multimodal approaches. Second, shared methodological challenges were compared across studies. These included ground truth heterogeneity, class imbalance, the balance between false positives and false negatives, external validation, changes over time, input sensitivity, reproducibility, explainability, and generalizability across contexts.

Following this distinction, the Datasets and Ground Truth Labels and Automatic Detection Methods sections describe datasets, labeling approaches, and computational methods. Their shared implications for reliability are addressed in the subsequent comparative synthesis section. Findings on large language models were interpreted within the limits of the available empirical evidence. Results from direct assessments of predatory or

questionable journals were kept conceptually separate from those concerning hijacked or cloned journals.

Datasets And Ground Truth Labels

Interpreting model performance in automatic predatory journal detection requires attention to both the algorithm and the data and class definitions used for training. A machine learning model learns the labels assigned to its training data. The criteria used to define classes such as “predatory”, “questionable”, and “legitimate” therefore directly determine what its outputs mean. Dataset size and class distribution must be considered alongside label sources, assignment dates, evidence types, and intended meanings.

Blacklist-Based Labeling

Blacklists are commonly used to define the positive class in automatic predatory journal detection. Journals identified as predatory, potentially predatory, or suspicious by a particular list or assessment process are treated as positive examples. Beall’s List has served this purpose in several computational and empirical studies (Al-Moghrabi et al., 2024; Burgiss et al., 2024; Chen et al., 2023; Richtig et al., 2023; Siess, 2025). Bin Ateeq and Al-Khalifa (2023) constructed their dataset using the DOAJ directory, lists of journals rejected or removed by DOAJ, and Beall’s List. Of the 9,866 journals collected, 5,533 were classified as legitimate and 4,333 as questionable. Classification experiments used the 6,836 journals for which web content was available.

Blacklist membership should not be treated as fixed or undisputed ground truth. Richtig et al. (2023) reported that some journals associated with Beall’s List also appeared in bibliographic and indexing systems. These included 570 journals in Scopus, 224 in DOAJ, and 50 in Web of Science (Richtig et al., 2023). This finding shows that blacklist membership and inclusion in an index

or bibliographic system are not necessarily mutually exclusive. The positive class derived from a blacklist should therefore be understood as journals labeled predatory or suspicious under that list's criteria. It does not establish that every journal is definitively predatory. This distinction matters when comparing models trained on different datasets. Kakamad et al. (2024) also showed that lists of predatory journals and publishers vary in their inclusion and exclusion criteria, transparency, and updating procedures. Blacklists are useful label sources, but they should not be treated as flawless ground truth on their own.

Whitelists and Indexing Databases

Some studies define the negative or "legitimate" comparison class using whitelists and indexing sources. Chen et al. (2023) compiled their list of legitimate journals from the BIH QUEST resource, which draws on DOAJ and PubMed Central data. Al-Moghrabi et al. (2024) selected comparison journals from Web of Science and DOAJ. However, treating inclusion in a particular index or whitelist as a universal indicator of legitimacy is a methodological assumption. Richtig et al. (2023) showed that some journals associated with Beall's List also appeared in Scopus, DOAJ, PubMed, PubMed Central, and Web of Science. Blacklist membership and indexing therefore do not always define mutually exclusive classes (Richtig et al., 2023). Similarly, Chen et al. (2023) noted that the absence of a new journal from sources such as DOAJ or Web of Science should not automatically be interpreted as evidence of predatory publishing.

Datasets derived from whitelists or indexes should therefore clearly identify the source behind each label. Where appropriate, descriptions such as "journals listed in DOAJ" or "journals selected from Web of Science" provide greater clarity than the broad label

“legitimate journals”. Making the source explicit allows differences in labels across studies to be interpreted more accurately.

Hybrid and Expert-Based Labeling

Some studies combine labels derived from lists with independent expert assessments. This allows model results to be examined against a separate reference set. It also makes it possible to assess different evidence sources separately, without assuming that list membership and expert judgment represent the same labeling process. Zhuang et al. (2025) classified 12,869 journals that remained on the DOAJ whitelist as “legitimate”. They classified 2,536 journals removed for guideline violations or “questionable practices” as “questionable”. The study also used a separate expert assessment set of 360 journals from Unpaywall that DOAJ had not previously classified as whitelisted or delisted. These journals were assessed by a librarian specializing in open access publishing and two of the study’s authors (Zhuang et al., 2025).

This design makes the distinction between ground truth derived from lists and expert assessment explicit. Expert assessment is not flawless ground truth either. The criteria, assessment protocol, and agreement between assessors can all affect how results should be interpreted. Hybrid datasets and those incorporating expert assessment should therefore report both the final labels and the evidence and assessment procedures used to assign them.

De-listing-Based Labeling

Delisting is another source of labels in some studies. Zhuang et al. (2025) did not classify every journal removed from DOAJ as questionable. They first excluded journals removed for legitimate reasons, such as publication ending or voluntary withdrawal. The remaining journals were assigned to the “questionable” class. This approach accounts for changes in a journal’s assessment status,

whereas a static blacklist does not. However, delisting and predatory publishing are not synonymous. As illustrated by Zhuang et al. (2025), the source's "questionable" label should therefore be retained rather than generalized to "predatory".

This distinction shows that models trained under different labeling protocols may perform technically similar classification tasks while predicting different concepts. Model performance must therefore be interpreted alongside the source and intended meaning of the labels.

Behavioral Evidence-Based Data Sources

Labels derived from lists generally reflect third party assessments. Other studies directly examine observable journal or publisher behavior and offer complementary evidence for automatic detection. Burgiss et al. (2024) used active open source intelligence to examine 256 suspicious journals selected from Beall's and Kscien lists. They assessed peer review, fees, location, and communication patterns. They also submitted five articles generated with ChatGPT 4 to the target journals. They reported that 55 journals accepted or published these articles, or both (Burgiss et al., 2024). This design made it possible to observe actual publisher behavior alongside claims made on publisher websites.

Fung et al. (2026) examined 422 publication invitation emails sent to a single academic. They checked claims about impact factors, indexing, and ISSNs against independent sources. Thallinger et al. (2026) assessed 188 invitation emails in a prospective analysis over two months. They examined communication features such as the alignment between the invitation topic and the recipient's expertise. These indicators should not be treated as class labels on their own. Their value lies in adding distinct and independent evidence that can be considered alongside lists and other sources.

Class Imbalance and Changes Over Time

Class distribution affects the interpretation of model performance alongside the labeling process. Chen et al. (2023) used a dataset containing 833 predatory and 1,213 legitimate journals. Zhuang et al. (2025) used a substantially imbalanced dataset of 2,536 questionable and 12,869 legitimate journals. Reported performance must therefore be assessed in relation to each dataset's class distribution. Accuracy alone may be insufficient when the positive class is the minority. Zhuang et al. (2025) explicitly acknowledged class imbalance and used the area under the precision recall curve, or PR AUC, to evaluate their models. Performance metrics should therefore be interpreted with particular attention to the balance between identifying minority class cases and producing false positives.

