

Article

SentinelGraph: Temporal Graph Reasoning for Sender Group Attribution in Honeypot Traffic

Shiyu Wang , Cheng Tu , Min Zhang * and Pengfei Xue 

College of Electronic Engineering, National University of Defense Technology, Hefei 230037, China; wangshiyu@nudt.edu.cn (S.W.); tucheng@nudt.edu.cn (C.T.)

* Correspondence: zhangmindy@nudt.edu.cn (M.Z.); xuepengfei@nudt.edu.cn (P.X.)

Abstract

Hosts generating unsolicited network traffic increasingly operate in a coordinated manner rather than in isolation. Scanning and exploitation activities are often distributed across multiple hosts that share common infrastructure, toolchains, and behavioral patterns, forming loosely coupled yet persistently aligned sender groups. Accurately attributing such groups is critical for understanding organized activities and strengthening network defense capabilities. However, existing attribution approaches face notable limitations. Methods that rely on threat intelligence suffer from delayed updates and limited coverage. Static feature-based approaches ignore temporal ordering and therefore fail to capture multi-stage behavioral evolution. Although dynamic sequence models incorporate temporal patterns, they typically overlook the collaborative structural relationships among coordinated senders. In this paper, we propose SentinelGraph, a temporal graph reasoning framework for sender group attribution from honeypot traffic. SentinelGraph constructs a temporal knowledge graph and integrates a recurrent graph evolution module to jointly model coordination structures and their temporal dynamics. A structure enhancement module further exploits contextual information available at the target time, while an auxiliary relation loss encourages the learning of enriched entity representations. This design enables accurate attribution even for previously unseen senders by leveraging information from their observed neighbors. Experiments on real-world honeypot data demonstrate that SentinelGraph substantially outperforms state-of-the-art methods in modeling coordinated network behaviors.

Keywords: sender group attribution; cybersecurity; honeypot traffic; temporal graphs



Academic Editor: Aryya Gangopadhyay

Received: 20 January 2026

Revised: 11 February 2026

Accepted: 13 February 2026

Published: 14 February 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

With the rapid development of the Internet, unsolicited network behavior increasingly exhibits multi-host coordination and temporal evolution rather than isolated actions [1,2]. While this enhances network information dissemination efficiency, it also amplifies the sophistication and destructiveness of organized network attacks [3,4]. For example, in modern cyberattacks, scanning and exploitation tasks are distributed across multiple hosts that share infrastructure, tools, and behavioral strategies, forming “collaborative behavior groups” [5,6].

Given the significance of collaborative behavior in network activities, analyzing such behavior requires grouping hosts by the coordinating entities behind them [7,8]. To this end, sender groups represent collections of IP addresses operated by the same entity, where a sender refers to an IP address that initiates unsolicited network interactions, whether

cyberspace search engines (such as Censys [9], Shodan [10]), autonomous botnets [1], or unknown collaborative activities. Unlike individual-level analysis, sender group attribution enables a holistic understanding of the organizational structure and evolutionary characteristics of collaborative behavior, providing a more hierarchical perspective for analysis [3,6]. First, by focusing on group behavioral patterns, advanced persistent threats and organized attack campaigns can be discovered earlier and more accurately [2,7], even in the presence of complex encryption algorithms [11]. Second, when any member of the group exhibits anomalous or malicious behavior, the overall risk attributes of the entire group can be quickly inferred, enabling more efficient threat attribution and response [5,8]. Third, group-oriented analysis not only applies to attack detection but also facilitates the identification of substantial non-essential benign traffic within organizations, thereby improving network and server resource utilization and enhancing overall system efficiency [6]. These capabilities make sender group attribution of paramount importance for network security.

To support sender group attribution in unsolicited traffic, prior studies leverage trap-based monitoring infrastructures, such as network telescopes and honeypots, to collect large-scale Internet traffic and infer relationships among senders. Existing attribution methods associate individual senders with coordinated groups by exploiting threat intelligence [12–14], static traffic features [15–17], or temporal patterns [18–21]. While these approaches provide valuable insights from different perspectives, each tends to capture only a partial view of coordinated sender behavior, introducing attribution uncertainty when different groups perform similar actions or when behavioral patterns drift over time [7]. Sephreat intelligence-based approaches rely on external knowledge sources whose coverage and timeliness are often limited, making them unsuitable for attributing senders in emerging or previously undocumented campaigns. To overcome this limitation, static feature-based methods learn attribution patterns directly from observed traffic by aggregating activity features within fixed time windows. However, by treating each window independently, they struggle to capture the temporal evolution of coordination behaviors, where different groups exhibit distinct adaptation patterns that serve as key discriminative signatures [22]. Dynamic sequence-based methods address this gap by modeling temporal dynamics through linear sequences or sliding-window co-occurrences. Yet these methods still face a fundamental limitation: by modeling sender activities independently or through pairwise co-occurrences, they cannot assess whether multiple senders jointly participate in a shared coordination structure that may evolve over time, in which member hosts adapt their actions based on feedback from others' observed outcomes [23–25]. As a result, attribution decisions rely on local co-occurrence signals, leading to unstable group attribution that may both conflate coincidental traffic overlaps and fragment genuinely coordinated sender groups.

Unlike existing studies, we aim to model higher-order coordination structures through which sender groups operate and evolve over time, thereby improving sender group attribution performance on honeypot traffic. Achieving this goal involves three key challenges. First, constructing a graph representation from large-scale honeypot traffic must preserve essential coordination signals while remaining computationally tractable. Encoding all traffic details naively results in excessively large and noisy graphs, whereas oversimplification loses critical relational patterns necessary to distinguish coordinated groups from incidental overlaps. Second, accurate attribution requires jointly modeling both the structural organization of sender groups and their temporal evolution, as coordination patterns emerge and change over time, which cannot be fully captured by pairwise or independent modeling. Third, practical attribution systems must generalize to previously unseen senders in temporal attribution scenarios, inferring group membership from emerging relational and behavioral contexts rather than relying solely on historical observations.

To tackle these challenges, we propose SentinelGraph, a sender attribution framework based on temporal graph reasoning. Our key insight is that coordinated sender behaviors are naturally expressed as evolving graph-structured relations, where coordination manifests as synchronized adaptation patterns in how group members jointly evolve, rather than in their absolute behavioral features. Specifically, SentinelGraph: (1) constructs a session-level temporal knowledge graph from honeypot traffic, with nodes representing senders, services, and ports, and edges capturing observed interactions; (2) explicitly models the co-evolution of historical coordination structures and temporal dynamics through a Recurrent Graph Evolution module; (3) leverages the Structure Enhancement module to incorporate neighborhood information at the target time, separating historical and current embeddings to support accurate attribution for new entities; and (4) employs an auxiliary relation modeling loss to regularize entity embeddings, improving attribution accuracy.

This paper makes the following contributions:

- We present SentinelGraph, a temporal graph reasoning framework for sender group attribution. By integrating honeypot traffic knowledge graphs, historical behavioral evolution, and structural dependency modeling, we achieve systematic analysis of coordinated network behavior. Experiments on real honeypot traffic data demonstrate superior performance compared to state-of-the-art methods, validating the effectiveness of our approach in modeling coordinated behavior.
- We construct the honeypot traffic graph at the session level, which preserves essential coordination information while remaining computationally efficient, enabling effective modeling of multi-entity interactions without incurring excessive graph size or noise, leading to more reliable sender group attribution (achieving an MRR improvement of 0.139 over flow-level approaches while providing a $2.8\times$ speedup).
- We propose a representation method that jointly fuses behavioral coordination structures and temporal evolution to characterize the temporal knowledge graph, thereby enabling the proposed method to comprehensively capture structural and temporal dependencies and further enhance SentinelGraph's attribution performance (removing temporal evolution modeling or relational structural modeling results in an absolute MRR drop of 0.026–0.206).
- We introduce a structure enhancement module that incorporates neighborhood information at the target time, separating historical and current embeddings to enable accurate attribution for newly observed entities, improving identification of new senders in practical temporal attribution scenarios (83% of attribution failures without this module correspond to new senders).

2. Background and Related Work

2.1. Sender Group and Cybersecurity

In large-scale Internet environments, network activities such as scanning, probing, and exploitation are often conducted in a coordinated manner by multiple hosts under shared operational control, which may stem from a common organization or a shared automation framework. We refer to such collections of hosts, identified by IP addresses, as *sender groups*. For instance, cyberspace search engine like Censys [9] and Shodan [10] each control multiple distributed scanner hosts to systematically probe Internet-wide services [26], while malicious botnets [1] control geographically dispersed compromised hosts to coordinate exploitation campaigns. Within each sender group, shared operational constraints lead hosts under common management to employ the same toolchains, follow similar task allocation strategies, and reuse infrastructure, collectively producing observable behavioral homogeneity in network traffic. At the same time, different sender groups implement these operational practices in distinct ways, resulting in differentiable behavioral patterns across

organizations. Specifically, while Censys, Shodan, and botnets all represent sender groups operating distributed hosts, their behavioral manifestations differ substantially: Censys exhibits broader coverage across services, Shodan demonstrates preference for specific ports and protocols [27], and different botnets produce distinct attack sequences and targeting patterns characteristic of their respective command-and-control frameworks [1].

Such shared control is reflected in network traffic through structured regularities along two complementary dimensions. First, sender groups engage in multi-stage activity workflows to maintain an up-to-date understanding of the network environment. Rather than performing a single pass of reconnaissance or probing, automated scanning and exploitation infrastructures organize these stages in a coordinated manner over time [22,28]. Although concrete operational choices may adapt to changes in the network environment, the organization of activity stages and their temporal evolution remain consistent, resulting in coherent phase-aligned temporal patterns in the observed traffic [25]. Second, coordinated sender group behavior emerges from feedback among member hosts. Outcomes observed by one host, such as successful responses, failures, or service-specific characteristics, can influence subsequent actions taken by other hosts, leading to adaptive task reassignment and coordinated execution across the group [23–25]. As a result, the traffic generated by multiple hosts reflects not only parallel execution but also interdependent, structured coordination that reveals the underlying operational logic of the sender group. These properties indicate that sender group activities are inherently iterative and feedback-driven, necessitating analysis beyond individual hosts to capture collective and temporally evolving behaviors.

2.2. Methods for Sender Group Attribution

Sender group attribution focuses on associating observed Internet traffic with coordinated sender groups, using large-scale measurement data such as darknet and honeypot traffic. Two primary trap-based monitoring approaches offer complementary strengths for this purpose. Darknets (network telescopes) passively capture traffic directed at unused IP space, offering broad visibility into global scanning patterns [29,30], but their passive nature limits interaction depth. In contrast, honeypots actively engage with scanners by mimicking real devices, capturing richer behavioral information through multi-round interactions, including payload content (e.g., commands and code, User-Agent variations, protocol-specific fields) and session-level temporal patterns [13,17,31,32].

With traffic captured, researchers have explored various methods to attribute senders to groups. We review representative work from three main categories: leveraging external threat intelligence, extracting static behavioral features, and modeling dynamic behavioral sequences.

