Multi-View Graph Model with Representation Alignment and Adaptive Fusion for Better Spam Detection*

Iryna Zamrii[1], Ivan Shakhmatov[1]

[1] State University of Information and Communication Technologies, Kyiv, Ukraine

Abstract

Web forms remain a vulnerable channel for coordinated web spam: automated submitters combine bulk submissions, repeated field refills, synchronized time bursts, and technical obfuscation—while labeled data are scarce and privacy rules are strict. We present a multi-view contrastive graph neural network to filter web spam. The approach integrates three event views: (1) a submission graph that links entries by field duplicates, temporal proximity, and shared contacts; (2) a behavioral heterogeneous graph of user–form–page interactions with technical attributes; and (3) a semantic k-NN graph built from text and URL embeddings. Each view uses a specialized GNN encoder (R-GCN for typed relations; GAT/GCN for submission and content graphs). We then apply node-wise fusion that dynamically balances behavioral and content signals. Training combines binary cross-entropy on labeled submissions with contrastive alignment across views (using drop-edge, feature masking, and stochastic k-NN augmentations) and light regularization for stability. Evaluation uses leak-free temporal splits. The model is interpretable via fusion weights and attention, respects privacy by anonymizing sensitive identifiers, and supports scalable graph updates. By calibrating thresholds to operational metrics, the model is ready for practical deployment in anti-spam systems.

Keywords

artificial intelligence, cybersecurity, web spam, web applications, spam detection, neural networks, vulnerability scanners, information security

1. Introduction

Web spam has evolved from simple text templates to coordinated campaigns with bulk form submissions, reuse of field structures, synchronized time bursts, distinctive URL parameters, and typical device fingerprints. These scenarios reduce the effectiveness of purely content-based filters, which are sensitive to distribution drift, and complicate the use of large language models that require large amounts of reliably labeled data and do not fully capture behavioral and network signals. Methods that focus on a single type of relation lose information about the consistency between textual, behavioral, and structural features. Graph modeling is natural for web interactions: users, pages, forms, devices, and individual submissions create a heterogeneous network. Useful signals are spread across several views of this network: at the submission level we observe content duplication and temporal clusters; at the behavioral level—links to devices, IP prefixes, and repeated activity paths; at the semantic level—similarity of texts and links. A strong solution must integrate these channels into aligned representations that are robust to obfuscation, drift, and partial labeling.

This paper presents a model for training a multi-view graph neural network with contrastive pretraining to filter web spam. Each view is encoded by a specialized GNN encoder: R-GCN for heterogeneous, typed interactions, and GAT/GCN for semantic links and submission-level graphs.

* SMICS'25: Workshop on Cryptology and Data Security, October 16-18, 2025, Lviv, Ukraine

1,* Corresponding author.

† These authors contributed equally.

i.zamrii@duikt.edu.ua (Iryna Zamrii); i.shahmatov@duikt.edu.ua (Ivan Shakhmatov)

0000-0001-5681-1871 (Iryna Zamrii); 0009-0004-9628-0365 (Ivan Shakhmatov)

Within each view, we aggregate the local context of neighbors, then perform adaptive fusion of the resulting vectors with node-dependent weights. This lets the model automatically favor the most informative view for a given submission. Contrastive pretraining aligns the representations across views and reduces the need for large labeled datasets; the final classifier outputs a spam probability with a threshold that can be tuned to operational metrics. We describe how to build the multi-view graph under privacy constraints, detail the model architecture and training objectives, and provide a high-level implementation plan in Python with PyTorch Geometric. The evaluation protocol uses temporal splits to avoid leakage, and includes comparisons with representative baselines plus ablation studies that show the contribution of each component and the robustness to drift and obfuscation.

The paper lays out a practical scheme (Fig. 1) for multi-view modeling of web interactions—submission, behavioral, and semantic levels—with feature sets built under privacy protection. On this basis, we design an architecture that combines local aggregation within each view with cross-view alignment and adaptive mixing of signals, able to operate when some channels are missing and in the presence of noise. The training process unites contrastive pre-alignment across views with supervision on labeled examples, which reduces reliance on large annotated datasets, improves generalization to new campaigns, and remains stable under distribution drift. Reproducibility is supported by an open, high-level Python processing pipeline with fixed configurations, temporal splits to prevent leakage, comparisons to representative baselines, and systematic ablation studies. The practical focus addresses latency and throughput requirements, threshold calibration to target operational metrics, control of false positives, and ethical considerations— including anonymization of personal data and additional review of high-confidence decisions before automatic blocking.