The temporal characteristics of datasets also require attention. Journal websites, editorial information, indexing status, and list membership can change over time. Features and labels collected on one date cannot be assumed to represent a later period equally well. This review treats such changes as a methodological concern. Reporting the data collection date, the period for which labels are valid, and, where possible, the dataset version is important for assessing reproducibility and validity over time.

Why Is There No Universal Ground Truth?

Taken together, the datasets used in the literature reveal at least six labeling approaches based on different sources and protocols. They do not provide a single universal ground truth for automatic predatory journal detection:

1. Labeling based on blacklists: Journals appearing on Beall's List or similar sources are used as the positive class (Burgiss et al., 2024; Richtig et al., 2023).

2. Labeling based on whitelists or indexes: Journals selected from DOAJ, Web of Science, or similar sources are used as the comparison or negative class (Al-Moghrabi et al., 2024; Chen et al., 2023).

3. Expert assessment: Independent assessments by librarians or researchers are used to establish ground truth (Zhuang et al., 2025).

4. Labeling based on delisting: Removal from a list or index is used to define a journal's class (Zhuang et al., 2025).

5. Assessment based on behavioral evidence: Observable practices, such as article acceptance behavior, verifiable publishing claims, and publisher communications, are used as evidence (Burgiss et al., 2024; Fung et al., 2026; Thallinger et al., 2026).

6. Hybrid labeling: Multiple lists or indexing sources are used together. Bin Ateeq and Al-Khalifa (2023) combined the DOAJ directory, lists of journals rejected or removed by DOAJ, and Beall's List to construct their dataset.

These categories do not necessarily represent the same target variable. Labels identifying journals as "listed on Beall's List", "removed from DOAJ", "judged questionable by experts", or showing behavioral risk indicators can carry different meanings depending on the underlying protocol. Reviews of predatory publishing lists have documented differences in inclusion and exclusion criteria, transparency, updating procedures, and removal processes (Kakamad et al., 2024, 2026). This review therefore treats ground truth as the labeling protocol that defines the target class and the evidence supporting it. Comparisons across studies should jointly consider the data source, unit of analysis, class definitions and label meanings, labeling protocol, data collection date, class

distribution, and independent validation status. This makes the conditions behind reported performance clearer.

Automatic Detection Methods

Studies on automatic predatory journal detection differ considerably in the types of data they use and how they represent those data. Early studies mainly used textual features extracted from journal websites and conventional machine learning algorithms. More recent research incorporates website design, visual features, bibliometric indicators, editorial board information, behavioral signals, explainable artificial intelligence, and large language models (LLMs). This section examines these approaches according to their methods and evidence types.

Conventional Machine Learning Approaches

Conventional machine learning approaches classify journals using features extracted from their websites or journal information. The Academic Journal Predatory Checking system developed by Chen et al. (2023) is a key example. Chen et al. (2023) combined data collection, feature extraction, and model prediction in a single system. Their dataset contained 833 predatory and 1,213 legitimate journals. They allocated 80% of the data to training and 20% to testing. Website texts were preprocessed, and Bag of Words and TF IDF representations were assessed alongside diff scores. Eight classifiers were compared. Random Forest produced the best results, with a reported recall of 0.982 and F1 score of 0.980.

Bin Ateeq and Al-Khalifa (2023) compared conventional machine learning, deep learning, and transformer models within the same framework. They used SVM, KNN, a neural network, LSTM, CNN, BERT, and ALBERT. The CNN model achieved an F1 score of 0.96 for journal classification. For identifying reasons related to predatory publishing criteria, SVM achieved a micro F1 score of

0.67. However, the performance of these models depends on their datasets and labels. Internal test results alone do not establish external validity. Independent evaluations of AJPC have reported problematic classifications across different journal groups (Teixeira da Silva & Kendall, 2023; Teixeira da Silva & Scelles, 2024; Teixeira da Silva et al., 2025). These findings show that automatic detection systems also need testing beyond their training data.

Natural Language Processing and Text-Based Analysis

Journal websites contain textual information about scope, peer review, publication ethics, fees, and indexing claims. Natural language processing can use this content for automatic detection. Chen et al. (2023) classified preprocessed web texts using Bag of Words and TF IDF representations. Bin Ateeq and Al-Khalifa (2023) assessed textual content using conventional methods such as SVM and KNN, alongside LSTM, CNN, BERT, and ALBERT. Text analysis has therefore been explored through methods ranging from word representations to contextual language models.

The presence of a claim in a text does not establish its accuracy. An indexing or impact factor claim remains a website statement unless verified against an independent source. Fung et al. (2026) compared claims about indexing, impact factors, ISSNs, and PubMed in publication invitations with verifiable sources. Their study shows the importance of assessing whether textual claims can be verified, alongside identifying their presence. These findings suggest that checking extracted claims against external sources is a useful research direction for NLP systems. This could extend text analysis beyond word patterns to include the verifiability of claims.

Website and Visual Analysis Approaches

Website structure and visual features can also inform automatic journal assessment. Zhuang et al. (2025) combined

website content, design features, and publication metadata to predict questionable open access journals. They used information on aims and scope, editorial boards, open access, copyright, and ethics policies, alongside homepage screenshots and website source code (Zhuang et al., 2025). This brought textual content and online presentation into a common assessment framework.

In the related literature, Andoohgin Shahri et al. (2018) focused on hijacked journals, which are conceptually distinct from predatory journals. They examined 104 websites representing 59 genuine and 45 hijacked journals and used a decision tree classification approach. Features related to domain names and websites were assessed. The resulting decision tree was also tested on a separate sample of five genuine and five hijacked journals. This study provides an early example of using website features as computational evidence for a related form of journal fraud. Abalkina (2021) examined hijacked journals through repeated content in the archives of cloned websites. Article titles, author names, and affiliations were searched using the Google Custom Search API. The method identified 62 cloned websites associated with 57 genuine journals. The study also reported that two cloned websites were anticipated before they became active. Although it did not directly classify predatory journals, it showed how archive similarities and relationships between websites could support hijacked journal detection.

Siess (2025) approached visual features differently by examining published article PDFs rather than journal websites. The dataset contained 443 legitimate and 555 potentially predatory publications. Features included typography, PDF metadata, page layout, visual complexity, and white space. The study treated these features as complementary indicators for large scale screening rather than definitive classification criteria. Website design and visual quality should therefore be interpreted alongside other evidence,

without assuming that they directly reflect publishing quality. Input sensitivity also matters in website assessment. Teixeira da Silva (2025) reported that entering URLs for different pages of the same journal into AJPC could produce different classifications. This finding highlights the need to report the URLs and page content used in automatic website assessment.