Threat Intelligence-Based Approaches. These methods leverage external knowledge to attribute observed traffic to sender groups. On one hand, Internet measurement studies commonly employ publicly available databases and threat intelligence feeds to identify specific senders. These databases maintain mappings between IP addresses and organizations or known malicious entities, enabling direct attribution of scanning traffic. Bennett et al. [12] conducted an empirical study of Censys and Shodan scanning behaviors, identifying their scanner IP addresses through DNS history searches and reverse DNS lookups, thereby attributing observed darknet traffic to these device search engine platforms. On the other hand, threat intelligence can be incorporated as features in machine learning clustering algorithms. Wu et al. [14] rely on manually curated threat intelligence to map frequent commands and their abstract syntax tree (AST) skeletons in IoT honeypot payloads to MITRE ATT&CK techniques, constructing TTP vectors that are subsequently used for hierarchical clustering of IP addresses. Crochelet et al. [13] enrich one-hour IP

aggregates with a one-shot VirusTotal history lookup, then combine this threat-intelligence-derived feature with command counts, protocol types and scan volumes to compute five-dimensional scores for sender-group clustering. Hobert et al. [20] map attacker-typed shell commands to MITRE ATT&CK techniques, encode them as ATT&CK TTP bit-vectors, and use cosine similarity to group 793 command traces into eight recurrent sender clusters without relying on IPs or hashes.

However, for large-scale sender group attribution, these methods face fundamental limitations. First, coverage limitations persist as the vast majority of scanning IPs lack documented ownership or organizational affiliation in public databases, with approximately two-thirds of senders in darknet traffic belonging to unknown categories [19]. Second, intelligence staleness renders them ineffective against emerging groups and zero-day campaigns that lack historical records, often requiring weeks or months before new threat patterns are documented. These limitations motivate methods that can automatically learn behavioral patterns from observed traffic data without relying on external threat intelligence as features.

Static Feature-Based Approaches. These methods extract aggregate statistics from fixed time windows, such as packet counts, average payload lengths, and protocol distributions, and identify sender groups through pattern matching. Kallitsis et al. [15] distill daily Darknet traffic into per-scanner vectors summarizing volume, port-set, TTL patterns, etc., embed these vectors with a deep autoencoder, group scanners into purely behavioral clusters without any external label. Shamsi et al. [16] extract behavioral features such as credentials, commands, outbound activity, timing, and sessions, and apply independent clustering (e.g., OPTICS, spectral clustering, and DTW) followed by consensus clustering to identify IPs likely controlled by the same attacker or organization. Scanner-Hunter [17] proposed a Self-Expanding Multi-Class Classification (SEMCC) model with hierarchical feature extraction, utilizing static features including sequence numbers and response times combined with IP prefix judgment to identify known and unknown scanner groups.

However, static approaches collapse temporal information into a single snapshot, thereby overlooking event ordering and limiting their ability to capture multi-stage behavioral evolution as well as temporal co-occurrence patterns among senders. In addition, the assumption of consistent tool usage within groups may reduce robustness in the presence of fingerprint obfuscation or tool variation across heterogeneous networks.

Dynamic Sequence-Based Approaches. To capture behavioral evolution and temporal patterns, recent work models sender activities as sequences. Ring et al. [18] introduced IP2Vec, adapting Word2Vec to learn similarities between IP addresses by treating network flows as context information. By extracting co-occurrence patterns from flow data, IP2Vec successfully clustered botnet IP addresses, demonstrating that IPs appearing in similar network contexts belong to related groups. DarkVec [19] builds per-service, time-windowed sequences of sender IPs and trains Word2Vec on their co-occurrence within sliding context windows, dynamically clustering sender populations that jointly appear in coordinated scans. Sachidananda et al. [21] cluster enterprise IDS/EDR alerts by timestamp and source IP to piece together kill-chain fragments, employ an LSTM to infer the missing stages, and supply the completed technique sequence to a neural classifier that outputs the most probable APT group. While sequence-based methods effectively capture local temporal co-occurrence, they abstract sender activities into linear sequences or sliding windows, overlooking higher-order interaction patterns among multiple senders. This representation makes it difficult to model the formation, evolution, and dissolution of sender groups over time, which are more naturally expressed as dynamic graph structures.

Overall, existing methods have not jointly modeled the scalability, temporal dynamics, and structural characteristics essential for comprehensive sender group attribution.

2.3. Temporal Graphs and Cybersecurity

Temporal Graph Reasoning builds on Temporal Graph Neural Networks (TGNNs), which extend message-passing models with time-aware aggregation and node memory to retain historical dependencies in dynamic graphs [33]. Continuous-time dynamic graphs align well with event-driven traffic, and representative architectures include TGAT [34], spatiotemporal GNNs integrating RNN or attention with graph convolutions [35], and lightweight models such as GraphMixer [36]. In cybersecurity, temporal graph learning has been applied to anomaly detection [37], federated intrusion detection [38], APT detection via causal alert analysis [39], and threat investigation [40].

Building on these capabilities, temporal graph representations are particularly well-suited for sender group attribution. As described in Section 2.1, this task inherently involves both structural connectivity, which reflects how multiple senders are related through shared targets or interaction patterns, and temporal dynamics, as coordinated activities often unfold across multiple stages over time. By modeling time-evolving interactions among entities as dynamic graphs, and leveraging time-aware message passing and historical context aggregation, TGNNs provide a natural and general analytical framework for reasoning about coordinated behaviors in large-scale, event-driven network environments.

3. Method

In this section, we present SentinelGraph, a knowledge graph reasoning framework for sender group attribution. The framework is designed to model coordinated sender behaviors that emerge through structured interactions over time, and to support accurate attribution under evolving traffic patterns. As illustrated in Figure 1, SentinelGraph consists of four components: a temporal knowledge graph construction module, a recurrent graph evolution module, a structure enhancement module, and a sender group attribution module.

To preserve essential coordination signals from large-scale honeypot traffic while maintaining computational efficiency, i.e., the graph construction challenge mentioned in Section 1, the temporal knowledge graph construction module aggregates traffic into a session-level temporal knowledge graph that encodes interactions among senders, services, and ports. This structured representation provides a clean and tractable substrate for subsequent reasoning.

To jointly capture the organizational structure of sender groups and their temporal evolution, i.e., the structural-temporal evolution challenge mentioned in Section 1, the recurrent graph evolution module learns entity embeddings in two complementary ways: (i) coordination pattern representation, capturing relational dependencies within each time slice to summarize group-level coordination patterns, and (ii) behavioral evolution representation, modeling dependencies across time slices to track how group behaviors evolve over time.

To attribute previously unseen senders based on emerging context without relying solely on historical observations, i.e., the new-sender generalization challenge mentioned in Section 1, the structure enhancement module refines embeddings at the target time by propagating information from neighboring entities, aggregating it across the local neighborhood, and selectively fusing historical and current signals. This allows the model to leverage current context for new senders while preserving long-term patterns learned from previously observed entities.

Finally, the sender group attribution module predicts the organizational affiliation of each sender based on the refined embeddings, while an auxiliary relation modeling loss regularizes these embeddings to reinforce structurally consistent coordination patterns. In this way, SentinelGraph integrates session-level interaction modeling, historical evolution, and

target-time context in a unified reasoning workflow, systematically capturing coordinated behaviors and enabling accurate sender group attribution from honeypot traffic.

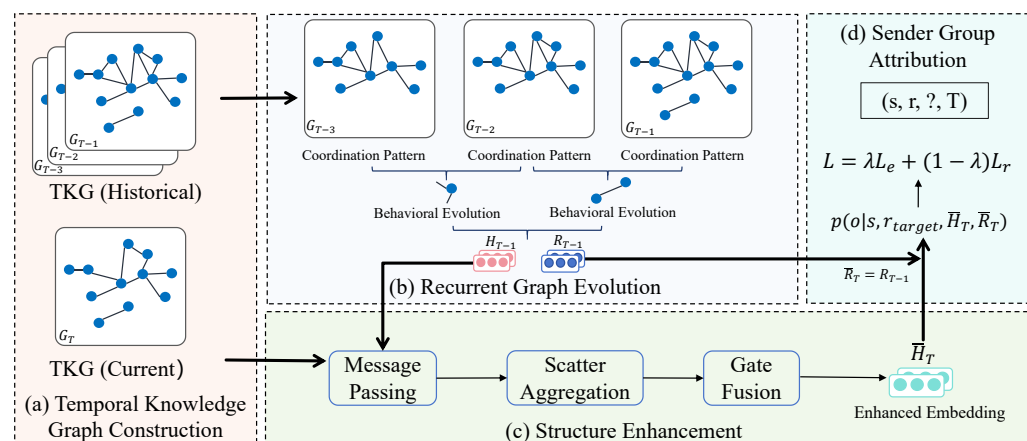


Figure 1. High-level workflow diagram of SentinelGraph. Starting from historical and current TKGs (a), the Recurrent Graph Evolution module (b) processes historical graph snapshots to learn coordination patterns and behavioral evolution. The Structure Enhancement module (c) then operates on current observable facts to enable attribution for new senders. Finally, Sender Group Attribution (d) performs probabilistic inference to predict organizational affiliations. Different colors denote different embedding states (H_{T-1} , R_{T-1} , and $\bar{H}_T$). The symbol “?” indicates the sender group entity to be predicted.

3.1. Temporal Knowledge Graph Construction

Our sender traffic originates from globally distributed multi-service honeypots that automatically convert captured network activities into flow-based log records, including IP addresses, ports, protocols, session IDs, and timestamps (Figure 2, left). Sender group attribution relies on multiple dimensions of coordinated activity that collectively characterize organizational or automated behavior: *who* interacted with *whom* reveals coordination patterns among distributed IPs, *what* services were targeted indicates operational focus, *when* activities occurred exposes synchronized campaigns, and *how* interactions proceeded distinguishes reconnaissance depth. Traditional feature-based approaches [15–17] flatten these dimensions into independent attributes (e.g., [source_IP, port_list, packet_count, timestamp]), losing the relational structure that encodes organizational fingerprints. We instead model traffic as a temporal knowledge graph where typed edges explicitly preserve semantic relationships, enabling joint reasoning over complementary attribution signals.

To meaningfully capture temporal coordination, we represent interactions using discrete graph snapshots rather than operating directly on raw event-level timestamps. Although honeypot logs provide millisecond-level temporal precision, interactions observed at this granularity are extremely sparse. Only a limited number of events co-occur at identical timestamps, which leads to fragmented graph structures and inefficient use of relational information [41]. As a consequence, coordination signals that depend on concurrency are weakened rather than enhanced. By aggregating interactions within broader temporal windows, discrete snapshots consolidate near-synchronous activities [41,42]. For example, multiple IPs associated with the same sender group may target similar services within minutes or hours of one another. These aggregated interactions form coherent graph structures that reflect organizational behavior. At the same time, temporal evolution is preserved, since variations across snapshots naturally capture the dynamics of sender group activity.

