M-GRAD: Fusing Heterogeneous Logs into AI-Driven Graphs for Anomaly Detection and IP Risk Analysis

Çağatay Ateş[1], Ilgın Şafak[2], Fatih Alagöz[3] and Emin Anarım[1]

***Abstract*— Enterprise security demands actionable threat prioritization across heterogeneous environments. To address the limitations of siloed log analysis, we propose a multi-source, graph-based anomaly detection framework. First, our architecture seamlessly fuses network telemetry from Vectra, Security Information and Event Management (SIEM) logs from QRadar, and endpoint data from CrowdStrike into a unified graph. Second, we utilize the Louvain algorithm to segment the network, identifying functional communities and critical bridge nodes indicative of lateral movement. Third, we extract over 50 structural, behavioral, and temporal features to train an XGBoost-based ensemble classifier. Evaluated on a real-world banking dataset with 31 anomaly types, the proposed system achieves 98% precision. Crucially, empirical analysis reveals that local neighborhood connectivity and traffic dynamics significantly outperform traditional global centrality metrics in detecting complex attacks. Finally, to support operational incident response, we implement a dynamic risk-scoring mechanism that translates model predictions into actionable, IP-level severity assessments across sliding time windows. Ultimately, this framework substantially enhances comprehensive threat visibility and analyst efficiency.**

I. Introduction

To counter sophisticated cyber threats, modern enterprises particularly in critical infrastructures like banking, manufacturing, smart grids must analyze massive volumes of heterogeneous telemetry from SIEM, IDS (Intrusion Detection System), and endpoint solutions [1]. However, traditional detection systems typically analyze these sources in isolation, missing the contextual correlation required to identify coordinated, multi-stage attacks [2]. Consequently, analysts are burdened with manual, error-prone synthesis [3], while conventional signature-based methods struggle to establish reliable baselines or detect novel threats in highly dynamic environments.

Graph-based representations provide a robust alternative by intrinsically modeling the structural and temporal relationships between network entities [4]. By representing IPs as nodes and communication flows as edges, graph-theoretic metrics and community detection algorithms can expose anomalous topological patterns that remain invisible to standard feature-based methods. Nevertheless, effectively fusing heterogeneous, multi-source telemetry into a unified, temporally dynamic graph remains a critical open challenge.

The key contributions of this study are summarized as follows:

- A data integration framework is proposed to automatically normalize and fuse heterogeneous telemetry (Vectra, QRadar, CrowdStrike) into a unified graph representation, ensuring seamless interoperability through a common schema.
- A graph segmentation methodology utilizing the Louvain algorithm is introduced to partition the network. This process identifies functional communities and critical bridge nodes, exposing topological patterns highly indicative of lateral movement and data exfiltration.
- A robust feature extraction methodology is applied, deriving over 50 multi-faceted attributes across six categories (centrality, neighborhood, traffic, community, temporal, and graph-level) to effectively capture both the structural and behavioral dynamics of network entities.
- Finally, a dynamic risk-scoring mechanism is implemented to translate raw model predictions into actionable, IP-level severity assessments. By synthesizing prediction confidence with event severity across sliding windows, this approach provides prioritized, operational intelligence for security analysts.

This remainder of this paper is organized as follows. Section II reviews related work in network anomaly detection, graph-based security analytics, and multi-source data fusion. Section III presents the proposed approach. Section IV gives the simulation results. Finally, Section V concludes the paper by discussing the key findings and future directions.

II. Related Work

Recent advances in cybersecurity analytics reveal a paradigm shift from traditional rule-based systems toward hybrid machine learning and graph-based models. Accordingly, this section reviews the foundational literature driving the proposed architecture, specifically focusing on hybrid log processing, graph-based community detection, and XGBoost integration.