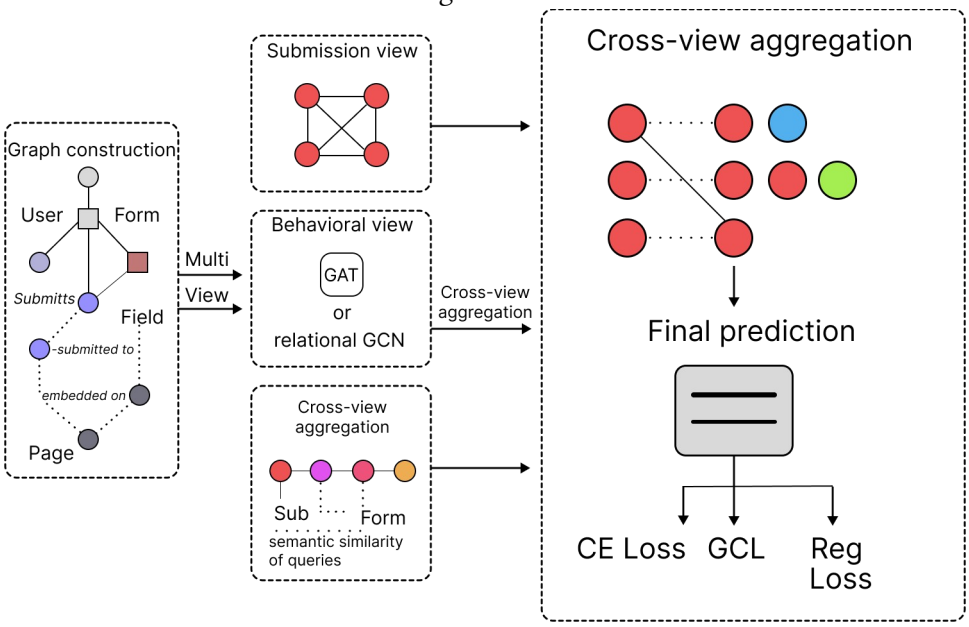

Figure 1: Multi-view graph model for web-spam filtering (with contrastive learning)

2. Literature Review and Problem Statement

The task is to protect web forms from spam by building a model that can score each submission in real time or near real time and decide to block, defer for moderation, or allow it. The model must align heterogeneous sources of signals—text/URL, behavioral and technical attributes, and structural links between events—without relying on any single type of indicator. We treat form submissions as the primary classification objects. For each submission, the following data are available: field contents (text, links), technical attributes (browser/platform, IP prefix, device), page/form context (URL, navigation path), temporal features (instant and aggregated), and relations to other submissions, users/devices, and pages.

For each submission, produce a spam-probability score and a controllable decision (threshold-based or cost-sensitive). The decision must be robust to changing spam tactics, text/URL obfuscation, and variation in technical attributes. We model the data as a multi-view structure: a submission level (links via field duplicates, temporal bursts, and shared technical traits), a behavioral level (users/devices/IP/pages and their relations), and a semantic level (similarity of texts and URLs). This design lets us combine content, technical, and structural signals. The model must learn from limited and partly noisy labels, so we combine supervised learning on labeled submissions with self-supervised representation alignment (contrastive pretraining). Automatic feature processing and aggregation are required: mapping texts and URLs to vector embeddings, deriving behavioral aggregates, and building links among objects without hand-crafted rule engineering.

The solution must meet latency and throughput requirements in production, support incremental context updates, operate with sliding time windows, and prevent information leakage across temporal slices. We need controlled error asymmetry (thresholds tuned to a target false-positive rate) and decision transparency at the level of channel contributions. User identifiers are anonymized (hashing, aggregating IPs to prefixes), only minimal attributes are stored, high-confidence decisions receive additional review before automatic blocking, and an appeal path is provided. Validation uses temporal splits, comparisons with representative content-based and graph baselines, and metrics suitable for imbalanced data (PR-AUC, F1 at a validation-selected threshold, and recall at fixed FP rates), plus ablations showing the contribution of channels and training techniques. The task is formulated as building a multi-view model for classifying web-form submissions that can automatically extract and integrate content, behavioral, and structural data while respecting production constraints, privacy, and reproducibility.