Accordingly, we model sender group attribution as a temporal knowledge graph (TKG) reasoning problem to explicitly encode these attribution dimensions in a unified

representation (Figure 2, right). Let $\mathcal{G} = \{G_1, G_2, \dots, G_T\}$ denote a sequence of graph snapshots over discrete timestamps. Each snapshot $G_t = (\mathcal{V}, \mathcal{R}, \mathcal{E}_t)$ is a directed, multi-relational graph, where $\mathcal{V}$ is the set of entities (including source hosts, destination hosts, and ports), $\mathcal{R}$ is the set of relation types (e.g., session-type relations, and port/protocol relations), and $\mathcal{E}_t$ is the set of observed facts at timestamp t . Each fact is represented as a quadruple (s, r, o, t) , where $s, o \in \mathcal{V}$ and $r \in \mathcal{R}$. We design relation types to encode complementary attribution signals, including service, protocol preferences, and port exposure, among others. In doing so, a critical design choice is determining the appropriate granularity at which network interactions are represented as graph edges. We adopt session-level aggregation rather than finer-grained representations such as flow-level or packet-level modeling for three reasons. First, honeypot traffic typically exhibits rich multi-interaction patterns within sessions, allowing session-level representations to capture complete organizational workflows and coordinated behavior beyond shallow probing interactions. Second, due to the prevalence of encrypted traffic in modern honeypot captures, attribution signals must be inferred from interaction structure rather than packet-level features, and session aggregation provides sufficient contextual information to distinguish surface-level probing from persistent reconnaissance. Third, aggregated interaction modeling substantially reduces graph size and computational overhead in large-scale honeypot deployments while preserving attribution-relevant behavioral semantics [43]. Accordingly, interaction relations in our graph are distinguished by session depth (Figure 2, right), defined as the number of flow records within a session: sessions containing more than k flows are labeled as *LongSessionTo*, while others are *ShortSessionTo*. This design leverages the session structure provided by session IDs in our honeypot logs, preserving attribution-relevant semantics related to reconnaissance depth and persistence.

Under this formulation, sender group attribution reduces to inferring the unique organizational affiliation of each source host from its observed interactions. While this formulation outputs a probability distribution over candidate organizations, each source host is associated with a single ground-truth organization, and attribution is determined by the top-ranked prediction. Specifically, the objective is to predict the missing object in the *BelongsToOrg* relation. Formally, let $r_{\text{target}} = \text{BelongsToOrg}$ denote the target relation. Given a query $(s, r_{\text{target}}, ?, T)$ for source host s at timestamp T , along with historical graph snapshots $G_{1:T-1}$ and observable interaction facts G_T at timestamp T , the task is to estimate the conditional probability over all candidate organizations:

$$p(o \mid s, r_{\text{target}}, G_{1:T-1}, G_T), \quad o \in \mathcal{V}_{\text{org}},$$

where $\mathcal{V}_{\text{org}} \subset \mathcal{V}$ denotes the set of organization entities.

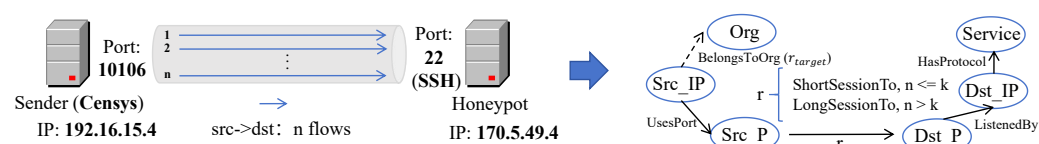


Figure 2. Temporal knowledge graph construction from honeypot traffic. Right: Temporal knowledge graph with nodes representing source/destination hosts, ports, protocols, and organizations. Solid edges encode observed relationships (*UsesPort*, *HasProtocol*, *ListenedBy*, *LongSessionTo*, *ShortSessionTo*); dashed edges (*BelongsToOrg*) represent target organizational affiliations to be inferred.

3.2. Recurrent Graph Evolution

Sender group attribution is not determined by isolated or independently modeled interactions, but by recurring patterns that emerge from how multiple senders behave collectively over time. While the temporal knowledge graph is explicitly constructed to

preserve multiple dimensions of coordinated activity across time, effective attribution ultimately depends on how these dimensions interact and persist across observations. This involves two tightly coupled aspects. First, sender groups exhibit consistent coordination behaviors. Senders belonging to the same group tend to interact with similar sets of services or targets within comparable time windows, and such coordination is often distributed across multiple IPs rather than concentrated on individual addresses. These relational dependencies provide structural evidence of shared control or common operational strategies. Second, sender behaviors evolve over time in a structured manner. Group-level activities are not static: coordination patterns may persist across multiple periods, gradually change as campaigns progress, or shift in response to external conditions. Capturing this temporal evolution is essential for distinguishing stable organizational behavior from transient or incidental coordination.

The constructed temporal knowledge graph naturally reflects these two aspects by encoding coordination structures among entities within each timestamp and organizing observations as an ordered sequence of graph snapshots. To extract attribution-relevant patterns from this representation, we model sender behaviors through dual-stream recurrent graph evolution over the historical sequence (timestamps $t = 1, 2, \dots, T - 1$), which integrates intra-snapshot relational reasoning with inter-snapshot temporal modeling. Unlike continuous-time models [44] that operate at the level of individual events or query-driven methods [45] that encode per-query subgraphs, our R-GCN-based snapshot design enables joint reasoning over all facts within each snapshot through graph-level parallel state updates. As shown in Figure 3, the module operates through two types of recurrent processes distinguished by different edge colors: intra-snapshot coordination pattern modeling (gradient-colored edges) captures relational structures within each timestamp through relation-aware aggregation, while inter-snapshot behavioral evolution modeling (black edges) propagates entity and relation representations across timestamps. By recursively updating entity representations across successive graph snapshots, the model captures how coordination structures form, persist, and evolve over time. This joint modeling strategy enables robust sender group attribution by leveraging both persistent organizational patterns and dynamic behavioral changes. The two processes are described separately below for clarity, though they operate in a tightly integrated manner where structural information learned at each timestamp informs the temporal propagation.

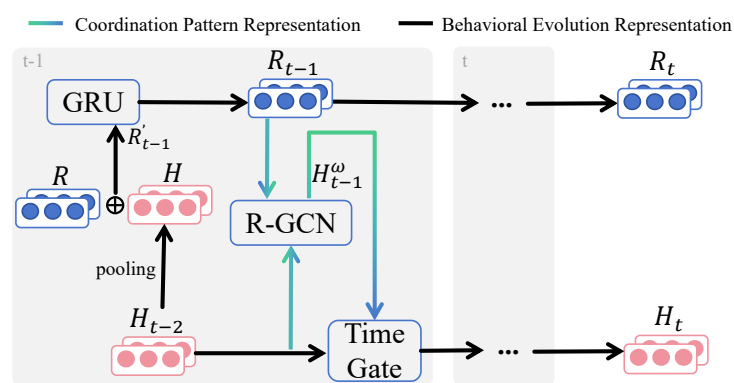


Figure 3. Recurrent graph evolution module for modeling historical graph dynamics. Gradient-colored edges represent intra-snapshot coordination pattern modeling through relation-aware aggregation, while black edges represent inter-snapshot propagation of entity and relation representations across time. H and R denote entity and relation representations respectively.

Coordination Pattern Representation. Within each snapshot G_t , we model coordination patterns among entities by capturing structured dependencies induced by multi-type interactions. Activities such as port scanning, session establishment, and infrastructure

reuse give rise to heterogeneous relations, where coordinated behaviors manifest through shared targets, repeated interaction motifs, and correlated activity patterns. To encode such relational structure, we employ Relational Graph Convolutional Networks (R-GCN) [46], which aggregate neighborhood information along typed edges through relation-aware aggregation so that entity embeddings jointly reflect both fine-grained behavioral interactions and higher-level organizational coordination. Formally, the R-GCN layer updates entity representations by propagating and aggregating messages conditioned on relation types as follows:

$$\mathbf{h}_i^{(l+1)} = \sigma \left(\sum_{r \in \mathcal{R}} \sum_{j \in \mathcal{N}_i^r} \frac{1}{c_{i,r}} \mathbf{w}_r^{(l)} \mathbf{h}_j^{(l)} + \mathbf{w}_0^{(l)} \mathbf{h}_i^{(l)} \right), \quad (1)$$

where $\mathbf{h}_i^{(l)} \in \mathbb{R}^d$ is the d -dimensional representation of entity i at layer l , $\mathcal{N}_i^r$ denotes the set of neighboring entities connected to entity i via relation r , $c_{i,r} = |\mathcal{N}_i^r|$ is a normalization constant, $\mathbf{W}_r^{(l)} \in \mathbb{R}^{d \times d}$ is a relation-specific transformation matrix, $\mathbf{W}_0^{(l)} \in \mathbb{R}^{d \times d}$ is the self-loop transformation, and $\sigma(\cdot)$ is the RReLU activation function. This relational GCN design allows the model to differentiate coordination patterns, such as shared sessions or synchronized port scans, from individual behaviors, while relation-specific transformations enable adaptive aggregation strategies for heterogeneous interaction types.

Behavioral Evolution Representation. Across successive timestamps, coordination patterns are not static but evolve as campaigns progress. As shown by the black edges in Figure 3, we capture this temporal evolution through two parallel recurrent mechanisms that propagate information across time. First, for entity-level evolution shown in the lower part of the figure, we employ a time-gated mechanism that operates on embeddings already encoding structural coordination from previous timestamps. For entity i , we update its embedding at timestamp t as

$$\mathbf{h}_i^{(t)} = (1 - \mathbf{U}_t) \odot \mathbf{h}_i^{(t-1)} + \mathbf{U}_t \odot \mathbf{h}_i^{(\text{GCN},t)}, \quad (2)$$

$$\mathbf{U}_t = \sigma(\mathbf{W}_u \mathbf{h}_i^{(t-1)} + \mathbf{b}_u), \quad (3)$$

where $\mathbf{h}_i^{(\text{GCN},t)}$ is the structural embedding from the relation-aware GCN, $\odot$ denotes element-wise multiplication, and the time gate $\mathbf{U}_t$ (depicted as the “Time Gate” component in Figure 3) adaptively balances historical stability with current observations. This time-gated design allows us to retain persistent coordination patterns while adapting to changes in entity behavior, such as shifts in scanning targets or evolving infrastructure usage.

Second, for relation-level evolution shown in the upper part of Figure 3, we update relation embeddings through GRU-based recurrence [47]. As indicated by the pooling arrow connecting entity embeddings to the GRU component, relation representations are updated by aggregating the embeddings of connected entities:

$$\mathbf{r}_t = \text{GRU}(\mathbf{r}_{t-1}, \text{pooling}(\{\mathbf{h}_i^{(t-1)}\}_{i \in V_r})), \quad (4)$$

where V_r is the set of entities connected by relation r . These relation-level embeddings capture evolving interaction patterns among entities, which inform and refine the R-GCN aggregation in subsequent timestamps, forming a coupled loop between entity-level and relation-level evolution. Through this recurrent process, the model learns attribution-relevant patterns that manifest both structurally at each timestamp and temporally across the historical sequence. The resulting entity representations $\mathbf{h}_i^{(T-1)}$ encode organizational behavioral patterns learned from historical observations, which serve as the foundation for attribution at the target timestamp T .

