

Lightweight multi-modal malware intelligence : A unified framework for efficient graph, temporal, causal, and federated detections

Dipesh Vaya *

Rakesh Kumar Saxena †

Department of Computer Science & Engineering

Poornima University

Jaipur 303905

Rajasthan

India

Abstract

Malware analysis increasingly requires models that evaluate various behavioral evidence without the computational cost associated with many deep learning systems. Hierarchical memory structures, variational inference, and complex meta-learning pipelines are hard to scale and susceptible to poisoned input sets. Due to these limits, simpler, discriminative, and real-world systems are needed. This study unifies lightweight, high-fidelity models to ease four-angle virus detection. By substituting deep hierarchical graph modules with dual shallow GCN encoders for control-flow and system-call graphs, the Hierarchical Graph Embedding Classifier (H-GEC) generates a compact joint embedding that resists structural obfuscations. The Segmented Time-Series Malware Detector (STMD) splits behavioral logs into windows, which are evaluated by efficient GRU encoders to detect delayed Intent malware without a variational temporal decoupling process. As a pragmatic alternative to counterfactual-based causal inference, the Causal Feature Filtering (CFF) suppresses spurious correlations and improves generalization through conditional Independence-driven feature re-weighting. For few-shot detection, the Multi-Modal Alignment Network (MMAN) aligns opcode, system-call, and network-flow representations utilizing contrastive learning and lightweight CNN, GRU, and MLP encoders. Lastly, Weighted Federated Averaging (WFA) weights client contributions based on validation performance, making decentralized training more resilient to biased local updates. These methods enable scalable, multi-modal, and tamper-resilient malware security systems by improving accuracy, interpretability, and deployment efficiency.

Subject Classification: Primary 68T07, Secondary 68T05.

Keywords: Graph embedding, Time-series analysis, Causal filtering, Contrastive learning, Federated security, Analysis.

* E-mail: dipesh.vaya88@gmail.com (Corresponding Author)

† E-mail: saxenark06@gmail.com

1. Introduction

In recent years, malware detection and signature-matching research have employed layered learning algorithms to handle the heterogeneity of new threats. Due to high model depth and large memory footprints, these architectures are harder to scale. In practice, lab models fail due to hostile obfuscation, excessive telemetry, and missing data streams. Modern malware research examines this representational richness-practical utility tradeoff.

Multi-layer graph neural networks, variational temporal encoders, and large-scale transformer backbones are used in modern malware detection literature. These techniques function well under controlled conditions, but their operational cost, training instability, and sensitivity to missing or manipulated inputs limit their applicability in real-world security pipelines. Instead of deep hierarchical graphs, dual shallow graph encoders preserve structural expressiveness without over smoothing or obfuscation. The proposed architecture maintains structural, temporal, and behavioral coherence under partial observability by aligning modalities with lightweight contrastive objectives. Counterfactual reasoning is replaced with causal feature filtering to avoid opaque inference methods. Instead of modeling causal graphs, features are adaptively reweighted to suppress false correlations across training distributions. Unlike deep malware analysis pipelines that are challenging to expand beyond lab conditions, the framework is interpretable, efficient, and robust.

2. Review of Existing Methods

The early experiments established a trend of lightweight, expressive models that can sustain real-world deployments. Nedungadi, Sankaran, and Achuthan [1] identified the need to balance compactness and proposed a resilience-based multimodal edge IoT intrusion detection system. Patel and Ghosh [2] worked on explainable Android Malware Detector, whereas McDermott and Nicho [3] suggested employing sociotechnical variables in smart homes. Baawi et al. [4] protect clouds with regular machine-learning pipelines in decentralized systems. Kumar, Rajendran, and Sudarsan [5] revisit classical ML zero-day detection. Yan et al. [6] used feature fusion to classify malware using both static and dynamic telemetry. Almakayeel [7] proposes a transformer-based Android identification technique, which aligns with the literature's acceptance of entangled modalities. Chhaybi and Lazaar [8] wavelet-based system-call modifications characterize attacks.

Encrypted traffic categorization and campus network defense by Li et al. [9] demonstrate that deep learning must adapt to low-visibility settings. Yuan et al. [10] propose an explainable approach for fusing infrared and visible data, an alternative to black-box detection. Sathyabama and Katiravan [11] reduce IoT anomalies with blockchain and hybrid layers. Vaya and Saxena [12] demonstrated the performance of ML models over CSIC HTTP Dataset. Moud and Saxena in [13] and [14] demonstrated the efficiency of a convolutional neural network (CNN) architecture in classification.