Research on spam detection is shifting from "flat" text features toward graph representations and multi-view architectures. Studies show that semantically induced graphs can capture long-range text dependencies without manual rule engineering and improve classification accuracy [1]; graph attention helps weight local links and noisy relations among interaction objects, which is crucial when some edges are misleading or adversarial [2]; and combining deep contextual embeddings (e.g., BERT) with multiple graph types—co-occurrence, syntax, and heterogeneous relations—covers both global and local patterns and remains robust on imbalanced datasets [3]. Together, these results suggest constructive principles for a web-spam model: work not with a single "flat" source, but in a unified graph space where content, behavioral, and structural signals reinforce each other. At the same time, web-form filtering has its own specifics that classic email, social-network, or SMS cases only partly address. Decisions are made at the submission level and must fit strict latency budgets; user/device identifiers are often anonymized and unstable; labels are scarce and imperfect; tactics change quickly, and text and URLs are regularly obfuscated. Hence the requirements: inductive capability (handle new nodes/edges without retraining), multi-channel alignment (integrate several views), robustness to drift, and the ability to use self-supervised pretraining to reduce dependence on labeled data.

The idea of an immutable ledger that records every SQL query [4] suggests a separate secure-telemetry layer for our model: store a "decision digest" asynchronously and with anonymization (hashes of fields/URLs, technical attributes, key links, and per-channel attributions). This adds transparency, traceability, and a basis for "block-before-DB" policies. To reduce the effect of intentionally mixed-in or noisy neighbors, we add neighbor-quality control to the graph encoders, following [5]: label/pseudo-label–aware top-k neighbor selection by cosine similarity; relation-aware aggregation (separate weights for edge types S–IP–S, S–UA–S, S–URL–S, S–FieldSim–S, S–Page–S, S–TimeBurst–S); and temporal decay of weights. For self-supervised alignment of representations, we use generatively augmented contrastive learning [6]: a generator proposes plausible edges in local ego-graphs, a discriminator filters out unnatural structures, and the encoder learns invariance to allowed graph changes. This improves robustness to obfuscation and drift when labels are scarce. Multi-view fusion via meta-paths [7] is implemented as parallel branches (GCN/GAT) on homogeneous views with weighted adjacency (real edges plus feature similarity)

and attention gating for adaptive channel mixing. We handle class imbalance with balanced/focal loss and with thresholds calibrated to a target FPR. Together, these ideas form a production-ready design: inductive, multi-channel, with controlled inference latency, channel-level explainability, and a built-in audit loop—delivering reliable filtering of web-form submissions in real or near-real time.

To strengthen the model against coordinated spam and mixed links, we add: a heterogeneous information network (HIN) branch with meta-paths plus embedding/clustering to detect collusive groups and a group-level "spaminess index" [8]; a fast ego-graph matching module based on dual contrastive matching for online search of similar patterns without quadratic cost [9]; a GMPT pretraining mode with intra- and inter-graph matching to improve generalization when labels are scarce [10]; spectral, multi-frequency, relation-aware convolutions with skip connections to handle heterophilous graphs (where nodes link to dissimilar nodes), including masking and class imbalance [11]; and a lightweight scoring head ("GNN embedding → SVM/Logistic Regression") to meet strict latency budgets (SLAs) at inference time [12]. Taken together with the built-in neighbor-quality control, cross-view contrastive alignment, and secure telemetry, these additions raise Recall at fixed FPR on rare attacks, reduce computation cost, and keep decisions explainable at the level of channels and link groups.

The training scheme for spam filtering addresses the "inconsistency problem" in graphs through consistency-guided neighbor selection and relation-aware aggregation, which down-weights deceptive links and balances edge-type weights [13]. Base semantic and behavioral views are enriched with protocol-agnostic features (lengths, URL indicators, compressibility, language) and merged in a shared feature space for cross-channel robustness [14]. Text/URL are encoded with contextual embeddings plus a light scoring head (logistic regression/linear model) to meet latency SLAs without losing accuracy [15]. Generalization in streaming settings and under label scarcity is strengthened by generatively constructed hard negatives and augmentations with feature fusion, which stabilize distributions and improve contrastive learning [16]; we also apply class balancing and threshold calibration to reach the target Recall@FPR.