3.3. Structure Enhancement

The recurrent graph evolution captures sender coordination patterns across historical snapshots (timestamps 1 to $T - 1$), producing entity representations $\mathbf{h}_i^{(T-1)}$ that encode learned organizational behaviors. However, this historical modeling faces a critical limitation: newly emerged senders lack historical context. At timestamp T , a previously unseen IP may appear for the first time, leaving the recurrent evolution with no prior observations to leverage. Yet, such new senders often exhibit coordination patterns similar to known groups through their current interactions, for instance, accessing the same services, using similar protocols, or exhibiting comparable session depths.

To enable attribution for new senders without historical records, we introduce a structure enhancement module that operates on current observable facts G_T . This decoupled design separates two attribution signals:

- Historical evolution (from recurrent module): This IP has behaved like Organization A over the past weeks.
- Current similarity (from structure enhancement): This IP's current interactions resemble Organization B's typical patterns.

For established senders with a rich history, both signals reinforce each other. For new senders, only the current similarity signal is available, requiring the model to infer affiliations purely from behavioral matching. As shown in Figure 4, the structure enhancement module operates through three stages: message aggregation of neighbor information, scatter-based accumulation via bidirectional propagation, and adaptive fusion of historical and current signals through gating. We perform structure-aware enhancement by aggregating current neighbor information through single-step message passing with scatter operations:

$$\mathbf{m}_i = \frac{1}{|\mathcal{N}_i|} \sum_{j \in \mathcal{N}_i} (\mathbf{h}_j + \mathbf{r}_{ij}), \quad (5)$$

where $\mathcal{N}_i$ denotes the set of neighboring entities connected to entity i in the current graph, $\mathbf{h}_j$ is the neighbor embedding, and $\mathbf{r}_{ij}$ is the relation embedding connecting entities i and j . This aggregation is implemented efficiently using scatter-based operations that accumulate messages from multiple neighbors in parallel. Relation embeddings provide relational context, distinguishing coordination through different interaction types (e.g., port scanning vs. session establishment). The inclusion of relation embeddings is critical because identical neighbor sets connected through different relations may indicate different organizational affiliations, for example, shared infrastructure accessed for reconnaissance versus active exploitation. To handle the directed nature of network interactions, we employ bidirectional propagation: messages flow both from targets to senders (capturing “who they contact”) and from senders to targets (capturing “who contacts them”), effectively doubling the relational context by considering both outgoing and incoming edges. This message passing propagates coordination signals: if a new IP accesses domains frequently visited by Organization A's known IPs, those domain nodes carry embeddings influenced by Organization A, which then flow back to the new IP through aggregation.

However, not all entities should rely equally on historical versus current signals. A stable, long-observed IP with consistent behavior should primarily trust its historical patterns, while a newly appeared IP must depend entirely on current observations. To adaptively balance these two sources of information, we employ a cross-temporal gating mechanism (shown as “Time Gate” in Figure 4):

$$\bar{\mathbf{h}}_i = \mathbf{g}_i \odot \mathbf{h}_i^{(\text{hist})} + (1 - \mathbf{g}_i) \odot \mathbf{m}_i, \quad (6)$$

where $\mathbf{h}_i^{(\text{hist})} = \mathbf{h}_i^{(T-1)}$ denotes the final historical embedding from recurrent evolution, $\mathbf{m}_i$ is the aggregated message from current neighbors (Equation (5)), and the gate $\mathbf{g}_i = \sigma(\mathbf{W}_g[\mathbf{h}_i^{(\text{hist})}; \mathbf{m}_i] + \mathbf{b}_g)$ is computed by concatenating (denoted by $[\cdot; \cdot]$) the historical and current embeddings followed by a linear transformation and sigmoid activation. The gate adaptively balances historical stability with current observations based on the compatibility between the two signals. This time-gated enhancement allows the model to incorporate both persistent organizational behaviors and emerging interactions, such as new scanning campaigns or active port exploitation, directly from traffic observations at timestamp T .

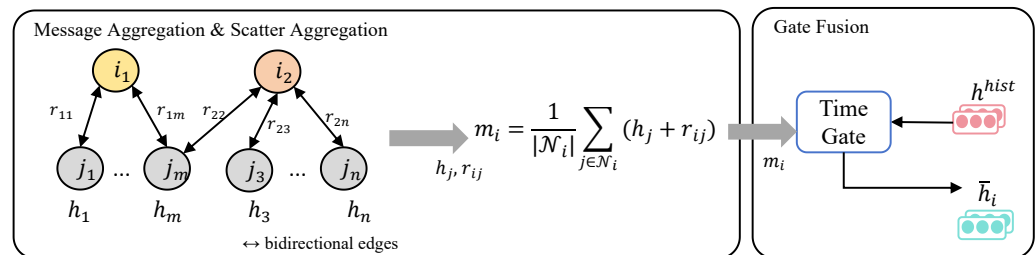


Figure 4. Structure enhancement module with three key operations: (1) Message aggregation combines neighbor embeddings $\mathbf{h}_j$ with relation embeddings $\mathbf{r}_{ij}$. (2) Scatter aggregation efficiently accumulates messages via bidirectional propagation to compute $\mathbf{m}_i$. (3) Gate fusion (Time Gate) adaptively balances historical embeddings $\mathbf{h}^{(\text{hist})}$ with current messages $\mathbf{m}_i$, producing enhanced embeddings $\bar{\mathbf{h}}_i$.

3.4. Sender Group Attribution

Given the enhanced entity representations $\bar{\mathbf{H}}_T$ from the temporal GNN encoder, our goal is to attribute each IP address to its corresponding sender organization. We formulate this as an entity prediction problem, estimating the plausibility of triples (s, r, o, T) at the target timestamp in the temporal knowledge graph.

Following previous work [48], we adopt the ConvTransE decoder, which captures interaction patterns between entity and relation embeddings via 1D convolution. For a candidate triple, with subject and relation embeddings $(\bar{\mathbf{h}}_s, \mathbf{h}_r)$, the probability vector over all entities is computed as:

$$\mathbf{p}(o \mid s, r, \bar{\mathbf{H}}_T, \bar{\mathbf{R}}_T) = \sigma\left(\bar{\mathbf{H}}_T^\top \text{ConvTransE}(\bar{\mathbf{h}}_s, \mathbf{h}_r)\right), \quad (7)$$

where $\bar{\mathbf{H}}_T \in \mathbb{R}^{|\mathcal{V}| \times d}$ is the matrix of all enhanced entity embeddings at timestamp T , $\sigma(\cdot)$ denotes the element-wise sigmoid function, and the result is a $|\mathcal{V}|$ -dimensional probability vector. Similarly, the probability vector over relations is:

$$\mathbf{p}(r \mid s, o, \bar{\mathbf{H}}_T, \bar{\mathbf{R}}_T) = \sigma\left(\bar{\mathbf{R}}_T^\top \text{ConvTransE}(\bar{\mathbf{h}}_s, \bar{\mathbf{h}}_o)\right). \quad (8)$$

where $\bar{\mathbf{R}}_T \in \mathbb{R}^{|\mathcal{R}| \times d}$ is the matrix of all relation embeddings. Note that $\bar{\mathbf{R}}_T = \mathbf{R}_{T-1}$, as relation embeddings evolve only through the recurrent graph evolution module over historical snapshots and remain fixed during structure enhancement at timestamp T (see Figure 1).

Although each IP address theoretically belongs to a single sender organization, we adopt a multi-label sigmoid formulation for entity prediction due to practical considerations in adversarial traffic analysis. This formulation supports open-world attribution, improves robustness to noisy organizational labels, and yields interpretable confidence scores. The primary learning objective is entity-level sender organization attribution at the target

timestamp. For each training query at timestamp T , the entity prediction loss is defined as a multi-label binary cross-entropy over all candidate entities:

$$\mathcal{L}_{\text{entity}} = - \sum_{(s,r,o^*,T) \in \mathcal{T}_{\text{train}}} \sum_{i=1}^{|\mathcal{V}|} y_i \log p_i(o \mid s, r), \quad (9)$$

where (s, r, o^*, T) denotes a training triple with ground-truth object o^* , $y_i \in \{0, 1\}$ indicates whether entity i is the ground-truth organization for source host s , and $p_i(o \mid s, r)$ is the i -th element of the predicted probability vector.

In large-scale traffic, entity-level supervision is often sparse and noisy, while coordination behaviors among senders, such as shared targets, scanning strategies, or protocol usage, are more consistently observable through relational patterns. To regularize entity representations with such a coordination structure, we introduce an auxiliary relation prediction loss:

$$\mathcal{L}_{\text{relation}} = - \sum_{(s,r^*,o,T) \in \mathcal{T}_{\text{train}}} \sum_{i=1}^{|\mathcal{R}|} y_{r,i} \log p_i(r \mid s, o), \quad (10)$$

where (s, r^*, o, T) denotes a training triple with ground-truth relation r^* , $y_{r,i} \in \{0, 1\}$ indicates whether relation i holds between entities s and o at timestamp T , and $p_i(r \mid s, o)$ is the i -th element of the predicted relation probability vector.

Importantly, relation prediction is not treated as an independent task. Instead, it serves purely as a regularization term that enforces structural consistency in the learned representations, encouraging entities involved in similar coordination behaviors to occupy coherent regions in the embedding space. Relation prediction is used only during training and is discarded at inference time. The final training objective emphasizes entity-level attribution, with relational regularization controlled by a weighting parameter:

$$\mathcal{J} = \lambda \mathcal{L}_{\text{entity}} + (1 - \lambda) \mathcal{L}_{\text{relation}}, \quad (11)$$

where $\lambda \in [0, 1]$ balances attribution accuracy against the strength of relational regularization.

4. Results

In this section, we conduct extensive experiments on real-world honeypot datasets to comprehensively evaluate SentinelGraph. We compare SentinelGraph with state-of-the-art sender group attribution methods, investigate the impact of temporal knowledge graph construction, recurrent graph evolution, structure enhancement, and auxiliary relation prediction loss, and analyze the underlying reasons for SentinelGraph's superior performance.

4.1. Experimental Setup

4.1.1. Implementation

We implement a prototype of SentinelGraph in Python 3.10 using PyTorch 2.2.1+cu118 and the Deep Graph Library (DGL) framework, which provides efficient graph neural network operations and supports large-scale graph processing. All experiments are conducted on a server equipped with an Intel Core i9-10900X CPU (3.70GHz), 256GB RAM, and an NVIDIA GeForce RTX 3090 GPU with 24GB memory.