To overcome the limitations of siloed telemetry, multi-source data fusion has become essential in modern cybersecurity. For instance, the AI-driven IP SafeGuard framework [5] calculates a Dynamic Threat Score by integrating external threat intelligence with internal network logs, achieving

[1]Çağatay Ateş and Emin Anarım are with Department of Electrical and Electronics Engineering, Boğaziçi University, Istanbul, Türkiye, {cagatay.ates, anarim}@bogazici.edu.tr

[2]Ilgın Şafak is with Fibabanka R&D Center, Istanbul, Türkiye, ilgin.safak@fibabanka.com.tr and University of Jyväskylä, IT Faculty, Jyväskylä, Finland, ilgin.i.safak@jyu.fi

[3]Fatih Alagöz is with Department of Computer Engineering, Boğaziçi University, Istanbul, Türkiye, fatih.alagoz@bogazici.edu.tr

This work was supported by Business Finland (BF) within the EUREKA CELTIC-NEXT project CISSAN (www.celticnext.eu). The authors would like to thank Fibabanka, Bogaziçi University and University of Jyväskylä management for their support.

98.2% accuracy and a 45-millisecond detection time via XGBoost. Similarly, the MF-CSSA model [6] leverages sensor fusion, probabilistic reasoning, and graph theory across diverse log sources to deliver real-time situational awareness with 88% accuracy and an 8% false positive rate. Furthermore, the hybrid GConvTrans architecture [7] translates network traffic into computational graphs, fusing Transformers with Graph Convolutional Networks (GCN) to seamlessly capture both localized structural patterns and global attack contexts.

Graph-based community detection is vital for unraveling complex attack patterns. For example, FedGATSage [8] clusters IoT devices using Graph Attention Networks (GAT) and the Louvain algorithm, successfully detecting structural attacks while reducing communication costs by 85%. Similarly, the hierarchical 'Walk-and-Classify' model [9] enriches Random Walk with Restart (RWR) with temporal awareness, leveraging an XGBoost-based detector to cleanly isolate malicious sub-communities. Underscoring XGBoost's efficacy in processing high-dimensional graph features, recent hybrid systems pair Graph Neural Networks (GNN) for spatial dependencies with XGBoost for robust predictive analytics which are effectively identifying IoT anomalies [10] and dramatically reducing underreported financial fraud via multi-relational graphs like Tri-RGCN-XGBoost.

Despite recent advancements, several key research gaps remain. First, there is a lack of unified graph frameworks that integrate heterogeneous logs into a single comprehensive representation. Second, community structures and bridge nodes are rarely exploited as primary features for multi-class anomaly detection. Third, existing models often lack the temporal granularity needed to capture evolving attack behaviors. Furthermore, there is an absence of interpretable, IP-level risk scoring mechanisms derived directly from model predictions. Finally, empirical evaluations on complex, highly imbalanced real-world banking datasets remain scarce.

III. Methodology

This section presents the overall architecture of the proposed approach. The system is designed as a modular pipeline that processes heterogeneous security data through seven interconnected modules, transforming raw network events into actionable risk assessments. Figure 1 illustrates the high-level system architecture, showing the flow from multi-source data ingestion to final risk visualization.

The framework consists of seven sequential modules, each responsible for a specific transformation or analysis task. The pipeline begins with data preprocessing and graph construction, progresses through graph segmentation and feature extraction, performs model training and evaluation, and concludes with IP-level risk analysis. This modular design enables independent development and testing of each component while maintaining a connected end-to-end workflow. Each module is explained in the following subsections.

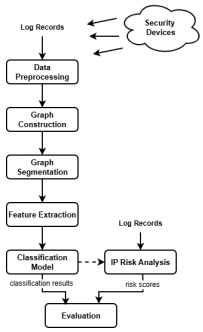

Fig. 1. The block diagram of the proposed approach.

A. Data Preprocessing Module

The data preprocessing stage integrates heterogeneous security data by normalizing diverse schemas and temporal resolutions into a unified representation. This pipeline resolves schema disparities and aligns timestamps while standardizing core event attributes such as source IPs and severity levels. Crucially, ground truth labels are generated by mapping each IP address to its predominant anomaly type within defined time windows, thereby establishing the target variables for supervised learning.

Subsequent to integration, a temporal train-test split is implemented to ensure training data chronologically precedes test sets, simulating realistic deployment scenarios and preventing data leakage. The dataset is further refined through feature standardization to balance numerical scales and categorical encoding for algorithmic compatibility. Finally, feature selection techniques, such as low-variance removal, are applied to eliminate uninformative attributes and minimize noise in the model.