Explainable Machine Learning Approaches

Automatic detection requires attention to both the predicted class and the features behind the prediction. Explainable artificial intelligence (XAI) methods help examine how individual features contribute to model outputs. Wu et al. (2024) developed an interpretable machine learning approach to assess early warning risk for predatory open access journals. They used several machine learning algorithms and interpreted model outputs with SHAP. This allowed them to examine the influence of indicators contributing to predictions. Such explanations can support decision making with human oversight. Users can inspect the indicators influencing a risk assessment and use this information to review unexpected results.

Explainability, however, does not establish accuracy. SHAP can show how a feature contributes to a model prediction. It does not prove that the feature alone is a reliable indicator of predatory publishing. Explanations must therefore be assessed alongside label quality and the validation approach.

Large Language Model-Based Approaches

The ability of large language models to interpret information and generate explanations has introduced a new approach to journal assessment. Al-Moghrabi et al. (2024) examined ChatGPT's performance in classifying biomedical and dental journals as predatory or legitimate. The study assessed 400 journals, comprising 200 predatory and 200 legitimate journals. The predatory group was

selected from Beall's List and unsolicited publication invitations. The legitimate group was selected from Web of Science and DOAJ (Al-Moghrabi et al., 2024). Journal names were queried using the free version of ChatGPT available in May 2023. Additional prompts and journal website information were provided when needed. Responses were coded under ten criteria.

ChatGPT correctly classified 92.5% of journals labeled predatory and 71% of those labeled legitimate. Reported overall accuracy was 0.82, sensitivity was 0.93, and specificity was 0.71. The false positive rate was 29%, while the false negative rate was 7.5% (Al-Moghrabi et al., 2024). Performance therefore differed between the two classes. The findings are also limited to the disciplines, reference sources, and ChatGPT version examined.

Hegedűs et al. (2026) provided complementary evidence on LLM systems through a study of hijacked medical journals. This study did not directly examine predatory journal classification. It assessed 21 previously identified and 10 newly identified hijacked journals alongside their genuine counterparts. ChatGPT, Gemini, Copilot, DeepSeek, Qwen, Perplexity, and Claude were compared. The systems could provide general information but could not reliably distinguish genuine journals from hijacked versions. Copilot performed better but still made errors (Hegedűs et al., 2026). Together, these two studies indicate that LLM assessments require careful testing before they can serve as final verification tools. A use that allows closer oversight would involve extracting evidence and supporting human assessment. This recommendation reflects the methodological synthesis developed in this review from the available findings.

Behavioral and Multimodal Approaches

Static website features reflect the information a journal presents about itself. Behavioral evidence adds information about

how a publisher operates in practice. Relevant sources include publication invitations, editorial board patterns, article acceptance behavior, and practices after publication. Burgiss et al. (2024) actively investigated 256 suspicious journals selected from Beall's and Kscien lists. The researchers submitted five articles generated with ChatGPT 4 to these journals. The articles were accepted or published by 55 journals. Peer review processes, fees, communication patterns, and relationships between publishers were also assessed (Burgiss et al., 2024). The study shows how observed publisher behavior can provide evidence alongside website statements.

Editorial board and publication invitation data offer further complementary evidence. Downes (2024) identified 96 individuals who appeared at least 20 times on the editorial boards of predatory publishers. The study identified different circumstances, including identity theft and voluntary membership. Frequent editorial board membership should therefore not be interpreted as proof of predatory publishing on its own. Fung et al. (2026) checked claims about indexing, impact factors, and ISSNs in publication invitations against independent sources. Thallinger et al. (2026) examined 188 publication invitation emails and reported that 148 did not match the recipient's expertise.

Practices after publication provide another layer of evidence. Xu et al. (2026) examined retraction practices among 1,511 potentially predatory journals on the updated Beall's List. Of these, 46 had retracted a total of 645 publications by the end of 2022. The study assessed the visibility of retraction policies, document accessibility, and the information provided in retraction notices (Xu et al., 2026). Such behavioral evidence can complement other data sources. However, tests involving active article submission can raise ethical and operational concerns. Behavioral indicators should

therefore be assessed alongside other evidence rather than treated as ground truth on their own.

Overall Comparison

Core computational studies differ in their datasets, label definitions, features, methods, and evaluation strategies. Their reported performance values are therefore not directly comparable. Table 1 summarizes studies that directly apply computational methods to predatory or questionable journal detection, alongside label meanings and validation approaches.

Table 1. Comparison of core computational studies on predatory or questionable journal detection

Study	Dataset / N	Labels / ground truth	Data / features	Method	Evaluation	Main finding / limitation
Bin Ateeq and Al-Khalifa (2023)	9,866 journals collected; web content from 6,836 journals used in classification experiments	DOAJ, journals rejected or removed by DOAJ, and Beall's List	Journal website texts; different text representations	SVM, KNN, NN, LSTM, CNN, BERT, ALBERT	Best classification: CNN, F1 = 0.96; violation identification or explanation task: SVM, micro F1 = 0.67	Web and NLP approach; dependent on label and web content sources
Chen et al. (2023)	2,046 journals: 833 predatory, 1,213 legitimate	Beall's List and Stop Predatory Journals; legitimate: DOAJ and PMC	Web texts; BOW, TF IDF, diff scores	Eight classifiers; RF selected	80% training, 20% testing; recall = 0.982, F1 = 0.980	Strong internal performance; dependent on lists and web content
Wu et al. (2024)	434 medical and 194 engineering observations; class ratio 1:1	CAS Early Warning List; negative class: open access or hybrid journals selected using SJR	17 bibliometric or journal indicators	LR, SVM, DT, RF, XGBoost, and SHAP	70%/30% split; cross validation with five folds; XGBoost: accuracy = 0.979, F1 = 0.979, AUC = 0.990	Strong internal performance; labels represent early warning or risk status
Al-Moghrabi et al. (2024)	400 journals: 200 predatory, 200 legitimate	Beall's List and unsolicited emails; legitimate: WoS/SCIE and DOAJ	ChatGPT responses; ten indicator categories	ChatGPT; RF/GB for criteria analysis	Accuracy = 0.818; sensitivity = 0.925; specificity = 0.710	FP = 29%, FN = 7.5%; assessed at one time point and limited to specific disciplines
Zhuang et al. (2025)	15,405 journals: 12,869 legitimate, 2,536 questionable	Journals remaining in DOAJ or removed for questionable practices	Web content, design, bibliometrics	RF; CNN for visual representations	Cross validation with five folds; PRC AUC = 0.79 ± 0.03	Combined feature model achieved the highest PRC AUC; expert review recommended

Source: Author’s synthesis of the studies summarized in this table.