Further directions confirm the effectiveness of training with minimal labels, cross-domain generalization, and explainability: self-supervised pretraining with feature ensembling across modalities (HTTP requests, email texts) reduces reliance on manual labeling and increases sensitivity to new tactics [17]; generative-adversarial loops with text extractors and domain-invariant representations enable detection of Sybil groups and fake reviews without loss when categories and styles shift [18]; cross-protocol signal fusion (web/email/DNS) in semi-supervised GNNs scales reputation analysis of network entities under label scarcity [19]; survey results document key production challenges (drift, multilingual settings, latency, subjective definitions) and support combining behavioral and semantic features with adaptive models for email and IoT [20]; matrix-factorization AMALS approaches together with TF-IDF and gradient procedures help with sparse/incomplete data in large systems while maintaining high accuracy [21]; and integrating XAI into web platforms shows that large datasets, classical and vector features, and transparent explanations can be joined into practical systems with F1 around 0.99 and strong generalization [22].

3. Model

The model is defined as a set of three aligned graph views—submission, behavioral/user, and semantic — processed by a multi-view contrastive GNN. Each view has its own nodes and edges; the semantic view is built as a k-NN graph by text/URL similarity, and the behavioral view is a heterogeneous graph with typed relations (1).

$$G=\{G^{(v)}\}_{v\in\{s,u,t\}}, G^{(v)}=\left(V^{(v)}, E^{(v)}\right) \quad (1)$$

Here, $v = s$ denotes the submission view, $v = u$ the behavioral (user) view with typed edges, and $v = t$ the semantic view. For each view we use a node-feature matrix and an adjacency matrix (2).

$$X^{(v)} \in R^{|V^{(v)}| \times d_v}, A^{(v)} \in \{0,1\}^{|V^{(v)}| \times |V^{(v)}|} \quad (2)$$

In the behavioral view we define a set of relation types, and decompose adjacency by relation for R-GCN aggregation:

$R^{(u)} = \{submits, shares\_device, shares\_IP, located\_on, \ldots\}$, supervision is applied on a subset of submission nodes; binary labels are used to train a classifier on the integrated representations (3).

$$L \subseteq V^{(s)}, \quad (3)$$

with labels $y_i \in \{0,1\}$. For local context, let $N_i^{(v)}$ denote the neighborhood of node i i (including a self-loop) in view v v. We describe the architecture layer-wise: for $l = 0, \ldots, L-1$, the transition $d_l ! \to d_{l+1}$ is parameterized by the matrices (4):

$$W^{(v,l)} \in R^{d_{l+1} \times d_l}; \quad (4)$$

in R-GCN we additionally use $W_r^{(l)}$ (for each edge type) and $W_0^{(l)}$ (self-loop).

Supervision is defined on a subset of submission nodes (5):

$$L \subseteq V^{(s)}, y_i \in \{0,1\}, \quad (5)$$

We seek $f_\theta : V^{(s)} \to [0,1] with \hat{y}_i = f_\theta(i)$, the spam probability for submission node i, using the structure and features from all views. For cross-view alignment, we assume anchor correspondences for the same entities (a submission i in s maps to its counterpart node/embedding in t and to related entities in u); these correspondences are later used in contrastive training.

Temporal organization of data: each node/edge has a timestamp τ, training/validation/test are split by time $\left(T_{\text{train}} < T_{\text{val}} < T_{\text{test}}\right)$ to avoid leakage. Online, the graph is built over a sliding window of width W days (with W a fixed hyperparameter).

Labels may be incomplete and noisy: spam is rare, some annotations are wrong or missing, and features can be distorted—texts are obfuscated, and IP/User-Agent values are often shared or spoofed (via NAT, VPN, emulators). Mappings across views are not always complete: for some submissions there is no correct anchor in all views, so we allow missing links. Feature and edge distributions change over time, so we evaluate chronologically with time-based splits. For privacy, identifiers (email, phone, IP) are anonymized: we hash them, aggregate IPs to the /24 level, and store only aggregated or hashed attributes. The data are large and sparse, so we train in mini-batches with neighbor sampling. In the user and submission graphs, edges are directed; in the semantic graph we usually make them bidirectional (k-NN) for stability. Error costs in production are asymmetric: false positives and misses have different impacts, so we tune the decision threshold to the target business metrics.