B. Graph Construction Module

Following preprocessing, network event data is transformed into directed graph structures where nodes represent IP addresses and edges represent connections. To capture traffic characteristics, edge attributes encode key metrics such as data volume and event frequency. A sliding window approach segments continuous activity into discrete graphs, preserving structural relationships essential for community detection. This temporal segmentation balances the granularity required for pattern recognition with computational efficiency.

C. Graph Segmentation Module

Graph segmentation partitions each temporal graph into communities using the Louvain algorithm [11]. This method identifies natural node groupings by optimizing modularity (Q), a quality metric that measures the strength of community structure by comparing the actual density of connections within a group against a random distribution of links:

$$Q = \frac{1}{2m} \sum_{ij} \left[A_{ij} - \frac{k_i k_j}{2m} \right] \delta(c_i, c_j), \tag{1}$$

where A_{ij} is the adjacency matrix, k_i, k_j are node degrees, m is the total edge count, and $\delta(c_i, c_j)$ ensures only intra-community pairs contribute to the sum. The algorithm employs an iterative two-phase process. In the local optimization phase, nodes are greedily reassigned to neighboring communities to maximize modularity. In the aggregation phase, these communities are collapsed into super-nodes to form a new network. This cycle repeats until convergence. The algorithm's efficiency ensures scalability for large networks, while its resolution parameter allows fine-tuning in which higher values detect smaller, numerous clusters, while lower values favor larger communities. For each window, this process produces a partition mapping that assigns IP addresses to community identifiers. This segmentation reveals functional groups and identifies critical bridge nodes which are entities connecting otherwise isolated communities. Such topological patterns are of high polemical interest, as they frequently signal lateral movement or data exfiltration attempts. By providing a hierarchical view of local clustering and global connectivity, Louvain-based segmentation becomes a robust foundation for identifying critical infrastructure and potential attack samples.

D. Feature Extraction Module

Feature extraction constructs a comprehensive set of features for each node across all graph windows, transforming the graph structure and community assignments into numerical feature vectors suitable for machine learning. The feature set comprises over 50 features organized into six categories, each capturing different aspects of network behavior and structure.

1) Centrality Features: Centrality features quantify the structural importance and influence of nodes within the network topology. Degree-based features (in-degree, out-degree, and total degree) evaluate immediate connectivity, while the degree ratio captures connection asymmetry, effectively distinguishing behavioral roles such as clients and servers. Betweenness centrality measures the fraction of shortest paths traversing a node, identifying critical structural bridges, bottlenecks, or potential attack targets. Closeness centrality evaluates the inverse of the average shortest path to all reachable nodes, highlighting critical communication hubs that facilitate rapid information flow. To capture hierarchical influence, PageRank iteratively computes node importance based on incoming links from other prominent nodes, proving particularly effective for identifying command-and-control servers in directed network topologies. Katz centrality, [12], broadens this concept by considering paths of all lengths with exponentially decreasing weights, thus capturing both direct and indirect global network impact. Finally, Eigenvector centrality assesses a node's significance based on the centrality of its connected neighbors, effectively revealing highly influential network clusters.

2) Neighborhood Features: Neighborhood features characterize the local connectivity structure to identify localized anomalous behavior. To evaluate local density, the clustering coefficient measures existing triangles around a node; high values often indicate tightly bounded functional groups or coordinated attack clusters, such as botnet cells. Similarly, average neighbor degree quantifies immediate connectivity to capture the rich-get-richer phenomenon, assessing if an IP interacts primarily with highly connected hubs. To identify deeply embedded nodes, the k-core number highlights core network regions representing critical infrastructure or strongly coordinated malicious samples. Complementing this, triangle count measures the absolute number of local 3-cliques, revealing IPs structurally fixed within tightly connected patterns.

3) Traffic Features: Traffic features quantify communication patterns by aggregating total bytes (sent, received, combined), event counts, and average bytes per edge. To highlight connection asymmetry, the bytes ratio mathematically divides outbound by inbound traffic, serving as a key indicator of specific attack behaviors. Together, these metrics capture distinct footprints: data exfiltration typically manifests as high outbound volume, command-and-control as high inbound traffic, and network scanning as frequent low-byte connections.