Table 1 shows why methodological differences must be considered alongside differences in data and labels. Bin Ateeq and Al-Khalifa (2023) combined several reference sources, while Chen et al. (2023) used labels derived from lists and website texts. Al-Moghrabi et al. (2024) assessed journals using ChatGPT. Zhuang et al. (2025) used a questionable journal label based on DOAJ delisting history. Wu et al. (2024) also examined feature contributions associated with model predictions. These studies used different target variables and validation designs. Their performance values should therefore not serve as a common ranking measure.

The methods also differ in their evidence sources and representations. Table 2 summarizes the main method families considered in this review by data source, typical processing approach, strengths, and limitations.

Table 2. Comparison of main method families in automatic predatory journal detection

Approach	**Main data source**	**Typical features / processing**	**Strength**	**Main limitation**
Metadata analysis	ISSNs, publishers, indexing information, bibliometric records	Structured journal and publisher features	Scalable and relatively easy to verify	Depends on database coverage and currency
Text analysis / NLP	Journal web pages, policy texts, invitation emails	BoW, TF IDF, textual patterns, claim extraction	Enables automatic processing of large amounts of unstructured content	The presence of a claim does not establish its accuracy
Web / visual analysis	Web pages, screenshots, PDF documents	Website design, visual representations, PDF and typography features	Adds signals beyond text	Design quality does not directly establish publishing quality
Conventional ML / deep learning	Structured and extracted features, web texts	SVM, RF, XGBoost, CNN, LSTM	Enables scalable classification of large datasets	Depends on label and training data quality
Explainable artificial intelligence	Model inputs and prediction outputs	SHAP and feature contribution analysis	Allows examination of features associated with predictions	Feature contributions do not establish causality
Large language models	Web texts and journal information expressed in natural language	Classification, information extraction, explanation, summarization	Can process complex and unstructured information in natural language	Sensitivity to model version, prompts, and context; risk of generating inaccurate information
Behavioral / multimodal approaches	Publication invitations, editorial boards, practices after publication, and external sources	Behavioral patterns, claim verification, evidence integration	Can provide complementary evidence beyond journal statements	Data collection is costly, and equivalent evidence may not be available for every journal

Source: Author’s synthesis of the studies reviewed in the Automatic Detection Methods section.

The method families in Table 2 focus on different evidence types. Conventional machine learning and NLP support scalable classification using web texts (Chen et al., 2023). Web and visual analysis incorporate structural and visual features (Siess, 2025; Zhuang et al., 2025). Explainable machine learning makes feature contributions associated with predictions visible (Wu et al., 2024). Assessments using LLMs provide empirical evidence on interpreting journal information through natural language (Al-Moghrabi et al., 2024). Behavioral studies add evidence from publication invitations, editorial board patterns, acceptance behavior, and practices after publication (Burgiss et al., 2024; Downes, 2024; Fung et al., 2026; Thallinger et al., 2026; Xu et al., 2026).

These differences make direct performance rankings inappropriate. Chen et al. (2023), Al-Moghrabi et al. (2024), and Zhuang et al. (2025) used different datasets, class definitions, and evaluation protocols. A higher accuracy or F1 score alone therefore does not establish that one method is generally superior. The comparison in this section considers four dimensions: data and evidence types, the source and meaning of ground truth labels, validation strategies, and explainability and reproducibility. the Comparative Analysis And Open Challenges section examines the methodological implications of these dimensions in greater detail.

Comparative Analysis And Open Challenges

The studies examined in previous sections show that automatic predatory journal detection can draw on diverse data sources and computational methods. However, reliability in practical use depends on more than algorithm selection. Ground truth heterogeneity, class imbalance, error types, changes over time, external validation, explainability, and generalizability must be considered together. Across the literature, the central challenge is to establish reliable and comparable evaluation frameworks.

Label Heterogeneity and Ground Truth Problem

One major obstacle to direct comparison is that identical class names can refer to different labeling protocols. Chen et al. (2023) constructed the positive class from blacklist sources. Zhuang et al. (2025) classified journals removed from DOAJ as "questionable" after excluding certain reasons for removal. Predatory publishing lists also differ in their criteria and updating procedures (Kakamad et al., 2024, 2026). Moreover, blacklist membership and indexing are not always mutually exclusive (Richtig et al., 2023). These differences affect the meaning of the target variable. "Blacklisted", "removed from an index", and "judged questionable by an expert" do not represent equivalent conceptual classes. Zhuang et al. (2025) also used an expert assessment sample separate from their main dataset. Accuracy, F1, and similar metrics from different studies should therefore not be interpreted as results obtained for the same target variable.

The lists examined by Kakamad et al. (2024, 2026) also vary in their inclusion criteria, updating practices, and removal procedures. Ground truth should therefore be described through more than the source name. The labeling protocol, assessment date, supporting evidence, and intended meaning of the label should also be reported. Algorithmic performance cannot be interpreted independently of ground truth quality. Strong performance may show that a model has successfully learned the assigned labels. It does not establish that those labels fully represent predatory publishing behavior in practice.

Class Imbalance and Performance Metrics

Class distributions differ across studies. Chen et al. (2023) used 833 predatory and 1,213 legitimate journals. Zhuang et al. (2025) used a more imbalanced dataset containing 2,536 questionable and 12,869 legitimate journals. Such differences affect

how performance metrics should be interpreted. Overall accuracy can mask errors in the minority class when performance on the majority class is strong. Precision, recall, F1, and PR AUC should therefore be considered together. Zhuang et al. (2025) accounted for class imbalance by comparing models using PR AUC.

The importance of each metric depends on the intended use. Recall becomes particularly important when the aim is to identify as many risky journals as possible. Precision becomes more important when the aim is to limit the proportion of legitimate journals among those flagged as risky. Model performance should therefore be interpreted alongside class distribution and error types, rather than through a single value.

Consequences of False Positives and False Negatives

False positives and false negatives have different consequences. A false positive can cause a journal considered reliable to be unnecessarily flagged as suspicious. A false negative can make a questionable journal appear to pose little risk. Evaluation should therefore consider error types alongside overall accuracy. In their assessment of ChatGPT, Al-Moghrabi et al. (2024) reported a false positive rate of 29% and a false negative rate of 7.5%. The system correctly classified 92.5% of journals labeled predatory and 71% of those labeled legitimate. These findings show that the same system can have different error profiles across classes. However, the meaning of these rates depends on the ground truth used. When reference labels are uncertain or disputed, observed false positive and false negative rates should not be treated as direct estimates of errors in practice.

An external evaluation of AJPC illustrates another aspect of this problem. Teixeira da Silva and Kendall (2023) assessed the top 2,500 SJR entries on 25 March 2023. The system labeled 65.64% as “suspected predatory” and 33.28% as “normal”, while 1.08% could

not be classified. This distribution does not itself represent a misclassification rate. It does, however, show the need to independently test results produced outside the training setting. False positives and false negatives should therefore be interpreted in relation to the reference labels and intended use of the system, alongside their numerical values.