4.1.2. Dataset

We evaluate our framework on real-world network traffic data collected in May 2024 by a distributed honeypot infrastructure. We deploy 50 honeypots across 8 geographic regions, with each honeypot running on a virtual machine configured with 2 CPU cores

and 2 GB RAM. Individual honeypots host 3-7 services selected from a pool of 19 available protocols, creating varied service combinations such as {HTTP, FTP, SSH}, {HTTP, MySQL, RTSP}, and {HTTP, Telnet, MongoDB}.

To establish ground truth for sender group attribution, we label each sender IP with organizational information through a multi-step process. First, we prioritize the highest-confidence labels by cross-referencing IPs against publicly disclosed IP lists from scanning organizations' official platforms (e.g., Censys, Shodan). Second, for remaining unlabeled IPs, we utilize WHOIS lookups combined with reverse DNS queries: an IP is labeled with an organization if WHOIS returns an organization name that is neither an ISP nor a cloud service provider, and reverse DNS returns a domain name explicitly linked to the same known scanning organization (e.g., census2.shodan.io for Shodan). Third, for IPs identified through reverse DNS, we extract organizational affiliation from the domain and derive naming patterns to discover additional IPs via reverse DNS and forward DNS queries, with each discovered IP independently validated against the second step's criteria. Through this labeling process, our dataset comprises 26,326 total unique sender IPs, among which 4838 are successfully labeled with organizational information spanning 38 distinct organizations. Table 1 summarizes the distribution of labeled sender IPs across major organizations, including the number of unique source hosts and corresponding traffic records for the most active organizations. To further analyze these ten major sender organizations, Figure 5 presents a heatmap of their temporal activity, target services, source ports, and session flow counts, revealing distinct behavioral patterns. While some organizations exhibit partially similar feature distributions, no two organizations display identical joint distributions, highlighting the diversity in their interaction patterns.

Following the knowledge graph construction method described in Section 3.1, we transform the traffic logs into a temporal knowledge graph. To ensure temporal consistency, we chronologically split the dataset: the first 25 days (80%) serve as the training set, days 26–28 (10%) as the validation set, and the final 3 days (10%) as the testing set.

Table 1. Statistics of source hosts and records for the top 10 organizations and others.

Sender Group Name	Unique Source Hosts	Count
stretchoid	2126	231,585
binaryedge	771	121,130
shadowserver	752	200,920
censys	294	768,250
internet-census	279	99,250
quake	145	472,615
internettl	91	13,825
internet-measurement	84	271
onyphe	66	9550
shodan	49	92,565
Others	181	185,235
Total	4838	2,195,196

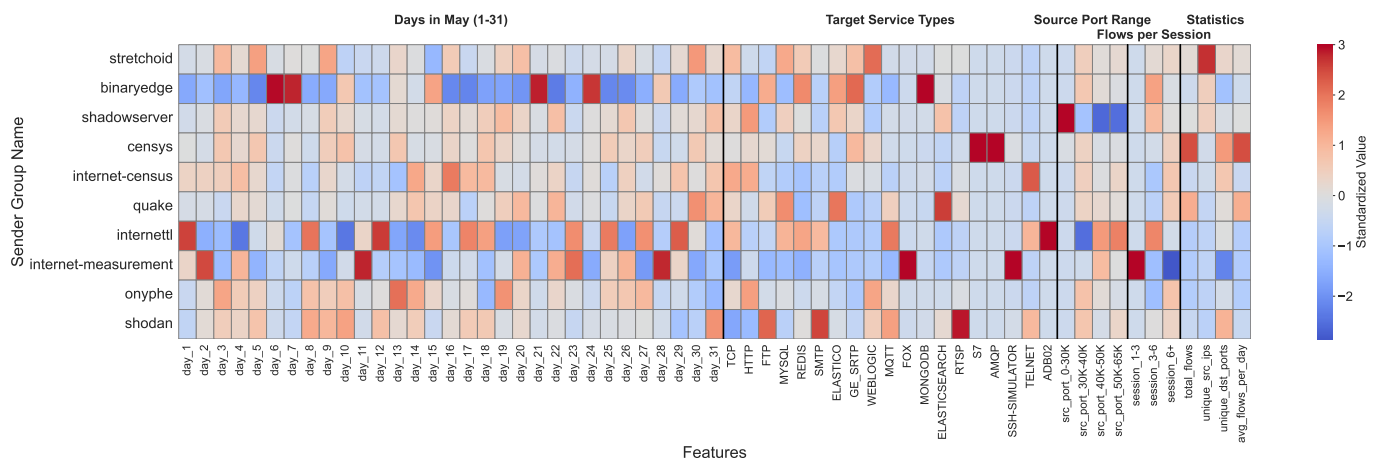


Figure 5. Heatmap of standardized features for the top 10 sender organizations in May, highlighting each organization's behavioral patterns. Rows represent organizations, and columns correspond to daily activity, target service types, source port ranges, flows per session, and overall statistics. Values are z-score standardized per feature: red indicates above-average, blue below-average, and white near-average activity, emphasizing relative differences in organizational patterns across multiple dimensions.

4.1.3. Evaluation Metrics

To comprehensively evaluate our approach under the single-label sender attribution setting, we report both classification and ranking metrics: (1) For comparison with existing baseline methods, which formulate sender attribution as a classification task, we report standard classification metrics including Accuracy and Precision, Recall, and F1-score. Specifically, Precision, Recall and F1-score are computed per class as $P = \frac{TP}{TP+FP}$, $R = \frac{TP}{TP+FN}$, $F1 = \frac{2 \times P \times R}{P+R}$, where TP denotes true positives, FP false positives, and FN false negatives. This ensures fair comparison since all baselines output a single predicted organization per query. (2) For ablation studies and detailed analysis of our temporal graph reasoning mechanism, we additionally report ranking metrics such as MRR and Hits@K with $K = 1, 3, 5$, while still computing P, R, and F1 by treating the top-ranked prediction as the final decision. Here, $MRR = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{1}{rank_i}$ measures the average reciprocal rank of the correct organizations ($rank_i$ is the position of the correct organization for query i), and $Hits@K = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \mathbf{1}\{rank_i \leq K\}$ measures the fraction of correct answers ranked within the top K predictions. Under our single-label attribution setting, Accuracy is equivalent to Hits@1, ensuring consistency between classification and ranking metrics.

4.1.4. Baselines

As discussed in Section 2, existing attribution approaches can be categorized into threat intelligence-based, static feature-based, and dynamic sequence-based methods. We exclude threat intelligence-based methods (e.g., [12–14]) because they either directly map IPs to known entities through external databases or incorporate threat intelligence as input features for attribution models. In contrast, our evaluation focuses on methods that learn attribution patterns from the traffic data itself. Among static feature-based methods, we select Scanner-Hunter [17] as our baseline because it represents a recent and comprehensive static feature-based approach with clear methodological design and reproducible implementation details. We adopt their core methodology of combining aggregated traffic features with Random Forest classification, but adapt the feature representation to our application scenario. For dynamic sequence-based methods, we choose DarkVec [19] because, like our honeypot deployment, it is designed for trap-based monitoring infrastructure,

whereas other sequence-based methods such as IP2Vec [18] and APTER [21] are designed for production network environments where traffic characteristics differ significantly.

As mentioned in Section 2, prior temporal graph-based methods in cybersecurity often focus on network or system-level flows, events, or alerts, modeling temporal or causal dependencies to detect anomalies, attacks, or alert patterns. In contrast, SentinelGraph operates over network flow data to reason about coordinated behaviors among multiple external actors, capturing heterogeneous relations and performing sender group attribution. These differences in reasoning units, relation semantics, and inference targets make prior methods not directly comparable as baselines. Instead, the effects of temporal graph design choices on attribution performance are assessed through controlled ablation experiments on SentinelGraph itself.

- Scanner-Hunter [17]: A scanning group identification system using Random Forest with One-vs-One (OVO) classification combined with IP prefix judgment. We adopt their ICS_RE feature representation and set the IP prefix length to /24 as suggested in their work. However, our dataset differs from theirs in two key aspects: (1) it encompasses protocols beyond ICS, and (2) not all payloads are unencrypted. Consequently, we exclude protocol-specific features from our representation. Additionally, since our dataset lacks predefined tool categories, we employ payload clustering as a substitute for tool categorization. We acknowledge that these adaptations may affect the baseline's performance compared to the original implementation; our goal is to fairly evaluate the static feature-based paradigm under our experimental setting rather than to reproduce the exact original method.
- DarkVec [19]: A Word2Vec-based approach that learns IP embeddings by treating IP addresses as words and their co-occurrence patterns in darknet traffic as sentences. We implement DarkVec with the recommended settings: embedding dimension of 128, context window of 10, and Skip-gram architecture with negative sampling. This method was originally designed for darknet traffic, where each connection contains only the initial packet due to the lack of responses. To evaluate this baseline, we adapt our dataset by retaining only the first SYN packet of each TCP connection and the first packet of each UDP session.

4.2. Attribution Performance

We evaluate the sender group attribution performance of SentinelGraph against state-of-the-art methods. Table 2 summarizes the comparison between static feature-based approaches (Scanner-Hunter) and dynamic sequence-based approaches (DarkVec), where all comparison experiments are independently repeated multiple times with different random initializations, and statistical significance tests are applied to support claims of superiority over baseline methods. Table 3 further presents the per-class performance breakdown for the top 10 most frequent sender groups by unique IP count (See Table 1), revealing the behavior of different methods across diverse scanner organizations.

Table 2. Overall performance comparison of sender group attribution methods. SentinelGraph significantly outperforms all baseline methods ($p < 0.001$).

Methods	Accuracy	F1 (Macro)	F1 (Weighted)	P (Weighted)
Scanner-Hunter	0.452	0.407	0.477	0.695
Darkvec	0.635	0.411	0.629	0.803
SentinelGraph (Ours)	0.935	0.886	0.965	0.966

Table 3. Per-class performance on top 10 sender groups by IP count across different attribution methods.

Class	SentinelGraph			Scanner-Hunter			Darkvec		
	Precision	Recall	F1-Score	Precision	Recall	F1-Score	Precision	Recall	F1-Score
stretchoid	0.943	0.922	0.932	0.748	0.283	0.409	0.805	0.782	0.793
binaryedge	0.887	0.732	0.802	0.883	0.252	0.394	0.318	0.982	0.480
shadowserver	0.998	0.993	0.996	0.877	0.503	0.641	0.956	0.998	0.977
censys	0.997	0.956	0.976	0.718	0.447	0.552	0.874	0.308	0.456
internet-census	0.912	0.948	0.930	0.782	0.518	0.623	0.679	0.471	0.556
quake	1.000	1.000	1.000	0.653	0.382	0.481	0.942	0.745	0.832
internettl	0.908	0.813	0.858	0.897	0.648	0.753	0.998	0.177	0.301
internet-measurement	0.742	0.597	0.662	0.247	0.153	0.189	0.000	0.000	0.000
onyphe	0.778	0.927	0.846	0.803	0.423	0.554	0.997	0.086	0.158
shodan	0.976	0.998	0.987	0.896	0.478	0.624	0.998	0.059	0.111

SentinelGraph achieves an F1 score of 0.886 (Macro) and 0.965 (Weighted), substantially outperforming DarkVec (F1 Macro: 0.411, F1 Weighted: 0.629) and Scanner-Hunter (F1 Macro: 0.407, F1 Weighted: 0.477). The strong correlation between macro and weighted F1 scores indicates robust performance across highly imbalanced sender groups, while also revealing fundamental limitations of static feature-based and dynamic sequence-based approaches in real-world scenarios.