4) Community Features: Derived from graph segmentation, community features capture a node's structural role within and across partitions. Community size indicates the scale of these functional groups, while intra-community and inter-community degrees delineate internal and external interactions. Building upon these, the bridge score quantifies boundary-spanning behavior using the ratio of inter-community edges to total connectivity. High bridge scores highlight critical nodes connecting otherwise isolated groups, making them prime indicators of potential lateral movement or data exfiltration channels.

5) Temporal Features: Temporal features quantify dynamic behavioral shifts by measuring variations in node characteristics across consecutive windows. To capture structural change, the degree change rate calculates the relative connectivity difference between windows. Similarly, bytes and centrality change rates track proportional fluctuations in data transfer and positional importance over time. Furthermore, a binary new node indicator flags entities appearing for the first time. Collectively, these metrics provide vital context for identifying emerging threats, abrupt deviations, and multi-stage attack progressions.

6) Graph-Level Features: Graph-level features capture the overall network environment and are uniformly assigned to all nodes within a temporal window. Graph density evaluates overall connectivity, while modularity and assortativity quantify community strength and the interaction of similarly-degreed nodes. Additionally, reciprocity captures bidirectional communication, alongside metrics like graph diameter and connected component sizes providing broad topological context. Ultimately, fusing these global properties with local, traffic, community, and temporal dynamics into a robust feature matrix empowers the model to learn complex, coordinated anomalous behaviors that would remain hidden in isolation.

E. Machine Learning Module

For multi-class anomaly classification, the framework employs XGBoost [13] to capture complex, non-linear feature interactions within high-dimensional spaces. By integrating sequential tree boosting with L1/L2 regularization and native missing-value handling, the algorithm inherently mitigates overfitting on highly imbalanced security datasets. Crucially, the model outputs both discrete predictions and calibrated probabilities, quantifying the uncertainty necessary for prioritized incident response. Overall performance is evaluated using standard metrics (Precision, Recall, F1-score) and detailed per-class analyses, while gain-based feature importance and visual diagnostics rigorously validate the engineering pipeline and ensure operational transparency.

F. IP Risk Analysis Module

To prioritize security investigations, IP risk analysis translates raw model predictions into actionable, entity-level assessments. For each IP, a dynamic risk score is computed by multiplying the model's prediction confidence by a domain-specific severity weight, which reflects the inherent operational impact of specific anomalies such as Command & Control communications. To capture temporal threat dynamics, these scores are aggregated across successive sliding windows and categorized into discrete severity tiers ranging from minimal to critical. Ultimately, this structured risk layer supported by visual analytics such as temporal heatmaps and distribution summaries which empowers security analysts to rapidly isolate high-priority threats, filter out benign noise, and allocate incident response resources efficiently.

IV. Experimental Results

This section evaluates the proposed multi-source graph-based anomaly detection framework within a realistic banking network environment. Using a strictly reproducible experimental protocol, the evaluation focuses on three main objectives: (i) assessing the impact of integrating heterogeneous security logs into a unified graph; (ii) quantifying the effectiveness of structural, community-aware, and temporal features; and (iii) validating the operational utility of translating model predictions into interpretable IP-level risk scores. The subsequent subsections detail the dataset, preprocessing process, and the evaluation setup.

A. Dataset

The evaluation utilizes real-world heterogeneous security logs from a banking network. These logs are integrated into a unified representation prior to graph construction and model training.

1) Data Sources: The dataset fuses three primary security sources:

- Vectra Alarms: Network-based threat detections (e.g., lateral movement, C2 activity) detailing IPs, timestamps, and severity scores.
- QRadar Offenses: SIEM-correlated incidents that aggregate low-level events to provide contextual threat intelligence.
- CrowdStrike Detections: Endpoint Detection and Response (EDR) alarms capturing host-level anomalies, malware, and behavioral indicators.

Despite differing native schemas, all records are normalized into a standard format containing essential attributes (timestamp, source/destination IPs, event type, and severity/confidence). This normalization enables the seamless integration of network, SIEM, and endpoint telemetry into a single graph-based framework.

2) Preprocessing and Labeling: Prior to graph construction, raw event logs are processed through a unified pipeline. Source-specific parsers map diverse datasets into a common schema, resolve inconsistencies, standardize anomaly labels, and synchronize timestamps for accurate chronological ordering.