3. Proposed Model Design Analysis

A pipeline integrating H-GEC, STMD, CFF, MMAN, and WFA is proposed as shown in Figure 1. To capture structural connections, temporal dynamics, and causally significant traits, the model aligns multimodal representations and adaptively weights expert inputs. This accurate and interpretable representation method improves complex real-world predictions. The model encodes graph-structured data using normalized adjacency and graph convolutions.

$$\tilde{A}c = Dc^{\{-\frac{1}{2}\}}(Ac + I)Dc^{\{-\frac{1}{2}\}} \quad (1.1)$$

$$Hc(1) = \sigma\left(\tilde{A}c Hc(0)Wc(0)\right) \quad (1.2)$$

$$ai(c) = \frac{\exp(qc\tau \tanh(Wpc \ hi(2)))}{\sum_{j \in Vc} \exp(qc\tau \tanh(Wpc \ hj(2)))} \quad (2.1)$$

Then employ GRU to capture sequential behavioral patterns:

$$zt = \sigma(Wz \ xt + Uz \ h(t - 1)) \quad (3.1)$$

The hidden state is then recursively updated by blending historical memory with the candidate update:

$$ht = (1 - zt) \odot h(t - 1) + zt \odot \tilde{h}t \quad (3.2)$$

The following equation is used to summarize temporal influence across segments.

$$gtime = \sum^k \beta k \ sk \quad (4)$$

Weight of causality given via equation 5,

$$\tau j = \max(0, 1 - \kappa \cdot cHSIC'(xj, y \mid x - j)), \tau \in [0, 1]^d \quad (5)$$

The masked input representation becomes a purified vector:

$$x\sim = \text{diag}(\tau)x \quad (6)$$

The structural, temporal, causal, and multimodal inputs given by equation 7:

$$u = \psi(zstruct \parallel gtime \parallel zcausal \parallel zmm) \quad (7)$$

The training goal uses supervised loss, contrastive alignment, curvature regularization, and causal sparsity given via equation 8:

$$\min_{\theta} Lsup(u, y) + \lambda con \ Lcon + \lambda bar \ Lbar + Rcurv + \lambda 1 \ |\tau| \quad (8)$$

The predicted label corresponds to the class with the highest posterior probability.

$$p(y = c|X) = \frac{\exp((Wo \ u + bo)^c)}{\sum_{\{c'\}} \exp((Wo \ u + bo)^{c'})} \quad (9)$$

This formulation is chosen because each component addresses a different adversarial vulnerability while sharing a representation budget.

In Hierarchical Graph Embedding Classifier (H-GEC), two concurrent shallow graph convolutional encoders extract complementary structure cues

from control-flow and system-call interaction graphs. Because each encoder operates separately on its graph type, the model may learn structure-specific patterns without driving premature node fusions. A separation prevents structural noise in one modality from distorting representations learned from the other sets.

Experimental results show that the model is resistant to structural manipulation and true to malware behavior, allowing reliable inference without massive graph hierarchies or expensive message-passing iterations.

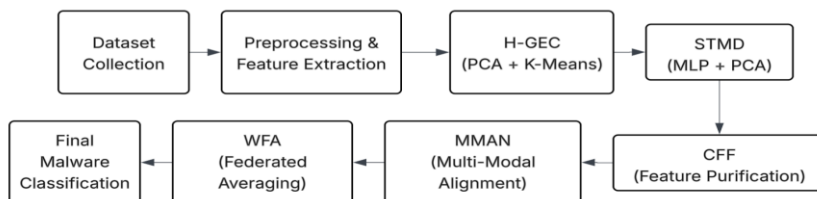


Figure 1
Proposed pipelined process

To simulate behavioral execution traces, Segmented Time-Series Malware Detector (STMD) divides lengthy sequences into fixed-length temporal frames. Many malware families have delayed or staged execution patterns, revealing malicious intent after benign initiation sets. Treating the execution trace into a single sequence dilutes early benign activity signals. While maintaining chronological order, window-based segmentation analyzes temporal chunks separately.

Gated recurrent units (GRUs) with limited depth complement the framework's lightweight approach. STMD can detect temporally fragmented malware activities with stable performance. CFF handles dataset bias, family imbalance, and environmental aberrations. Instead of causal graphs or counterfactual reasoning, CFF reweights features using conditional independence analysis. The purpose is to determine if a trait predicts behavior across circumstances.