Temporal Drift and Input Variability

Time is an important factor in automatic detection systems that use web data. Journal websites, editorial boards, indexing status, and publishing policies can change. Blacklists and whitelists may also be updated. A system should therefore not be assumed to behave identically at different times. Teixeira da Silva and Kendall (2023) used AJPC to assess the top 2,500 journals in the SJR ranking on 25 March and 30 March 2023. In the first assessment, 65.64% were classified as "suspected predatory". Five days later, this proportion had risen to 89.20%. The proportion classified as "normal" fell from 33.28% to 10.16% (Teixeira da Silva & Kendall, 2023). These differences show that system outputs can change over time.

Input variability presents a related problem. Teixeira da Silva (2025) showed that AJPC outputs could change when different URLs for the same journal were entered. These included homepages, editorial board pages, and author instructions. Among the ten journals examined, 50% showed inconsistent classifications depending on the URL used (Teixeira da Silva, 2025). This finding highlights the need to specify which page was assessed. Studies using web data should therefore clearly report the assessment date, input URL, and system version. Archiving web content where possible could also support later examination of the results. This recommendation is the review's methodological interpretation of the temporal and input sensitivity reported in (Teixeira da Silva & Kendall, 2023; Teixeira da Silva, 2025).

Reproducibility and External Validation

Performance on a held out test set does not establish that a model will perform equally well across other journal groups. Reproducibility and external validation therefore require separate consideration when assessing the reliability of automatic journal evaluation systems. AJPC, developed by Chen et al. (2023), was subsequently evaluated independently across different journal groups (Teixeira da Silva & Kendall, 2023; Teixeira da Silva & Scelles, 2024; Teixeira da Silva et al., 2025). Teixeira da Silva and Scelles (2024) reported that the system classified 47 of the 124 journals in the SJR Sports Science category, or 37.9%, as "suspected predatory". Teixeira da Silva et al. (2025) reported that 44 of the 50 journals in the FT50 group, or 88%, received the same classification.

These studies did not establish an independent gold standard. Their reported proportions therefore cannot be interpreted directly as false positive rates. They nevertheless show the need to examine AJPC outputs for journal groups outside its training data (Teixeira da Silva & Kendall, 2023; Teixeira da Silva & Scelles, 2024; Teixeira da Silva et al., 2025). Taken together, the findings suggest that validation based solely on random training and test splits is insufficient. Future studies could clarify reproducibility and generalizability by explicitly reporting evaluations conducted at different times and on independent journal groups. This recommendation reflects the methodological synthesis developed in this review.

Explainability and Human Oversight

Understanding which features are associated with a prediction is important for interpreting automatic detection outputs. Wu et al. (2024) used SHAP to examine feature contributions in a model assessing early warning risk. However, a strong contribution to a model prediction does not establish that a feature is a reliable

indicator of predatory publishing in practice. An explanation primarily reveals the patterns used by the model. Explainability should therefore be assessed separately from label quality and data source reliability.

This distinction supports the use of automatic systems as decision support tools. Showing the evidence contributing to a result can help users assess it more carefully than a class label or risk score alone would allow. Expert review of the final decision can also prevent automatic outputs from being treated as definitive judgments, particularly in uncertain cases. This proposed use with human oversight reflects the review's methodological synthesis of the available findings.

Fairness, Geographical Generalizability, and Language Bias

You et al. (2025) examined the relationship between questionable publishing, countries, and regional publishing environments. They found that publishing patterns can vary geographically. Their study also indicated that regional journals and questionable journals should not be treated as the same phenomenon. Perera (2025) discussed fairness concerns surrounding the "predatory" label, particularly when applied to new, less familiar, or poorly resourced journals. Together, these studies support the view that geography or publishing environment alone should not be treated as an indicator of quality (Perera, 2025; You et al., 2025). However, the available sources do not provide comparative experiments that directly measure language or geographical bias in automatic detection models. They therefore do not establish that models systematically learn to treat languages other than English, website design, or economic resources as risk indicators.

An important open challenge identified in this review is to validate automatic systems across languages, regions, and publishing environments. Subgroup evaluations could help

determine whether models learn publishing behavior or indirect characteristics such as geography and language. However, the limited evidence does not support strong generalizations about fairness or language bias.

Reliability and Reproducibility of Large Language Models

Al-Moghrabi et al. (2024) reported an overall accuracy of 0.82 for ChatGPT's assessment of biomedical and dental journals. Correct classification reached 92.5% for journals labeled predatory and 71% for those labeled legitimate. This difference highlights the need to verify LLM outputs before using them as final decisions. The findings are limited to the ChatGPT version, assessment period, disciplines, and sample examined (Al-Moghrabi et al., 2024). Additional prompts and journal website information were also used when needed. Reporting the model version, prompts, supplied context, and assessment date is therefore important for reproducibility. This recommendation is the review's methodological interpretation of the study design.

Explanations generated by LLMs are not equivalent to verified evidence. Hegedűs et al. (2026) compared ChatGPT, Gemini, Copilot, DeepSeek, Qwen, Perplexity, and Claude using hijacked medical journals. The systems could provide general information but could not reliably distinguish genuine journals from hijacked versions (Hegedűs et al., 2026). Although the study did not directly examine predatory journal classification, it provides complementary evidence for checking LLM outputs against independent sources. The roles of LLMs in extracting evidence, structuring information, and generating explanations should therefore be distinguished from their role as final classifiers. This distinction allows their use within decision support processes that can be checked and supervised.

Transition from Static Features to Behavioral Evidence

Static features such as website text, design, and metadata provide accessible information about a journal. They do not, on their own, establish how a publisher operates in practice. Behavioral evidence can therefore serve as a separate source of information that complements static features. Fung et al. (2026) checked claims about indexing, impact factors, ISSNs, and PubMed in publication invitations against independent sources. Thallinger et al. (2026) examined invitation content and found that the invitation topic did not match the recipient's expertise in 148 of 188 emails. These studies show how publisher statements can be compared with verifiable information and observed communication behavior.

Practices after publication can provide additional evidence. Xu et al. (2026) examined retraction practices among 1,511 potentially predatory journals on the updated Beall's List. They identified 645 retraction records across 46 journals by the end of 2022. The study assessed the visibility of retraction policies, document accessibility, and the quality of retraction notices (Xu et al., 2026). These findings suggest that automatic assessment can consider the verifiability of claims and observable publishing practices alongside textual and design features. However, equivalent behavioral evidence may not be available for every journal. This review therefore proposes using behavioral indicators as a complementary evidence layer, rather than as ground truth on their own.