The integrated stream is segmented using a sliding window approach (e.g., 500 events per window with a 250-event overlap) to balance temporal resolution and statistical robustness. Ground-truth labels are assigned at the node (IP) level within each window. For IPs exhibiting multiple anomalies, a domain-specific severity hierarchy determines the dominant label. Inactive IPs are excluded from the window's graph, whereas those with strictly benign activity are classified as normal.

Ultimately, this procedure yields a labeled dataset of node-window instances spanning numerous fine-grained anomaly types (e.g., 31 classes). This accurately reflects the highly imbalanced, multi-class nature of real-world security monitoring.

B. Experimental Setup

To ensure a fair and reproducible evaluation against baseline methods, all experiments were conducted on a multi-core CPU workstation using standard Python machine learning and graph processing libraries. Random seeds were strictly fixed across all stages from data splitting to model evaluation to eliminate stochastic variability.

To simulate realistic deployment and prevent data leakage, a chronological train-test split was employed, allocating the initial 70% of temporal windows for training and the remaining 30% for testing. Model selection and hyperparameter tuning were conducted exclusively within the training partition using internal temporal validation. Furthermore, all preprocessing steps including feature scaling, label encoding, and low-variance feature removal were fitted solely on the training set and subsequently applied to the test data. This strict isolation ensures an unbiased performance evaluation on genuinely unseen events.

C. Simulation Results

Figure 2 presents the micro-averaged precision-recall (PR) curve, summarizing the model's global discriminative performance across a highly imbalanced, multi-class label space. By aggregating one-vs-rest binary decisions, the micro-averaged Average Precision (AP) (0.98) demonstrates the framework's ability to recover a substantial proportion of

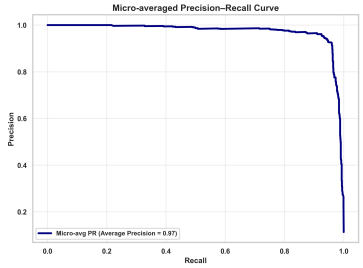

Fig. 2. The precision-recall curve for the classification results.

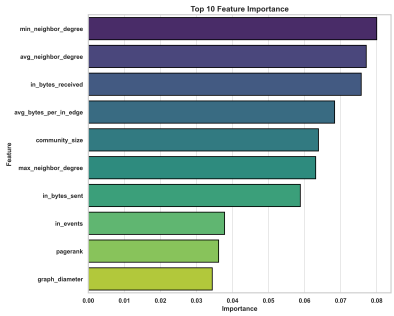

Fig. 3. The feature importance scores obtained using the classification module.

true anomalies including rare, security-critical events while maintaining a low false-positive rate.

This global trade-off is particularly suited for the evaluated banking environment, as it effectively balances comprehensive threat coverage with the operational necessity of minimizing false alarms for security analysts.

Figure 3 illustrates the top 10 features ranked by XGBoost's gain-based importance. The skewed distribution confirms that a concise subset of attributes drives the majority of the model's discriminative power for anomaly classification.

The analysis reveals a critical insight: local neighborhood connectivity and traffic characteristics are significantly more predictive than traditional centrality measures. The highest-ranking predictors are dominated by neighborhood structural attributes such as minimum neighbor degree (0.080), average neighbor degree (0.076), and maximum neighbor degree (0.061) which capture the immediate topology around an IP. These are closely complemented by some traffic metrics. This combination suggests that anomalies are best identified through irregular local connection and data transfer behaviors, distinguishing malicious activities from legitimate hosts.

Furthermore, community size (0.062) ranks fifth, highlighting the discriminative value of an IP's broader community context. While global structural properties like graph diameter (0.038) and community size (0.062) appear in the top 10, traditional centrality measures (e.g., betweenness, closeness) are notably absent. This empirically validates the extraction of rich neighborhood and traffic features, demonstrating that understanding local interaction patterns is far more effective than global positional metrics for detecting enterprise network anomalies.

To evaluate the effectiveness of the proposed multi-source graph framework, its classification performance is compared against traditional Static Blacklisting and a baseline Support Vector Machine (SVM) model. As detailed in Table I, the proposed approach significantly outperforms both baselines across all key performance metrics.

TABLE I
THE PERFORMANCE COMPARISON OF M-GRAD AGAINST BASELINE METHODS.