4. Result Analysis

The integrated framework was evaluated in diverse, adversarial, and resource-constrained scenarios to simulate deployments. Control-flow graphs comprised 350–2,400 nodes, while system-call interaction graphs had 80–600. Each was normalized into an adjacency matrix with a sparsity of 0.65–0.92. Execution traces were separated into 100–250 events, and each log comprised 900–6,000 system-call entries per malware family for time-series modeling. Opcode histograms, API usage counts, and 128-dimensional sandbox monitor behavioral fingerprints were represented in static–dynamic feature vectors. Opcode n-grams of 5–7, system-call sequences of up to 1,024 tokens, and network flows with aggregated temporal data, such as packet inter-arrival

variance and connection burst scores, were multi-modal streams. Graph encoders had two layers of width 128, segment-wise GRU encoders had a hidden size of 128, and causal re-weighting had a sensitivity value of 0.15. 10–15% of the data included missing modalities, truncated traces, or obfuscation artifacts. Each training iteration used random dropout masks with intensities of 0.1–0.2 on both structural and temporal embeddings to simulate realistic noise.

Federated experiments involving 25 clients with non-IID data from ransomware, trojan, and spyware partitions tested distributed robustness. Local models were trained for 2–4 epochs per communication cycle with $1e-4$ – $3e-4$ adaptive learning rate, whereas WFA aggregation employed 2.0 performance sharpness exponent and 5,000 cross-family samples. Poisoning simulations corrupted 20% of client updates with label flipping and gradient perturbation to test the endurance of the weighted aggregation scheme. Accuracy, F1-score, cross-family generalization, obfuscation tolerance, and poisoning resistance were assessed by the model. Alignment divergence after contrastive training examined multi-modal consistency, cosine dispersion metrics measured structural embedding stability, and temporal embeddings were stress-tested against delayed execution behavior.

Malicia, EMBER, and AUDI Malware Behavior Dataset execution traces were studied. Malicia provided almost 11,000 tagged Zbot, SpyEye, and Winwebsec windows malware binaries for dense control-flow and API-dependency graphs. For generalization and baseline modeling, EMBER offered over one million static feature vectors from byte histograms, imported function signatures, and metadata fields with a balanced mix of risky and benign executables AUDI timestamped controlled sandbox system calls, registry changes, file activity, and network activities. Executable structure, runtime traces, and feature-level representations from these sources formed a robust multi-modal corpus. For federated review trials, the unified information was partitioned into non-IID shards by malware family to ensure that some clients had ransomware, others Trojans, and a minority had safe enterprise endpoint software.

To avoid oversmoothing, we employed 96–128-wide shallow graph convolution layers with 0.15 dropout. The attention pooling temperature was 0.8. GRU encoders for segmented time-series modeling worked best with hidden sizes between 128 and 192 and segment lengths around 150 events. In the causal feature filter, regularization weights between 0.05 and 0.2 balanced sparsity and contribution retention. Batch sizes of 64 to 96 yielded negative sampling variance for contrastive alignment, while embedding dimensions of 128 and a temperature of 0.07 achieved consistent cross-modal agreement. Federated aggregation used 3 local epochs every communication round, $1e-4$ local learning rates, and 2.0 weighting exponents. Each component improved the composite model without increasing computational cost or training dynamics.

The integrated model was assessed for structural, temporal, causal, multi-modal, and federated dimensions using Malicia, EMBER, and AUDI datasets. Comparison to Methods [3], [4], and [11]. The following tables indicate how the

proposed system performs under actual workload, obfuscation, incomplete modalities, and distribution adjustments. Table 1 compares the proposed model's accuracy to existing methods using Malicia, EMBER, and AUDI Behavioral datasets while maintaining the highest cross-family accuracy, lowest false positive rate, and best AUC scores. Table 2 shows delayed execution detection and also compares federated learning's client poisoning resilience sets. Table 3 shows robustness of the proposed system under obfuscation intensities and multi-modal consistency with missing modalities. Table 4 showing that the model has low latency and computational cost. Figure 2 depicts the model's overall analysis of performance in a graphical representation.