There are three data views (Fig. 1): submissions (s), users/technical signals (u), and content—texts/URLs (t). In the t-view, nodes are also submissions, connected by k-NN edges based on similarity of their text/URL embeddings. For each view we apply a simple graph layer: standard GAT/GCN for s and t, and a relation-typed layer (R-GCN-like) for u. Within each view we aggregate neighbors (GAT/GCN for s and t; R-GCN for u). We then fuse the three vectors with a mask for missing views and adaptive weights so the model gives more weight to the most informative signal. The fused vector is passed to a lightweight classifier that outputs the spam probability. Training combines a standard binary objective on labeled examples with cross-view contrastive alignment, which reduces labeling needs and improves robustness.

After intra-aggregation, for each submission i we have top-layer vectors $\overline{h_i^{(v)}}$ from views $v \in \{s, u, t\}$. Next, a fusion module with learnable parameters computes node-dependent weights $\gamma_i^{(v)}$. These weights are interpreted as the contribution of each view to the decision and are normalized over the available channels. We record missing channels with a binary mask $m_i^{(v)} \in \{0,1\}$ (0 = missing, 1 = present), so fusion uses only those $\overline{h_i^{(v)}}$, where $m_i^{(v)} = 1$. The final representation z_i: is a weighted combination (via $\gamma_i^{(v)}$ and $m_i^{(v)}$) of the top-layer vectors, with an

internal nonlinearity inside the fusion module. A classifier with parameters (w , b) then maps z_i to $\widehat{y}_i \in [0,1]$ the estimated spam probability. Thanks to $\gamma_i^{(v)}$ the fusion module is interpretable: for each submission we can see which channel—behavioral *u*, semantic *t*, or submission *s* — provides the main signal (the largest $\gamma_i^{(v)}$) Training combines a binary objective on labeled nodes (L_{C}) with cross-view contrastive alignment $(L_{G\mathrm{GL}})$, which pulls together representations of the same submission and pushes apart random pairs. This reduces the need for many labels and improves robustness to drift. For efficiency, we use neighbor sampling, and we pre-build semantic links in the *t* t-view using k-NN indexing of text/URL embeddings.

For the submission view, we use numeric form features: text length, character entropy, digit ratio, URL depth/length, hour of day, and a burst indicator (number of submissions within window Δt) For the semantic/text view, nodes are also submissions, with features given by precomputed 256-dimensional sentence embeddings of the text/URL; adjacency is built by k-NN using cosine similarity. For the behavioral/user view, we build a heterogeneous graph: for nodes of type user/device/ip/page we compute aggregated activity stats (action frequencies, inter-arrival times, page diversity, number of links), encode categorical attributes as fixed-size embeddings, and decompose adjacency by relation types (submits, shares_device, shares_IP, located_on).

Training and evaluation use temporal Train/Val/Test splits; all transformations (scaling, PCA, k-NN indices) are fitted on Train only. Because of class imbalance, the main metric is PR-AUC; we also report F1@τ (τ chosen on Val) and Recall@FP-rate at 10^{-3} and 10^{-4} . The model is compared to common baselines (TF-IDF+LR, Char+XGB, BERT-CLS, single-view GNN). We further assess component contributions via ablations that disable individual views, remove the GCL module, and test alternative fusion and augmentation schemes. In addition, we compare against two simple rules—a keyword-list filter and a fixed cap on submissions from a single IP/device per time window —and report PR-AUC, F1 at the validation-selected threshold, and Recall at fixed FP levels 10^{-3} and 10^{-4} with 95% confidence intervals on the same temporal splits.