Scanner-Hunter's poor performance stems from its heavy reliance on payload information and protocol-specific feature engineering. As observed in the per-class results in Table 3, the method exhibits consistently low recall across most sender groups, particularly for large Internet-wide measurement organizations such as Stretchoid (0.283) and BinaryEdge (0.252), indicating a systematic failure to generalize across heterogeneous scanning behaviors. This pattern arises because many contemporary measurement infrastructures employ diverse protocols and encrypted communication, a trend that directly undermines payload-dependent and protocol-specific features. Once protocol-related features are removed to accommodate this diversity, the remaining static features become weakly discriminative and insufficient for reliable attribution, as treating each time window independently makes attribution vulnerable when absolute behavioral features lose discriminative power. Furthermore, Scanner-Hunter requires explicit tool category labels (e.g., nmap, zmap), which are rarely available in practice as they depend on threat intelligence or deep payload inspection. Our necessary substitution of tool categories with payload clustering further introduces noise and deviates from the method's original design assumptions. Consequently, static feature-based approaches such as Scanner-Hunter degrade substantially under realistic scanning conditions where protocol diversity and encryption are prevalent.

DarkVec exhibits a distinct performance pattern that is closely tied to sender group scale and data density. As shown in the per-class results in Table 3, where sender groups are ordered by IP count, DarkVec achieves noticeably better performance for groups in the upper part of the table, such as Stretchoid and Shadowserver, while its performance degrades sharply for groups with fewer IPs or sparser activity patterns. More specifically, DarkVec performs best when a sender group is characterized by both a large number of IPs and rich interaction records per IP, degrades when IP participation becomes sparse, and fails almost entirely for groups where both IP diversity and per-IP interaction records are limited, such as Internet-Measurement. This trend suggests that DarkVec benefits strongly from dense IP co-occurrence signals and struggles when such signals become weak or fragmented. This behavior can be attributed to DarkVec's reliance on IP co-occurrence-based embeddings, which require sufficient repetition and overlap to learn

stable representations. When sender groups generate limited interactions per IP or exhibit bursty and irregular scanning behaviors, the resulting co-occurrence patterns become too sparse to support effective attribution. Furthermore, our inspection reveals that DarkVec generalizes poorly to newly observed IPs that appear only in the test set but are absent from the training data, as the lack of prior co-occurrence context prevents DarkVec from assigning meaningful embeddings, leading to severe recall loss. It is worth noting that DarkVec was originally designed for darknet traffic and was adapted in our evaluation to retain only the first packet of each connection to align with its intended operating assumptions; nevertheless, these limitations persist under realistic honeypot settings with high IP churn and uneven interaction density.

SentinelGraph does not rely on static payload features or IP co-occurrence statistics, distinguishing it from both Scanner-Hunter and DarkVec. Instead, it models sender interactions as session-level temporal graphs and integrates historical behavioral evolution with current observable context, enabling robust attribution across diverse sender groups (Table 3). This approach consistently achieves high precision and recall across organization sizes (effectively handling class imbalance), even for groups exhibiting sparse or irregular scanning behaviors. Nevertheless, error analysis reveals remaining challenges for extremely sparse patterns. For BinaryEdge, two distinct scanning behaviors are observed: one involves daily scans over a month targeting a port with no deployed service, while the other consists of scans on a single day targeting deployed services such as Redis or SMTP. SentinelGraph handles the former effectively, but exhibits reduced recall on the latter, where signals are short-lived and weakly connected. A more pronounced limitation occurs for internet-measurement, whose IPs appear only one to three days per month, each generating only three to four interaction records per day and belonging to a single session. With our 24-h time window and session-level graph construction, such interactions form extremely sparse graphs, limiting the propagation of coordination signals despite our modeling of synchronized adaptation patterns. Overall, these cases have only a minor effect on SentinelGraph's performance, which remains robust and effective across most sender groups.

4.3. Ablation Study

To validate the contribution of each component in SentinelGraph, we conduct a comprehensive ablation study spanning multiple levels of model design, including temporal graph formulation, core architectural paradigms for temporal reasoning, and auxiliary training objectives. Our method is built on three key hypotheses: (1) session-level aggregation better captures organizational coordination than finer-grained flow-level modeling (Section 4.3.1); (2) effective attribution requires jointly modeling historical behavioral evolution and current structural signals, with adaptive fusion between the two, which entails distinct architectural paradigms for temporal graph learning (Sections 4.3.2 and 4.3.3); and (3) relation prediction serves as structural regularization rather than an independent objective (Section 4.3.4). We validate these hypotheses through controlled experiments that isolate the impact of each design choice.

4.3.1. Impact of the Temporal Knowledge Graph Construction

The construction of temporal knowledge graphs requires discretizing continuous traffic streams into temporally meaningful interaction units. Two fundamental design choices determine this temporal abstraction: the granularity of interaction modeling (flow-level vs session-level) and the hyperparameters governing session aggregation (session depth threshold k and temporal window size h). We systematically investigate these choices to understand their impact on attribution performance.

Session-level modeling outperforms flow-level approaches. We compare session-based modeling against flow-level alternatives to assess the trade-offs between information preservation and semantic abstraction. Four configurations are evaluated at a fixed temporal window ($h = 24$ h) and session threshold ($k = 2$): (1) Flow w/ Payload: each flow becomes a graph edge with payload-derived features (content patterns); (2) Flow w/o Payload: individual flows using only basic information (source and destination IPs, source and destination ports, target services, timing); (3) Session w/ Distinction ($k = 2$): sessions distinguished by depth threshold; (4) Session w/o Distinction ($k = 0$): uniform sessions without depth differentiation.

As shown in Table 4, session-based modeling with depth distinction achieves the highest MRR (0.951) and Hits@1 (0.935), outperforming flow-level approaches even when payload features are included (Flow w/ Payload: $MRR = 0.812$, $Hits@1 = 0.768$). The performance gap suggests that session-level aggregation captures organizational patterns more effectively than fine-grained flow information in our honeypot traffic. Interestingly, payload features provide only moderate improvements in flow-level modeling (Flow w/ Payload vs. w/o Payload: +0.088 MRR), indicating that while payload information can reveal certain organizational characteristics, the prevalence of encrypted payloads limits its overall contribution to attribution performance in our dataset [27,49]. Session-based modeling also achieves $2.5\times$ to $2.8\times$ speedup in test-time runtime (18 s vs. 45–50 s for flow-level) while reducing graph size substantially. Even Session w/o Distinction ($MRR = 0.893$) outperforms Flow w/ Payload despite the simpler representation, though depth distinction ($k = 2$) contributes an additional 0.058 in MRR. These results suggest that the semantic context provided by session aggregation, capturing complete interaction sequences rather than isolated flows, outweighs the information loss from aggregation.

Table 4. Comparison of interaction granularity in graph construction.

Granularity	MRR	Hits@1	RunTime (Seconds)
Flow w/ Payload	0.812	0.768	50
Flow w/o Payload	0.724	0.661	45
Session w/ Distinction	0.951	0.935	18
Session w/o Distinction	0.893	0.865	18

Session depth thresholds exhibit a stable optimum at $k = 2$ across temporal windows. We conduct a comprehensive grid search over temporal window sizes $h \in 1, 2, 6, 12, 24$ h and session size thresholds $k \in 0, 1, 2, 3, 4, 5$, evaluating all 30 (h, k) combinations. The baseline $k = 0$ represents uniform session modeling without depth distinction (all sessions labeled as SessionTo regardless of size). As shown in Figure 6, performance consistently peaks at $k = 2$ across all five temporal windows and all four metrics, demonstrating that $k = 2$ corresponds to a consistent behavioral separation observed in our honeypot traffic across temporal resolutions. The threshold $k = 2$ effectively partitions sessions into short sessions (1–2 flows) and long sessions (3+ flows), distinguishing opportunistic scanning from systematic reconnaissance. This simple distinction of session relations proves critical for attribution. Without session relation type distinction (i.e., at $k = 0$, all session interaction edges are of a single relation type), our model treats relations as learnable vectors similar to entities, resulting in learning less discriminative information from session interactions. This leads to performance degradation compared to configurations with session classification ($k > 0$). Across all temporal windows, Macro-F1 at $k = 0$ is consistently 4 to 15 points lower than the best-performing configurations that distinguish

session interaction edges, demonstrating that the ability to differentiate interaction intensity provides an essential attribution signal under optimal distinction thresholds.

By contrast, when session interaction relations are distinguished ($k > 0$), both overly sensitive and overly restrictive thresholds degrade performance, revealing the importance of the 3+ flow criterion in our honeypot traffic dataset. The threshold $k = 1$ classifies any 2+ flow session as “long reconnaissance”. However, empirical analysis of our dataset shows that 2-flow sessions often represent transient interactions (connection attempts followed by single probes, or failed handshakes) rather than systematic reconnaissance. This introduces noise that fragments the organizational signal, leading to performance drops: 24 h yields 0.822 (6.4 points below optimum), 1 h yields 0.781 (11.1 points below). Conversely, the thresholds $k = 3, 4, 5$ require 4, 5, or 6+ flows to qualify as “long sessions”. Our data analysis reveals that many substantive reconnaissance activities (targeted port scans, service enumeration sequences) in the dataset involve exactly 3–5 flows, and these higher thresholds misclassify such legitimate reconnaissance as shallow probing, weakening the depth distinction. For 24 h: $k = 3$ yields 0.826, $k = 4$ yields 0.842, $k = 5$ yields 0.839, all substantially below 0.886. The optimal threshold $k = 2$ balances precision (correctly identifying persistent reconnaissance) and recall (not missing legitimate reconnaissance activities) based on the behavioral patterns observed in our dataset, providing the strongest organizational fingerprint across all temporal scales. While the exact numeric threshold is data-dependent, the underlying principle of separating transient interactions from sustained reconnaissance is broadly applicable to honeypot deployments that support session reconstruction.

Daily temporal windows achieve optimal performance and efficiency. From the perspective of temporal windows, Table 5 summarizes peak performance, ranking quality metrics, and computational cost across temporal windows at the optimal session threshold. Daily windows ($h = 24$) achieve the highest MRR (0.951), Hits@1 (0.935), Hits@3 (0.964), and Hits@10 (0.977) while requiring the shortest runtime (18 s). The superiority of 24-h aggregation aligns with how coordinated reconnaissance operations actually unfold. Organized threat actors typically follow daily operational cycles: they distribute scanning activities across multiple hours to evade rate-limiting and detection mechanisms, while completing target enumeration within operational deadlines (e.g., before IP rotation or resource reallocation). Daily windows naturally encapsulate these complete campaigns without fragmenting semantically related activities, preserving campaign-level structure that carries strong organizational signals. In contrast, while hourly aggregation also yields strong ranking quality ($MRR = 0.946$, $Hits@10 = 0.973$), it requires 92 s of runtime, representing a $5.1\times$ slowdown compared to daily windows with slightly lower ranking metrics overall. For large-scale deployment processing millions of flows daily, this computational overhead is prohibitive. Moreover, Figure 6 shows hourly windows exhibit greater sensitivity to session threshold selection, with steeper performance drops at suboptimal k values, requiring more careful tuning. This sensitivity suggests that hourly windows fragment some coordinated activities that span multiple hours, making the model more dependent on precise session depth encoding to recover the organizational signal.