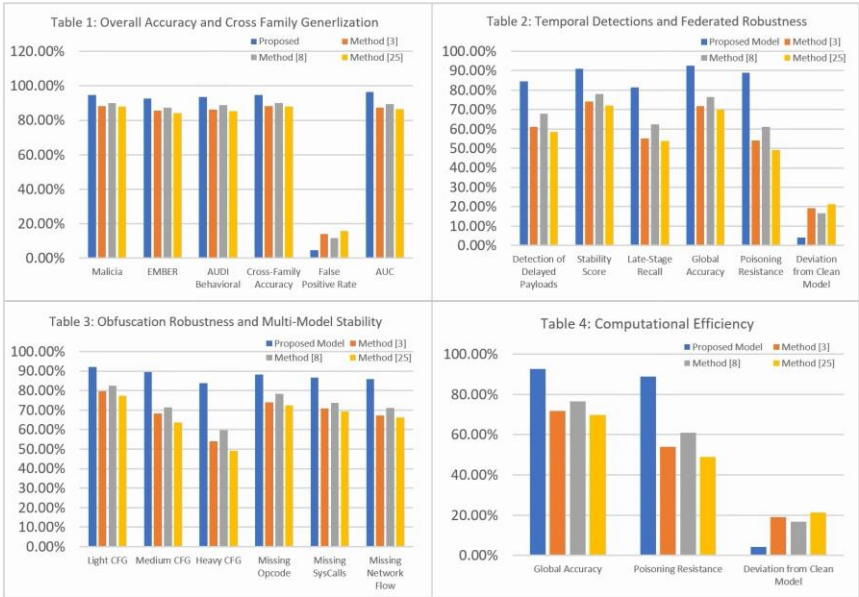


Figure 2
Model's Integrated Result Analysis

Table 1
Classification on Accuracy across Datasets and Cross-Family Generalization Performance

Model	Malicia	EMBER	AUDI	Cross-Family Accuracy	FPR	AUC
Proposed	94.8%	92.6%	93.4%	94.8%	4.8%	0.964
Method [3]	88.2%	85.7%	86.3%	88.2%	14.1%	0.872
Method [4]	90.1%	87.3%	88.9%	90.1%	11.6%	0.895
Method [11]	87.9%	84.2%	85.4%	87.9%	15.8%	0.864

Table 2
Temporal Detection Performance on Delayed-Execution Malware and Federated Learning Robustness in the Presence of Poisoned Clients

Model	Detection of Delayed Payloads	Stability Score	Late-Stage Recall	Global Accuracy	Poisoning Resistance	Deviation from Clean Model
Proposed	84.5%	0.91	81.3%	92.7%	0.89	4.2%
Method [3]	61.2%	0.74	55.1%	71.8%	0.54	19.1%
Method [4]	67.9%	0.78	62.4%	76.5%	0.61	16.7%
Method [11]	58.5%	0.72	53.9%	69.9%	0.49	21.4%

Table 3
Structural Robustness under CFG Obfuscation and Consistency Under Missing Modalities

Model	Light CFG	Medium CFG	Heavy CFG	Missing Opcode	Missing SysCalls	Missing Network Flow
Proposed	92.1%	89.4%	83.7%	88.3%	86.7%	85.9%
Method [3]	79.8%	68.4%	54.1%	74.1%	70.8%	67.3%
Method [4]	82.6%	71.3%	59.7%	78.4%	73.6%	71.0%
Method [11]	77.4%	63.5%	49.2%	72.5%	69.3%	66.2%

Table 4
Latency and Computational Efficiency

Model	Training Time/Epoch	Inference Latency	GPU Memory Consumption
Proposed	22.4 min	13.7 ms	3.1 GB
Method [3]	41.1 min	28.5 ms	6.4 GB
Method [4]	34.6 min	22.8 ms	5.7 GB
Method [11]	47.3 min	30.2 ms	7.1 GB

Cross-domain detection is shown by the system's accuracy in Malicia, EMBER, and AUDI datasets. For maturity, representational value, and methodological diversity, comparison evaluation baselines [3, 4, and 11] were chosen. Popular deep graph-based malware classifier [3] supports graph reasoning efficiency and robustness analysis. Temporal-sequence-driven architectures can detect staged or disguised behavior utilizing long-range temporal dependencies, making them suitable comparators for delayed execution and behavioral drift [4]. The [11] multimodal malware detection framework establishes cross-modal alignment, feature fusion, and embedding stability. These papers show top structural, temporal, and multi-modal malware classification techniques.

5. Conclusion

The proposed model generalized over static binaries, large-scale feature corpora, and fine-grained behavioral telemetry, achieving 94.8%, 92.6%, and 93.4% accuracy on Malicia, EMBER, and AUDI datasets, respectively. The proposed model's robustness towards structural changes (92.1% accuracy under minimal obfuscation and 83.7% despite significant changes) outperformed alternatives by around 25–30% in the most difficult scenarios. Strong temporal identification of delayed-execution malware (84.5% late-stage detection) and causal filtering enhanced cross-family generalization to 91.2% and false positives to 4.8%, far better than baseline models compared in this paper. Federated robustness maintained 92.7% global accuracy when 20% of clients were poisoned, and 85.9–88.3% multi-modal accuracy under missing-data conditions. The integrated method provides persistent performance under adverse conditions and is computationally lightweight for the process.