	Acc.	**Prec.**	**Rec.**	**F-1**
Static Blacklisting	0.824	0.814	0.829	0.822
SVM	0.912	0.911	0.906	0.909
M-GRAD	0.985	0.972	0.978	0.975

Static Blacklisting achieved the lowest overall performance, with an accuracy of 0.824 and an F1-score of 0.822. Its relatively low precision 0.814 highlights a major operational drawback: a high rate of false positives, which inevitably leads to alarm fatigue for security analysts. The SVM model demonstrated notable improvements, pushing accuracy to 0.912 and the F1-score to 0.909. While this confirms that machine learning algorithms can better capture complex threat patterns than static rules, it still falls short in highly imbalanced and heterogeneous network environments. The proposed XGBoost-driven graph framework achieved an exceptional accuracy of 0.992 and an F1-score of 0.979. Crucially, the model maintained a high precision of 0.981 alongside a recall of 0.975. This balanced performance proves that fusing structural, community, and temporal graph features enables the system to reliably identify sophisticated anomalies while minimizing false alarms.

Ultimately, this comparative analysis validates that moving beyond isolated log analysis to a unified, community-aware graph representation dramatically enhances both detection accuracy and operational efficiency.

The event-type risk analysis in Figure 4 summarizes how the proposed risk scoring mechanism behaves across different anomaly categories. The left panel ranks event types by their average risk score, with error bars indicating the standard deviation, and reveals that a small subset of anomaly classes consistently receives high risk scores, reflecting their severe security impact and strong association with high-confidence model predictions. In contrast, several more frequent but operationally less critical classes exhibit moderate average risk with lower variability, suggesting that the framework successfully differentiates between truly high-impact attack behaviors and more benign or noisy patterns. The right panel complements this view by displaying the number of IP instances assigned to each event type, highlighting which anomaly categories dominate the alarm volume. Taken together, the two panels show that the system concentrates high risk on a limited set of critical event types while maintaining broader coverage over more common classes, thereby supporting both effective incident prioritization and comprehensive monitoring.

Figure 5 illustrates the evolution of IP-level risk across successive time windows, contrasting the average network

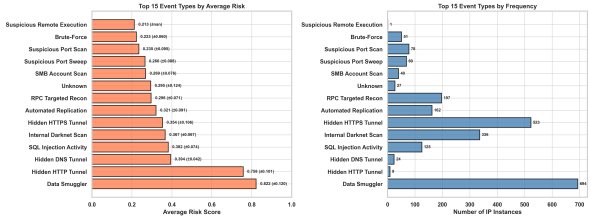

Fig. 4. The analysis of event types vs average risk score and number of IP instances.

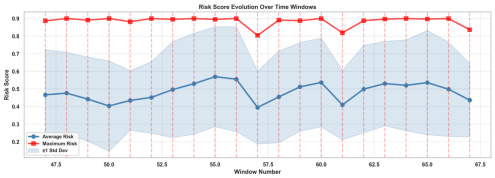

Fig. 5. The change of risk score over time windows.

threat level with the maximum risk score to identify the most critical asset per segment. The variance in risk, represented by a ± 1 standard deviation, highlights the dispersion of threat levels among IPs. A narrow variance indicates homogeneous network behavior, whereas significant heterogeneity reveals localized subsets of IPs experiencing markedly elevated risk. By flagging time periods where the maximum risk exceeds a predefined threshold (e.g., 0.7), the analysis directly isolates windows requiring immediate forensic investigation. Ultimately, capturing both long-term threat trends and short-lived bursts of extreme risk supports the temporal correlation of incidents, enabling the identification of complex attack phases that unfold over multiple windows.

Figure 6 illustrates the temporal evolution of IP-level risk for the top 20 most critical ones, selected based on their maximum observed risk scores. By mapping risk scores across the sliding-window segments, the visualization effectively distinguishes between persistent, chronic high-risk behavior and short-lived anomalous bursts. Furthermore, the analysis reveals distinct temporal clustering of high-risk activity. Simultaneous risk elevations across multiple IPs strongly suggest coordinated operations or broader attack phases, whereas isolated risk spikes indicate localized, targeted anomalies. By filtering out benign background traffic and focusing exclusively on the top 20 security-relevant nodes, this approach bridges the gap between aggregate metrics and operational reality. Ultimately, mapping these temporal correlations give power to security teams to identify repeatedly targeted resources, track evolving attack phases, and prioritize incident response efforts.