Table 5. Effect of temporal window size on sender group attribution performance.

Window Size	MRR	Hits@1	Hits@3	Hits@10	RunTime (Seconds)
1 h	0.946	0.941	0.952	0.973	92
2 h	0.934	0.921	0.941	0.961	69
6 h	0.942	0.932	0.946	0.961	41
12 h	0.934	0.921	0.944	0.956	30
24 h	0.951	0.935	0.964	0.977	18

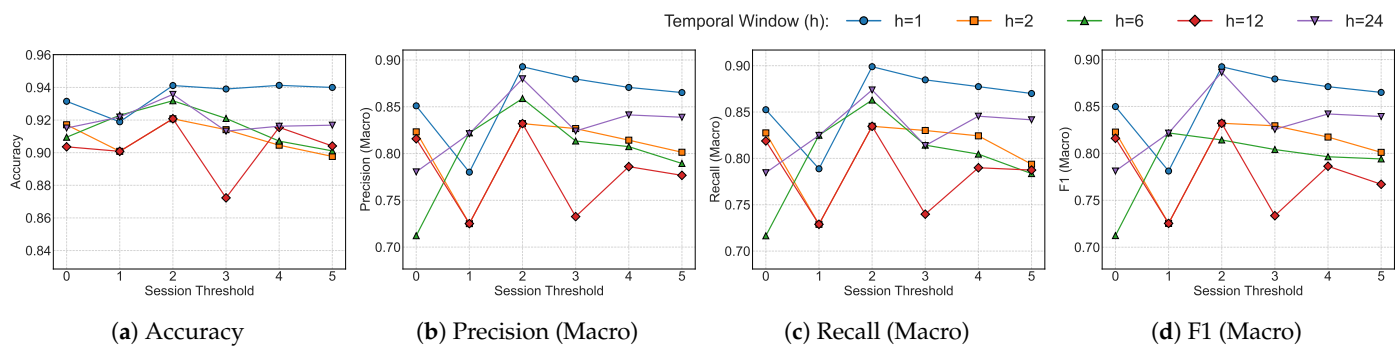


Figure 6. Impact of temporal window size (h) and session threshold (k) on sender group attribution performance. Each subplot shows how a specific evaluation metric varies across different temporal window configurations (1 h, 2 h, 6 h, 12 h, 24 h) and session thresholds (0–5): (a) overall accuracy, (b) macro-averaged precision, (c) macro-averaged recall, and (d) macro-averaged F1 score. All metrics consistently demonstrate that performance peaks at intermediate session thresholds ($k = 2$), with both 1-h and 24-h windows achieving strong results.

4.3.2. Impact of the Recurrent Graph Evolution

The recurrent graph evolution module integrates two complementary mechanisms: intra-snapshot coordination modeling through relational GCN and inter-snapshot behavioral evolution through temporal propagation. As detailed in Section 3.2, this module relies on four key design choices: (1) relation-aware aggregation via R-GCN for capturing typed coordination patterns within each snapshot, (2) time-gated recurrent updates for adaptively integrating historical context, (3) historical propagation across timestamps for tracking persistent organizational behaviors, and (4) relation-level recurrence through GRU for evolving interaction semantics. To validate the necessity of each component, we systematically disable them and measure the resulting performance degradation. Figure 7 presents the results across multiple metrics.

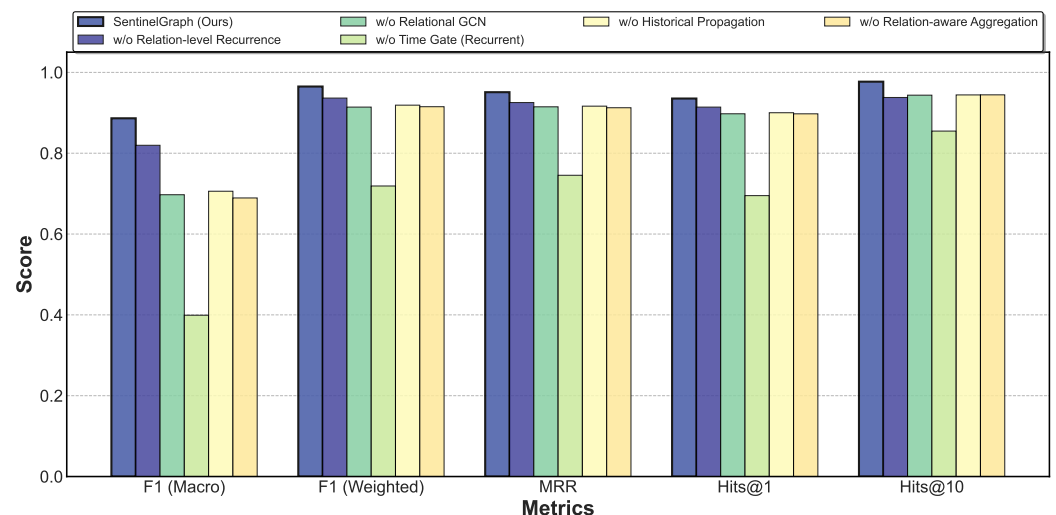


Figure 7. Ablation study performance on key design choices in the recurrent graph evolution module. All designs contribute to overall performance, with time gating mechanisms being the most critical for capturing behavioral evolution across timestamps.

Relation-aware aggregation is essential for capturing coordination patterns. We first examine whether relation-specific transformations in R-GCN (Equation (1)) are necessary for capturing coordination patterns within each snapshot. Without relation-aware aggregation, the model treats all edges uniformly regardless of interaction type, effectively reducing the R-GCN to a standard GCN and shifting from typed relational reasoning to

homogeneous neighborhood aggregation. Under this setting, entity-level ranking performance degrades noticeably, with MRR dropping to 0.913 and Macro F1 to 0.689. The model can no longer distinguish between different interaction semantics and thus conflates coordination through shared sessions with coordination induced by synchronized port scans, aggregating heterogeneous neighborhood signals indiscriminately. This validates our design rationale: aggregation must be conditioned on relation types to differentiate genuine organizational coordination from incidental structural similarities. While relation prediction performance remains perfect ($MRR = 1.0$), indicating that basic relation identification is still feasible, the substantial degradation in entity-level attribution metrics highlights that relation-aware aggregation is critical for encoding coordination structures that drive accurate sender group attribution.

The time gate enables adaptive integration of historical context. We next investigate whether the time-gated recurrent mechanism (Equation (2)) is necessary, or whether simpler temporal integration suffices. Removing the learnable gate U_t , we replace the gated update with a fixed averaging scheme: $\mathbf{h}_i^{(t)} = 0.5 \cdot \mathbf{h}_i^{(t-1)} + 0.5 \cdot \mathbf{h}_i^{(GCN,t)}$. This leads to a sharp deterioration in entity-level attribution performance, with MRR dropping to 0.745, Hits@1 to 0.695, and Macro F1 collapsing to 0.399. Notably, this is substantially worse than removing historical context entirely ($MRR = 0.916, F1 = 0.706$), indicating that naïve temporal integration without adaptive control is counterproductive. The time gate determines when persistent coordination patterns should be retained and when new observations should dominate, based on behavioral consistency across timestamps. Without such adaptive gating, the model either over-emphasizes outdated coordination structures or becomes overly sensitive to transient fluctuations, failing to distinguish stable organizational behaviors from temporary anomalies.

Historical propagation is necessary for tracking persistent organizational patterns. To assess whether temporal evolution is necessary at all, we remove historical propagation and process only the current snapshot G_{T-1} , without any recurrent updates across timestamps. This results in an entity-level MRR of 0.916 and a Macro F1 of 0.706. While the model can still capture intra-snapshot coordination through relation-aware aggregation, it loses the ability to model how coordination patterns persist and evolve over time. Sender groups often exhibit recurring behaviors across multiple timestamps, such as consistent target selection or sustained infrastructure usage. Without historical propagation, each snapshot is treated independently, preventing the model from distinguishing stable organizational patterns from sporadic or coincidental activities. The roughly 15-point drop in Macro F1 compared to the full model demonstrates that temporal evolution through recurrent propagation is essential for robust sender group attribution.

Relation-level recurrence provides complementary signals for attribution. Finally, we examine whether evolving relation embeddings through the GRU mechanism (Equation (4)) provides additional benefits beyond entity-level temporal modeling. When relation embeddings are kept static across timestamps ($\mathbf{r}_t = \mathbf{r}_0$), the model achieves an entity-level MRR of 0.925 and a Macro F1 of 0.8196. Performance remains relatively strong, as entity-level recurrent propagation continues to capture most coordination dynamics. However, the consistent performance degradation across metrics indicates that relation-level recurrence contributes complementary information. Evolving relation representations encode how interaction semantics shift over time, such as changes in scanning strategies or target distributions. By aggregating entity embeddings into relation contexts and propagating them through a GRU, the model gains a higher-level temporal view of coordination dynamics beyond individual entity behaviors. The moderate impact suggests that while relation-level recurrence enhances attribution performance, the primary temporal signal is driven by entity-level recurrent propagation.

Overall, these results validate our design principle articulated in Section 3.2: coordination patterns are expressed through typed interactions within snapshots, requiring relation-aware aggregation, while group-level behaviors emerge from how these patterns evolve over time, necessitating historical propagation with adaptive gating. Both relational reasoning and temporal modeling are indispensable, and their integration through time-gated updates is critical for accurately capturing organizational coordination behaviors.

Performance improves with historical context up to 25 snapshots. Beyond architectural choices, we further analyze the effect of history length on model performance. We vary the number of historical snapshots m used for recurrent evolution. Using only the current-time snapshot (no historical propagation) yields an MRR of 0.874, confirming that temporal context is important. Incorporating past snapshots consistently improves performance, with MRR increasing from 0.889 at $m = 5$ to a peak of 0.951 at $m = 25$ (Table 6). Notably, performance gains diminish beyond $m = 15$, suggesting that sender group behaviors exhibit characteristic patterns within roughly 15 snapshots (approximately 15 weeks in our dataset), and adding longer histories provides limited additional benefit while potentially introducing gradient vanishing or overfitting to outdated patterns. For practical deployment, using $m = 25$ snapshots offers the best balance between attribution accuracy and computational cost.

Table 6. Effect of historical window length m on sender group attribution performance (MRR and Hits@k).