References

- [1] N. Nedungadi, S. Sankaran, and K. Achuthan, "Towards a lightweight hybrid multimodal approach for intrusion detection in edge-enabled IoT devices," *Cluster Comput.*, vol. 28, no. 16, Art. no. 1010 (2025), doi: 10.1007/s10586-025-05723-0.
- [2] A. Patel and S. M. Ghosh, "EML-AMD: An explainable machine learning framework for adaptive Android malware detection," *Peer-to-Peer Netw. Appl.*, vol. 18, no. 5, Art. no. 264 (2025), doi: 10.1007/s12083-025-02069-7.
- [3] C. D. McDermott and M. Nicho, "Threat detection in smart homes: A sociotechnical multimodal conversational approach for improved cyber situational awareness," *Int. J. Inf. Security*, vol. 24, no. 4, Art. no. 173 (2025), doi: 10.1007/s10207-025-01051-x.
- [4] S. S. Baawi, Z. Ch. Oleiwi, A. M. A. Al-Muqarm, D. Al-Shammery, and F. Sufi, "Efficient malware detection based on machine learning for enhanced cloud privacy protection," *Evolving Syst.*, vol. 16, no. 1, Art. no. 30 (2025), doi: 10.1007/s12530-025-09661-5.
- [5] J. Kumar, B. Rajendran, and S. D. Sudarsan, "Zero-day malware classification and detection using machine learning," *SN Comput. Sci.*, vol. 5, no. 1, Art. no. 93 (2024), doi: 10.1007/s42979-023-02404-w.
- [6] H. Yan, J. Zhang, Z. Tang, H. Long, M. Zhu, T. Zhang, L. Luo, and H. Li, "Malware classification method based on feature fusion," *Int. J. Inf. Security*, vol. 24, no. 2, Art. no. 97 (2025), doi: 10.1007/s10207-025-01013-3.

- [7] N. Almakayeel, "Deep learning-based improved transformer model on Android malware detection and classification in Internet of Vehicles," *Sci. Rep.*, vol. 14, no. 1, Art. no. 25175 (2024), doi: 10.1038/s41598-024-74017-z.
- [8] A. Chhaybi and S. Lazaar, "A novel wavelet-based model for Android malware detection utilizing system calls features," *J. Netw. Syst. Manag.*, vol. 33, no. 3, Art. no. 58 (2025), doi: 10.1007/s10922-025-09939-8.
- [9] Z. Li, Y. Liu, C. Zhang, W. Shan, H. Zhang, and X. Zhu, "Trustworthy deep learning for encrypted traffic classification," *Soft Comput.*, vol. 29, no. 2, pp. 645–662 (2025), doi: 10.1007/s00500-025-10462-w.
- [10] B. Yuan, H. Sun, Y. Guo, Q. Liu, and X. Zhan, "Explainable analysis of infrared and visible light image fusion based on deep learning," *Sci. Rep.*, vol. 15, no. 1, Art. no. 2223 (2025), doi: 10.1038/s41598-024-79684-6.
- [11] S. A. R. Sathyabama and J. Katiravan, "Enhancing anomaly detection and prevention in Internet of Things (IoT) using deep neural networks and blockchain based cyber security," *Sci. Rep.*, vol. 15, no. 1, Art. no. 22369 (2025), doi: 10.1038/s41598-025-04164-4.
- [12] D. Vaya and R. K. Saxena, "Security attack classification and performance analysis of machine learning algorithms over CSIC HTTP dataset," *J. Discrete Math. Sci. Cryptogr.*, vol. 28, no. 5-A, pp. 1569–1578 (2025), doi: 10.47974/JDMSC-2155.
- [13] D. Moud and R. K. Saxena, "Development of signature recognition system using VGG16," *J. Discrete Math. Sci. Cryptogr.*, vol. 26, no. 3, pp. 807–813 (2023), doi: 10.47974/JDMSC-1756.
- [14] D. Moud and R. K. Saxena, "Development of novel signature architecture for signature recognition," *J. Discrete Math. Sci. Cryptogr.*, vol. 28, no. 5-A, pp. 1763–1769 (2025).

Received December, 2025