4. Conclusions

The proposed architecture performs graph-based web-spam filtering over multiple data views. It operates three complementary channels—form submissions, behavioral/technical relations, and the semantics of text/URLs—and combines intra-view encoding with adaptive cross-view fusion. The contrastive pretraining/joint-training component aligns representations of the same entity across channels, improving robustness to obfuscation and tactical drift. Adaptive fusion adds interpretability: node-dependent weights show which channel contributed most to a given decision. The combined training objective (binary cross-entropy plus contrastive alignment) lowers dependence on large labeled datasets and improves transfer to new campaigns. The evaluation protocol with temporal splits and imbalance-aware metrics (PR-AUC, $F\_1 @ \tau$, Recall@FP) matches production needs and yields reproducible results. Limitations relate to graph construction quality and cost (especially k-NN at large scale), incomplete cross-view correspondences, and the need for periodic re-indexing/retraining under drift. Compute requirements and inference latency depend on local ego-graph density and sampling settings. Overall, the model offers a balanced mix of accuracy, robustness, and practical readiness for real-world web-spam filtering, while meeting privacy constraints and keeping operational risk controlled. Promising directions include continual learning on streaming data, active selection of examples for labeling, richer “hard negative” design in the contrastive module, and integration of differential privacy mechanisms.

Declaration on Generative AI

During the preparation of this work, the authors used ChatGPT to: translate certain text fragments into English, perform grammar and spelling checks, and paraphrase or reword content. After using

these tools, the authors carefully reviewed and edited the content as needed and take full responsibility for the publication's content.

References

[1] Weisen Pan, Jian Li, Lisa Gao, Liexiang Yue, Yan Yang, Lingli Deng, Chao Deng, Semantic Graph Neural Network: A Conversion from Spam Email Classification to Graph Classification, Scientific Programming, vol. 2022, Article ID 6737080, 8 pages, 2022. doi:10.1155/2022/6737080.

[2] Chensu Zhao, Yang Xin, Xuefeng Li, Hongliang Zhu, Yixian Yang, Yuling Chen, An Attention-Based Graph Neural Network for Spam Bot Detection in Social Networks, Applied Sciences, 10(22) (2020) 8160. doi:10.3390/app10228160.

[3] Linjie Shen, Yanbin Wang, Zhao Li, Wenrui Ma, SMS spam detection using BERT and multi-graph convolutional networks, International Journal of Intelligent Networks, 6 (2025) 79–88. doi:10.1016/j.ijin.2025.06.002.

[4] Iryna Zamrii, Ivan Shakhmatov, Vladyslav Yaskevych, BlockchainSQLSecure: Integration of Blockchain to Strengthen Protection Against SQL Injections, Bulletin of Taras Shevchenko National University of Kyiv. Series: Physics and Mathematics, 1 (2024). doi:10.17721/1812-5409.2024/1.29.

[5] Jiangnan Tang, Youquan Wang, Jie Cao, Haicheng Tao, Guixiang Zhu, Inter- and Intra-Graph Attention Aggregation Learning for Multi-relational GNN Spam Detection, Procedia Computer Science, 214 (2022) 1522–1530. doi:10.1016/j.procs.2022.11.339.

[6] Cheng Wu, Chaokun Wang, Jingcao Xu, Ziyang Liu, Kai Zheng, Xiaowei Wang, Yang Song, Kun Gai, Graph Contrastive Learning with Generative Adversarial Network, in: Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '23), ACM, 2023, pp. 2721–2730. doi:10.1145/3580305.3599370.

[7] Xuxu Zheng, Chen Feng, Zhiyi Yin, Jinli Zhang, Huawei Shen, Research on Fraud Detection Method Based on Heterogeneous Graph Representation Learning, Electronics, 12(14) (2023) 3070. doi:10.3390/electronics12143070.

[8] Jinbo Chao, Chunhui Zhao, Fuzhi Zhang, Network Embedding-Based Approach for Detecting Collusive Spamming Groups on E-Commerce Platforms, Security and Communication Networks, vol. 2022, Article ID 4354086, 11 pages. doi:10.1155/2022/4354086.

[9] Luzhi Wang, Yizhen Zheng, Di Jin, Fuyi Li, Yongliang Qiao, Shirui Pan, Contrastive Graph Similarity Networks, ACM Transactions on the Web, 18(2) (2024) Article No. 17, pp. 1–20. doi:10.1145/3580511.