Overall Synthesis

The literature shows that the main challenges in automatic predatory journal detection are interconnected. Differences in label sources and criteria affect the meaning of performance values. Class imbalance can alter error distributions. Assessment dates and web inputs can affect output stability. A high performance score alone

therefore does not establish system reliability. Internal validation is also insufficient on its own. Systems require external validation on independent data, separate assessment of false positives and false negatives, and clear identification of the evidence supporting predictions. Explainability can reveal which features influence a model, but it cannot resolve weaknesses in labels or evidence quality. Limited comparative evidence across languages and geographical publishing environments also calls for caution when judging generalizability.

These findings show that the research problem extends beyond identifying the classifier with the highest accuracy. The central need is to combine evidence from different sources in ways that remain verifiable, explainable, and reproducible as that evidence changes over time. Automatic detection should therefore be treated as a decision support problem that incorporates evidence quality, uncertainty, and human review.

Future Research Directions

Future research needs to address more than improvements in classification performance. Lists used to identify predatory publishing differ in their coverage and criteria (Kakamad et al., 2026). Zhuang et al. (2025) accounted for class imbalance and used a separate expert assessment set. External evaluations of AJPC show that results obtained outside the training setting require further testing (Teixeira da Silva & Kendall, 2023). Different page URLs for the same journal can also produce different results, highlighting the importance of clearly defining inputs in systems using web data (Teixeira da Silva, 2025). Drawing on these findings, this review considers future research in terms of data standardization, evidence from multiple sources, external validation, explainability, and decision support with human oversight.

Standardized, Timestamped, and Updatable Datasets

The literature uses different lists and data sources to establish ground truth. Kakamad et al. (2024, 2026) showed that lists of predatory journals and publishers differ in coverage and criteria. Richtig et al. (2023) found that some journals on Beall's List also appeared in Scopus, DOAJ, PubMed, PubMed Central, and Web of Science. Zhuang et al. (2025) supplemented listing and delisting information in their main dataset with a separate expert assessment set. These examples challenge the assumption that labels from different sources have equivalent meanings. Future datasets may therefore need to record more than final labels such as "predatory", "questionable", or "legitimate". Recording the label source, assessment date, criteria, and any label changes would provide useful context. This recommendation reflects the review's methodological synthesis of the observed label heterogeneity.

Standardized reporting of journal identifiers, ISSNs, collection dates, assessed URLs, and indexing status could also improve comparability. Recording URLs is particularly important because different web pages can produce different results (Teixeira da Silva, 2025). These records would clarify the temporal and input conditions represented by a dataset. They would also support temporal validation. Teixeira da Silva and Kendall (2023) showed that repeated AJPC assessments produced different classification distributions five days apart. Teixeira da Silva (2025) reported that different page URLs for the same journal could lead to different classifications. Data records should therefore include collection dates, input URLs, and system versions where available. This review also recommends testing models on data collected during a later period, alongside random training and test splits.

Multi-Source Evidence Fusion and Integration of Behavioral Evidence

Existing studies show that journal assessment can use several types of evidence. Zhuang et al. (2025) combined website content, design features, and publication information. Siess (2025) examined PDF features of published articles, while Downes (2024) investigated editorial board patterns. Fung et al. (2026) and Thallinger et al. (2026) examined publication invitations. Xu et al. (2026) assessed retraction practices as a form of quality control after publication. Although these studies use different data, they show how web content, visual features, editorial information, invitations, and practices after publication can provide distinct evidence layers (Downes, 2024; Fung et al., 2026; Siess, 2025; Thallinger et al., 2026; Xu et al., 2026; Zhuang et al., 2025). Publisher statements should be distinguished from information that can be independently verified. Fung et al. (2026) demonstrated this distinction by checking indexing and impact factor claims against external sources.

Future systems should therefore consider the sources and verifiability of evidence when combining it. Independent datasets should be used to test whether behavioral indicators add information beyond static features. Studies should also assess whether combining these evidence types reduces false positives and false negatives. These recommendations arise from this review's methodological synthesis. They do not describe a common evidence fusion model already tested by the reviewed studies.

Explainable and Human-Supervised Decision Support Systems

Wu et al. (2024) used SHAP to examine feature contributions to model predictions. This approach can reveal which features influence a model. A strong contribution to a prediction, however, does not establish that a feature is reliable evidence in practice. Tang (2025) proposed supplementing quantitative machine assessment

with qualitative community assessment. This provides a direct conceptual basis for using automatic systems to support human judgment.

Future systems should distinguish model explanations from the verifiability of the underlying evidence. Outputs could include the source, currency, and verification status of the evidence used. Intermediate outcomes such as “review required” or “insufficient evidence” could also be considered for uncertain cases. These proposed output formats and the recommendation for human oversight reflect the review’s methodological synthesis.

Reliable and Reproducible Use of Large Language Models

Available findings call for caution when using large language models as sole decision makers in journal assessment. Al-Moghrabi et al. (2024) reported different classification performance for journals labeled predatory and legitimate. Hegedűs et al. (2026) compared several chatbots using hijacked medical journals. The systems could not reliably distinguish genuine journals from hijacked versions. Although this second study did not directly examine predatory journal classification, it provides complementary evidence against treating LLM outputs as conclusive evidence without independent verification.

Future studies should examine the reproducibility of LLM assessments. They should clearly report the model and version, prompts, supplied context, and assessment date. Evaluating the same journals across models, prompts, and time points could also help assess output stability. These recommendations form part of this review’s methodological synthesis. They do not represent a common protocol directly tested in the existing studies.

Within this framework, LLMs could primarily extract evidence, structure information, and provide explanations. Sources

should accompany extracted information, and critical claims should be checked against independent data. This would position LLMs as components of decision support with human oversight.

External Validation, Generalizability, and Fair Assessment

Future automatic detection systems require evaluation beyond the datasets used to develop them. Independent validation should cover different data sources, disciplines, languages, and publishing environments. This would help establish whether learned patterns are specific to a particular dataset. Existing literature indicates that publishing practices can vary across regions and contexts (Perera, 2025; You et al., 2025). However, these studies do not directly establish systematic bias across all linguistic or geographical groups in current detection systems. Language, geography, and website design should therefore not automatically be treated as indicators of poor quality.

Where samples are sufficient, future evaluations should report performance separately across languages, disciplines, regions, and publisher groups. They should also examine whether false positive and false negative rates differ across these groups. The purpose is to assess how well model performance holds across contexts. This review proposes this evaluation framework based on the contextual findings in (Perera, 2025; You et al., 2025).

Conceptual Architecture for Next-Generation Automated Journal Evaluation

Taken together, the reviewed studies support developing automatic journal assessment within a decision support framework that draws on multiple sources. Relevant inputs could include web content, journal metadata, visual features, editorial information, bibliometric data, and behavioral evidence. LLMs could assist with extracting and explaining unstructured information. Combining

evidence would be only one part of this system. Each item's source, date, and verification status should also be recorded. Outputs could present a risk assessment alongside uncertainty and supporting evidence. Cases with insufficient or conflicting evidence could be referred for human review.