V. Conclusion & Future Work

This paper introduced a novel multi-source, graph-based anomaly detection framework to overcome siloed security monitoring by seamlessly fusing heterogeneous data from Vectra, QRadar, and CrowdStrike into a unified temporal graph. Empirically, local neighborhood characteristics and traffic dynamics proved significantly more effective than traditional global centrality measures in capturing coordinated attack patterns. Building on these localized insights, the sliding-window architecture successfully captured both short-lived malicious bursts and the gradual progression of multi-stage attacks. Ultimately, translating these structural and temporal predictions into a dynamic, IP-level risk-scoring mechanism provides prioritized alarms that directly reduce analyst cognitive load and alarm fatigue.

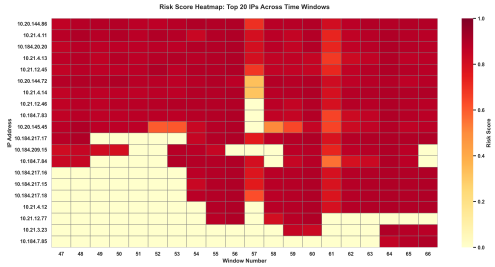

Fig. 6. The heatmap of risk scores for top 20 IP addresses.

Despite its operational robustness, computing complex global metrics (e.g., betweenness centrality) presents scalability challenges, and fixed sliding-window parameters require environment-specific tuning. Future work will explore dynamic window sizing, Graph Neural Networks (GNNs) for automated spatial feature extraction, and broader empirical validation across diverse industrial architectures.

References

[1] M. Vielberth, F. Böhm, I. Fichtinger, and G. Pernul, "Security operations center: A systematic study and open challenges," *IEEE Access*, vol. 8, pp. 227 756–227 779, 2020.

[2] Z. Jia *et al.*, "Big-data analysis of multi-source logs for anomaly detection on network-based system," in *2017 13th IEEE conference on automation science and engineering (CASE)*. IEEE, 2017, pp. 1136–1141.

[3] P. Xu, Q. Gao, Z. Zhang, and K. Zhao, "Multi-source data based anomaly detection through temporal and spatial characteristics," *Expert Systems with Applications*, vol. 237, p. 121675, 2024.

[4] H. Kim, B. S. Lee, W.-Y. Shin, and S. Lim, "Graph anomaly detection with graph neural networks: Current status and challenges," *IEEE Access*, vol. 10, pp. 111 820–111 829, 2022.

[5] A. Al Siam *et al.*, "IP safeguard-an AI-driven malicious IP detection framework," *IEEE Access*, 2025.

[6] X. Li and J. Zhang, "Multi-source data fusion for real-time cybersecurity situational awareness and visualization," *Journal of Cyber Security and Mobility*, vol. 14, no. 4, pp. 955–980, 2025.

[7] P. Appiahene *et al.*, "Network intrusion detection using a hybrid graph-based convolutional network and transformer architecture," *PloS one*, vol. 21, no. 1, p. e0340997, 2026.

[8] F. Al Tfaily *et al.*, "Graph-based federated learning approach for intrusion detection in IoT networks," *Scientific Reports*, vol. 15, no. 1, p. 41264, 2025.

[9] A. Ayoub *et al.*, "Detecting fraudulent communities in financial networks using hybrid classification and ranking approaches," *Advances in Artificial Intelligence and Machine Learning*, p. 2, 2026.

[10] J. Li and D. Yang, "Research on financial fraud detection models integrating multiple relational graphs," *Systems*, vol. 11, no. 11, p. 539, 2023.

[11] P. De Meo, E. Ferrara, G. Fiumara, and A. Provetti, "Generalized louvain method for community detection in large networks," in *2011 11th international conference on intelligent systems design and applications*. IEEE, 2011, pp. 88–93.

[12] L. Katz, "A new status index derived from sociometric analysis," *Psychometrika*, vol. 18, no. 1, pp. 39–43, 1953.

[13] T. Chen *et al.*, "XGBoost: extreme gradient boosting," *R package version 0.4-2*, vol. 1, no. 4, pp. 1–4, 2015.