[10] Yupeng Hou, Binbin Hu, Wayne Xin Zhao, Zhiqiang Zhang, Jun Zhou, Ji-Rong Wen, Neural Graph Matching for Pre-training Graph Neural Networks, in: Proceedings of the 2022 SIAM International Conference on Data Mining (SDM), SIAM, 2022, pp. 738–747. doi:10.1137/1.9781611977172.20.

[11] Ijeoma A. Chikwendu, Xiaoling Zhang, Chiagoziem C. Ukwuoma, Okechukwu C. Chikwendu, Yeong Hyeon Gu, Mugahed A. Al-antari, Spectrum-Constrained and Skip-Enhanced Graph Fraud Detection: Addressing Heterophily in Fraud Detection with Spectral and Spatial Modeling, Symmetry, 17(4) (2025) 476. doi:10.3390/sym17040476.

[12] Saif Safaa Shakir, Leyli Mohammad Khanli, Hojjat Emami, Convolutional Graph Network-Based Feature Extraction to Detect Phishing Attacks, Future Internet, 17(8) (2025) 331. doi:10.3390/fi17080331.

[13] Zhiwei Liu, Yingtong Dou, Philip S. Yu, Yutong Deng, Hao Peng, Alleviating the Inconsistency Problem of Applying Graph Neural Network to Fraud Detection, in: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '20), ACM, 2020, pp. 1569–1572. doi:10.1145/3397271.3401253.

[14] María Novo-Lourés, David Ruano-Ordás, Reyes Pavón, Rosalía Laza, Silvana Gómez-Meire, José R. Méndez, Enhancing representation in the context of multiple-channel spam filtering, Information Processing & Management, 59(2) (2022) 102812. doi:10.1016/j.ipm.2021.102812.

[15] Sawsan Alshathtawi, Amani Shatnawi, Anas M.R. AlSobeh, Aws A. Magableh, Beyond Word-Based Model Embeddings: Contextualized Representations for Enhanced Social Media Spam Detection, Applied Sciences, 14(6) (2024) 2254. doi:10.3390/app14062254.
[16] Venkateswarlu B, Viswanath Shenoi V, Optimized generative adversarial network with fractional calculus based feature fusion using Twitter stream for spam detection, Information Security Journal: A Global Perspective, (2021). doi:10.1080/19393555.2021.1956024.
[17] Paul Ntim Yeboah, A. S. M. Kayes, Wenny Rahayu, Eric Pardede, Syed Mahbub, A Framework for Phishing and Web Attack Detection Using Ensemble Features of Self-supervised Pre-trained Models, TechRxiv, (2025). doi:10.36227/techrxiv.173603362.21995515/v1.
[18] Zheng Qu, Qingyao Jia, Chen Lyu, Jia Liu, Xiaoying Liu, Kechen Zheng, Detecting Fake Reviews with Generative Adversarial Networks for Mobile Social Networks, Security and Communication Networks, (2022) 1164125. doi:10.1155/2022/1164125.
[19] Yonghong Huang, Joanna Negrete, John Wagener, Celeste Fralick, Armando Rodriguez, Eric Peterson, Adam Wosotowsky, Graph neural networks and cross-protocol analysis for detecting malicious IP addresses, Complex & Intelligent Systems, 9 (2023) 3857–3869. doi:10.1007/s40747-022-00882-2.
[20] Naeem Ahmed, Rashid Amin, Hamza Aldabbas, Deepika Koundal, Bader Alouffi, Tariq Shah, Machine Learning Techniques for Spam Detection in Email and IoT Platforms: Analysis and Research Challenges, Security and Communication Networks, (2022) 1862888. doi:10.1155/2022/1862888.
[21] Ruchi Agarwal, Anshita Dhoot, Surya Kant, Vimal Singh Bisht, Hasmat Malik, Md. Fahim Ansari, A Novel Approach for Spam Detection Using Natural Language Processing With AMALS Models, IEEE Access, 12 (2024) 124298–124313. doi:10.1109/ACCESS.2024.3391023.
[22] Abdulla Al-Subaiey, Mohammed Al-Thani, Naser Abdullah Alam, Kaniz Fatema Antora, Amith Khandakar, S. M. Ashfaq Uz Zaman, Novel interpretable and robust web-based AI platform for phishing email detection, Computers and Electrical Engineering, 120 Part A (2024) 109625. doi:10.1016/j.compeleceng.2024.109625.