This architecture has not been implemented and validated in full by a single study in the reviewed literature. Chen et al. (2023), Zhuang et al. (2025), Wu et al. (2024), Al-Moghrabi et al. (2024), Fung et al. (2026), Thallinger et al. (2026), Downes (2024), Siess (2025), and Xu et al. (2026) provide evidence on different data and assessment components. The integrated framework proposed here is a conceptual synthesis of their findings.

Future research should therefore pursue more than higher accuracy. Priorities include traceable label sources, assessments that can be updated over time, independent verification of evidence, testing across contexts, and human review of uncertain cases. Together, these priorities could advance automatic predatory journal detection as a decision support process grounded in verifiable evidence.

Conclusion

Automatic predatory journal detection has expanded from conventional machine learning and natural language processing to web and visual analysis, explainable artificial intelligence, large language models, and behavioral evidence (Al-Moghrabi et al., 2024; Chen et al., 2023; Wu et al., 2024; Zhuang et al., 2025). These developments offer opportunities to assess large numbers of journals at scale. However, the reviewed studies show that system reliability depends on more than an algorithm's classification performance.

For RQ1, the literature indicates that reliability is directly affected by dataset construction, the sources and meanings of ground

truth labels, feature representations, and validation strategies. Blacklists, whitelists, indexing databases, delisting records, expert assessments, and behavioral evidence produce different class definitions (Kakamad et al., 2024, 2026; Richtig et al., 2023; Zhuang et al., 2025). Labels such as "predatory", "questionable", and "legitimate" therefore cannot be assumed to represent the same target variable across studies. Class imbalance, the different consequences of false positives and false negatives, changing web content, URL sensitivity, and limited external validation also affect performance interpretation (Teixeira da Silva & Kendall, 2023; Teixeira da Silva & Scelles, 2024; Teixeira da Silva et al., 2025; Teixeira da Silva, 2025; Zhuang et al., 2025). Accuracy, F1, and AUC should therefore be interpreted alongside data and label construction, ground truth quality, collection conditions, temporal validity, and independent validation.

For RQ2, the empirical evidence is more limited. The study directly examining ChatGPT for predatory journal assessment showed that LLMs could generate classifications and explanations. However, limitations remained, including false positives, explanations that had not been independently verified, and results tied to a particular sample and assessment conditions (Al-Moghrabi et al., 2024). A related study comparing AI chatbots using hijacked medical journals provided complementary evidence of difficulties in distinguishing genuine journals from hijacked versions (Hegedűs et al., 2026). These findings should not be treated as directly equivalent because hijacked journal detection and predatory journal detection address different target problems. Overall, the reliability of LLMs as independent, final journal classifiers has not yet been sufficiently established. Their use requires further investigation through broader datasets, independent validation, repeated assessments over time, and reproducible evaluation protocols.

The literature also suggests that reducing journal assessment to a binary classification can discard important information. Website text and structure, bibliometric information, editorial board patterns, publication invitations, indexing claims, and practices after publication can provide distinct and complementary evidence (Burgiss et al., 2024; Downes, 2024; Fung et al., 2026; Thallinger et al., 2026; Xu et al., 2026). Explainable artificial intelligence can help reveal features associated with model predictions (Wu et al., 2024). However, a feature's importance to a model does not establish its causal relevance or reliability as an indicator of predatory publishing. Similarly, explanations generated by LLMs are not equivalent to independently verified evidence.

One limitation of this review is that the numbers of screened, duplicate, excluded, and fully assessed records were not recorded quantitatively during study selection. These counts were not reconstructed retrospectively, and no selection flow diagram was presented. This limits assessment of the review's comprehensiveness and the reproducibility of its selection process.

More reliable automatic detection requires clear tracking of evidence sources, currency, and verifiability. Datasets and labeling protocols should be reported transparently, and internal test results should be supported by independent and temporal validation. The proposal to complement quantitative machine assessment with qualitative expert and community assessment (Tang, 2025) also supports this approach. The current evidence therefore favors using automatic detection as a process that systematically gathers evidence and supports human judgment through decision support tools whose outputs can be checked and reviewed.

References

Abalkina, A. (2021). Detecting a network of hijacked journals by its archive. *Scientometrics*, *126*(8), 7123–7148. https://doi.org/10.1007/s11192-021-04056-0

Al-Moghrabi, D., Abu Arqub, S., Maroulakos, M. P., Pandis, N., & Fleming, P. S. (2024). Can ChatGPT identify predatory biomedical and dental journals? A cross-sectional content analysis. *Journal of Dentistry*, *142*, Article 104840. https://doi.org/10.1016/j.jdent.2024.104840

Andoohgin Shahri, M., Jazi, M. D., Borchardt, G., & Dadkhah, M. (2018). Detecting hijacked journals by using classification algorithms. *Science and Engineering Ethics*, *24*(2), 655–668. https://doi.org/10.1007/s11948-017-9914-2

Bin Ateeq, W. M., & Al-Khalifa, H. S. (2023). Intelligent framework for detecting predatory publishing venues. *IEEE Access*, *11*, 20582–20618. https://doi.org/10.1109/ACCESS.2023.3250256

Burgiss, L., Tatum, B., Henshaw, C., Boswell, M., & Michaels, A. (2024). Predatory publication of AI-generated research papers. In *Proceedings of the First International Conference on Natural Language Processing and Artificial Intelligence for Cyber Security* (pp. 1–9). International Conference on Natural Language Processing and Artificial Intelligence for Cyber Security. https://aclanthology.org/2024.nlpaics-1.1/

Chen, L.X., Su, S.W., Liao, C.H., Wong, K.S., & Yuan, S.M. (2023). An open automation system for predatory journal detection. *Scientific Reports*, *13*(1), Article 2976. https://doi.org/10.1038/s41598-023-30176-z

do Carmo, F. A., Rezende, S. O., Paxiúba, C. M. C., & Lobato, F. M. F. (2026). Predatory publishing in the academic landscape: researchers' awareness and vulnerabilities. *Scientometrics*, *131*(5), 3029–3054. https://doi.org/10.1007/s11192-026-05614-0

Downes, M. (2024). The stock characters in the editorial boards of journals run by predatory publishers. *Learned Publishing*, *37*(3), Article e1608. https://doi.org/10.1002/leap.1608

Frandsen, T. F., Lamptey, R. B., & Borteye, E. M. (2024). Promotion standards to discourage publishing in questionable journals: a follow-up study. *The Journal of Academic Librarianship*, *50*(5), Article 102895. https://doi.org/10.1016/j.acalib.2024.102895

Fung, C., Razi, B., Basto, M., Bariol, S., Hossack, T., Baskaranathan, S., Ende, D., & Woo, H. H. (2026). Trust, but Verify: An Exploratory Audit of Predatory Journal and Publisher Solicitations. *ANZ Journal of Surgery*, *96*(4), 786–791. https://doi.org/10.1111/ans.70568

Hegedűs, M., Dadkhah, M., & Dávid, L. D. (2026). Benchmarking AI chatbots: assessing their accuracy in identifying hijacked medical journals. *Diagnosis*, *13*(1), 82–91. https://doi.org/10.1515/dx-2025-0043

Kakamad, F. H., Abdalla, B. A., Abdullah, H. O., Omar, S. S., Mohammed, S. H., Ahmed, S. M., Mohammed, K. K., Hassan, H. A., Baba, H. O., Ahmed, J. O., Mustafa, M. Q., Omar, D. A., Salih, R. Q., Rahim, H. M., Hussein, D. A., Hassan, M. N., Mikael, T. M., Hassan, H. A., & Najar, K. A. (2024). Lists of predatory journals and publishers: a review for future refinement. *European Science Editing*, *50*, Article e118119. https://doi.org/10.3897/ese.2024.e118119

Kakamad, F. H., Mustafa, A. M., Abdalla, B. A., Mohammed, S. H., Ahmed, S. M., Abdullah, H. O., Ahmed, J. O., Abdullah, F., Abdalla, S. S., Mikael, T. M., Hassan, H. A., Najar, K. A., & Omar, D. A. (2026). Predatory publishing lists: a systematic review. *Annals of Medicine & Surgery*, *88*(2), 1842–1848. https://doi.org/10.1097/MS9.0000000000004733

Kashyap, K., Islam, A. A., & Gielen, J. (2024). Why do healthcare researchers in South Asia publish in predatory journals? A scoping review. *Developing World Bioethics*, *24*(2), 54–65. https://doi.org/10.1111/dewb.12388

Khabour, O. F., Alzoubi, K. H., & Aldarabseh, W. M. (2024). Awareness of Jordanian Researchers About Predatory Journals: A Need for Training. *Science and Engineering Ethics*, *30*(6), Article 58. https://doi.org/10.1007/s11948-024-00519-8

Laine, C., Babski, D., Bachelet, V. C., Bärnighausen, T. W., Baethge, C., Bibbins-Domingo, K., Frizelle, F., Gollogy, L., Kleinert, S., Loder, E., Monteiro, J., Rubin, E. J., Sahni, P., Wee, C. C., Yoo, J.H., & Zakhama, L. (2025). Predatory journals: what can we do to protect their prey?. *BMJ*, *388*, Article q2850. https://doi.org/10.1136/bmj.q2850

Martinino, A., Smeenk, F. W.J.M., Basile, V., Soto, A., & Pouwels, S. (2025). Understanding the Influence of Predatory Journals Articles Included within Systematic Reviews: A Scoping Review. *The Serials Librarian*, *86*(3-4), 213–222. https://doi.org/10.1080/0361526X.2025.2523389

Öztürk, O., & Taşkın, Z. (2024). How metric-based performance evaluation systems fuel the growth of questionable publications?. *Scientometrics*, *129*(5), 2729–2748. https://doi.org/10.1007/s11192-024-04991-8

Perera, H. S. (2025). Questioning the Predator of the Predatory Journals: How Fair Are Global Publishing Standards?. *Learned Publishing*, *38*(2), Article e1662. https://doi.org/10.1002/leap.1662

Richtig, G., Berger, M., Koeller, M., Richtig, M., Richtig, E., Scheffel, J., Maurer, M., & Siebenhaar, F. (2023). Predatory journals: Perception, impact and use of Beall's list by the scientific community–A bibliometric big data study. *PLOS ONE*, *18*(7), Article e0287547. https://doi.org/10.1371/journal.pone.0287547

Siess, A. (2025). Evaluating the visual design of science publications—a quantitative approach comparing legitimate and predatory journal papers. *Scientometrics*, *130*(9), 5043–5066. https://doi.org/10.1007/s11192-025-05411-1

Tang, B. L. (2025). Are there accurate and legitimate ways to machine-quantify predatoriness, or an urgent need for an automated online tool?. *Accountability in Research*, *32*(2), 182–187. https://doi.org/10.1080/08989621.2023.2253425

Teixeira da Silva, J. A., & Kendall, G. (2023). (Mis-)Classification of 17,721 Journals by an Artificial Intelligence Predatory Journal Detector. *Publishing Research Quarterly*, *39*(3), 263–279. https://doi.org/10.1007/s12109-023-09956-y

Teixeira da Silva, J. A., & Scelles, N. (2024). An artificial intelligence tool misclassifies sport science journals as predatory. *Journal of Science and Medicine in Sport*, *27*(4), 266–269. https://doi.org/10.1016/j.jsams.2023.12.006

Teixeira da Silva, J. A., Tsigaris, P., & Moussa, S. (2025). Are most FT50 journals truly "suspected predatory", or has an AI-driven "predatory" journal detector erred in its classification?. *Journal of Electronic Resources Librarianship*, *37*(2), 135–146. https://doi.org/10.1080/1941126X.2025.2497727

Teixeira da Silva, J. A. (2025). Variation in journal website characterization by an AI-driven "predatory" journal detector. *Expert Systems with Applications*, *283*, Article 127933. https://doi.org/10.1016/j.eswa.2025.127933

Thallinger, C., Aschbacher, V., & Zielinski, C. C. (2026). Invitations to Publish in a Journal by Email: A Prospective Analysis. *Learned Publishing*, *39*(3), Article e2076. https://doi.org/10.1002/leap.2076

Tomlinson, O. W. (2024). Predatory publishing in medical education: a rapid scoping review. *BMC Medical Education*, *24*(1), Article 33. https://doi.org/10.1186/s12909-024-05024-x

Wu, J., Liu, T., Mu, K., & Zhou, L. (2024). Identification and causal analysis of predatory open access journals based on interpretable machine learning. *Scientometrics*, *129*(4), 2131–2158. https://doi.org/10.1007/s11192-024-04969-6

Xu, S. B., Liu, T., Nejadghanbar, H., & Hu, G. (2026). Retraction handling by potential predatory journals. *Accountability in Research*, *33*(1), 1–27. https://doi.org/10.1080/08989621.2024.2434245

You, T., Park, J., Lee, J. Y., & Yun, J. (2025). Regional profile of questionable publishing. *Journal of Informetrics*, *19*(3), Article 101670. https://doi.org/10.1016/j.joi.2025.101670

Zhuang, H., Liang, L., & Acuna, D. E. (2025). Estimating the predictability of questionable open-access journals. *Science Advances*, *11*(35), Article eadt2792. https://doi.org/10.1126/sciadv.adt2792


BİDGE