History Length	MRR	Hits@1	Hits@3	Hits@10
$m = 1$ (No history)	0.874	0.842	0.893	0.929
$m = 5$	0.889	0.855	0.915	0.951
$m = 10$	0.931	0.919	0.936	0.956
$m = 15$	0.943	0.925	0.958	0.973
$m = 20$	0.931	0.920	0.937	0.953
$m = 25$	0.951	0.935	0.964	0.977
$m = 30$	0.914	0.894	0.929	0.944

4.3.3. Impact of the Structure Enhancement

The structure enhancement module addresses a critical limitation of purely historical modeling: the inability to attribute newly emerged senders without prior observations. As described in Section 3.3, this module incorporates three key mechanisms: (1) single-step message passing with scatter aggregation for efficiently gathering current coordination signals, (2) relation embeddings for distinguishing heterogeneous interaction types, (3) bidirectional propagation for capturing reciprocal coordination patterns, and (4) a time gate for adaptively fusing historical behaviors with current observations. To validate the contribution of each design choice, we systematically ablate these components and measure the resulting performance degradation. Figure 8 presents the results.

Structure enhancement is critical for incorporating current coordination signals. Removing the entire structure enhancement module leads to catastrophic performance degradation across all metrics. F1 (Macro) plummets from 0.886 to 0.456, while MRR drops from 0.951 to 0.576. This dramatic decline demonstrates that current interaction facts at timestamp T , including session characteristics, port usage, and host-level relations observed in G_T , provide essential contextual signals for accurate sender attribution.

Further analysis reveals that 83% of failed attributions without structure enhancement correspond to newly emerged senders at the target timestamp. These senders lack historical records in the recurrent evolution module (timestamps 1 to $T - 1$), leaving the model with no prior behavioral patterns to leverage. Without structure enhancement, the model relies solely on historical patterns and cannot attribute these cold-start cases, regardless

of how clearly their current interactions reveal organizational coordination. This finding validates our architectural decision to decouple temporal propagation from structural enhancement: while historical evolution captures persistent organizational patterns for established senders, current structure enhancement enables attribution based on behavioral similarity, which is particularly crucial for cold-start scenarios where senders lack historical records. The ability to attribute new senders through current coordination signals is essential in adversarial traffic analysis, where threat actors frequently rotate infrastructure or launch campaigns from previously unseen IP addresses.

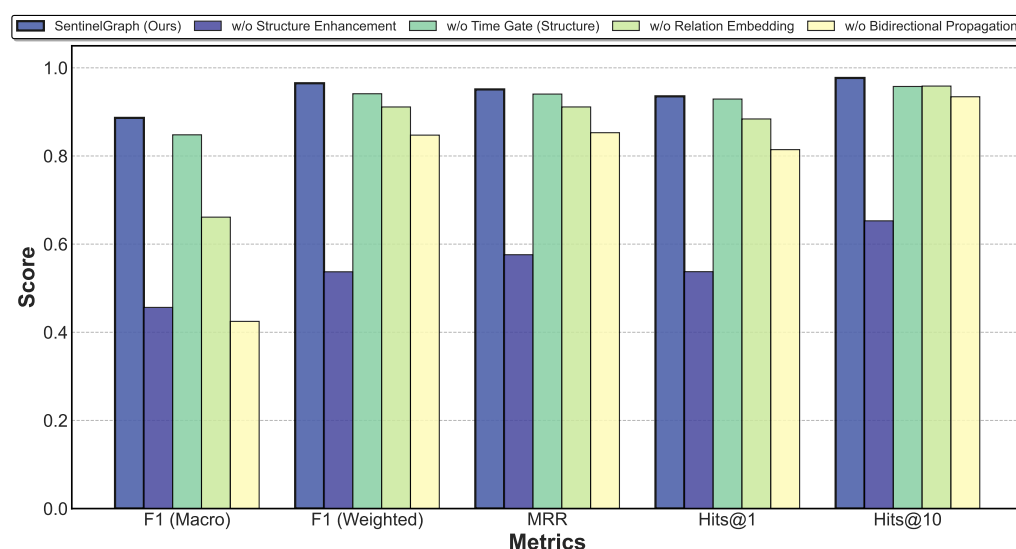


Figure 8. Ablation study performance on the structure enhancement module and its key design choices. Removing the entire module causes the most severe degradation, with bidirectional propagation being the most critical individual design.

Relation embeddings distinguish heterogeneous interaction types. When relation embeddings r_{ij} are removed from the message passing mechanism (Equation (5)), treating all edges uniformly and reducing relation-aware message passing to homogeneous neighborhood aggregation $\mathbf{m}_i = \frac{1}{|\mathcal{N}_i|} \sum_{j \in \mathcal{N}_i} \mathbf{h}_j$, F1 (Macro) drops significantly to 0.661. This 22.5% performance degradation validates the importance of relation-aware message passing. Different interaction types, such as protocol-based connections, port-based scans, and session-based communications, carry distinct semantic meanings for inferring organizational coordination patterns. As discussed in Section 3.3, identical neighbor sets connected through different relations may indicate different organizational affiliations (e.g., shared infrastructure accessed for reconnaissance versus active exploitation). By explicitly modeling these relation types through learned embeddings, the model effectively distinguishes between heterogeneous interaction patterns and captures the nuanced ways different organizations coordinate their activities.

Bidirectional propagation captures reciprocal coordination patterns. Restricting message passing to unidirectional flow (only from targets to senders) results in severe performance loss, with F1 (Macro) dropping to 0.425 and MRR declining to 0.853. This degradation, comparable to removing the entire module, reveals that coordination patterns in network traffic are inherently bidirectional. As emphasized in our design rationale, sender behaviors are reflected not only in “who they contact” (outgoing edges) but also in “who contacts them” (incoming edges). Bidirectional propagation enables the model to capture reciprocal relationships and mutual dependencies between entities. For instance, a coordinated scanning campaign manifests in both the attack vectors initiated by senders and the response patterns from targets. This bidirectional information flow doubles the

relational context and is crucial for accurately identifying organizational membership based on interaction patterns.

Time gate provides adaptive fusion of historical and current evidence. While removing the learnable gate g_i (Equation (6)) and replacing it with fixed averaging ($\bar{\mathbf{h}}_i = 0.5 \cdot \mathbf{h}_i^{(\text{hist})} + 0.5 \cdot \mathbf{m}_i$) still yields relatively strong performance (F1 Macro: 0.848, MRR: 0.940), a notable gap remains compared to the full model. The time-gated mechanism adaptively balances historical organizational patterns $\mathbf{h}_i^{(\text{hist})} = \mathbf{h}_i^{(T-1)}$ from the recurrent evolution module with current observations $\mathbf{m}_i$ from structure enhancement. This adaptive fusion is particularly valuable when organizations shift their strategies or launch new scanning campaigns. Rather than applying a fixed weighting, the learnable gate allows the model to dynamically adjust the importance given to recent evidence versus established historical trends on a per-entity basis. For stable, long-running organizations, the gate may favor historical patterns, while for emerging threats or tactical shifts, it can emphasize current interaction signals. The consistent performance gains across metrics demonstrate that this entity-specific adaptive mechanism is more effective than uniform fusion strategies.

These ablation results validate the design principles articulated in Section 3.3: current coordination signals are essential for attribution (especially for new senders), relation-aware and bidirectional message passing effectively captures heterogeneous coordination patterns, and adaptive gating enables robust fusion of complementary evidence sources. Together, the structure enhancement module complements historical evolution by grounding attribution in both persistent behaviors and immediate observations.

4.3.4. Impact of the Auxiliary Relation Prediction Loss

To systematically study the effect of relation prediction in the attribution objective, we perform a grid search over the weighting parameter $\lambda \in \{0.5, 0.6, 0.7, 0.8, 0.9, 1.0\}$ in the objective defined in Equation (11). The case $\lambda = 1.0$ corresponds to removing the relation regularizer entirely.

Relation prediction is most effective when used as a lightweight regularizer rather than a co-equal objective. As shown in Table 7, introducing a small amount of relation regularization consistently improves attribution performance over the no-regularization baseline. The best performance is achieved at $\lambda = 0.9$, yielding a 13.9% improvement in MRR compared to $\lambda = 1.0$. When λ approaches 1, the entity attribution objective dominates, while the auxiliary relation loss provides a mild but effective structural inductive bias. As λ decreases, relation regularization becomes more influential, but over-emphasizing it gradually degrades overall performance, indicating that the model is distracted from the primary attribution objective. Minor fluctuations in Hits@1 or Hits@3 (e.g., $\lambda = 0.7$ vs. 0.8) are observed, but they do not contradict the overall trend. Overall, these results confirm that relation prediction functions best as a lightweight regularizer, validating our design choice to prioritize entity-level sender attribution.

Table 7. Effect of auxiliary loss weight λ on sender group attribution performance.

λ	MRR	Hits@1	Hits@3	Hits@10
1.0 (No auxiliary loss)	0.885	0.829	0.938	0.957
0.9 (Ours)	0.951	0.935	0.964	0.977
0.8	0.951	0.902	0.939	0.955
0.7	0.937	0.927	0.940	0.960
0.6	0.904	0.878	0.925	0.946
0.5	0.895	0.872	0.918	0.940

4.4. Case Study

To illustrate SentinelGraph's sender group attribution on previously unseen IPs, we present a representative case from our test set. On day 30 of the test period, IP address 162.142.x.x appeared for the first time, with no historical observations in the training data from days 1 to 25. Ground truth verification via official sources confirms that this IP belongs to Censys, a well-known Internet measurement service. Within a 24-h window, the IP exhibited multiple sessions across different services, all categorized as LongSessionTo relations:

- HTTP (port 8080): 8 sessions, 5–15 flows each.
- MQTT (port 1883): 1 session, 10 flows.
- RTSP (port 554): 1 session, 5 flows.

These interactions illustrate the activity pattern used by the structure enhancement module (Section 3.3) to infer the IP's identity in the absence of historical embeddings from the recurrent graph evolution module (Section 3.2). Through message passing (Equation (5)), the model aggregates information from neighboring service nodes. The same services were accessed by multiple known Censys IPs during the same time window, with 15 for HTTP, 12 for MQTT, and 8 for RTSP, which creates strong structural coordination signals. The observed pattern of multiple long sessions across HTTP, MQTT, and RTSP matches Censys's characteristic scanning behavior documented in our training data. The aggregated structural message produces a substantial update to the IP representation, with the embedding norm increasing from 1.0 (base embedding) to 6.26 and a change magnitude of 5.98. The cross-temporal gate assigns a weight of 0.61 to the current structural component, indicating that attribution is mainly driven by present interactions rather than historical information.

SentinelGraph assigns the highest probability to Censys at 0.927, substantially ahead of alternative organizations such as Stretchoid (0.068) and BinaryEdge (0.042), correctly attributing the new IP with high confidence. In contrast, baseline methods struggle: Scanner-Hunter cannot predict without historical features, while DarkVec incorrectly predicts BinaryEdge (confidence 0.324) due to insufficient structural distinction between Censys and BinaryEdge scanning patterns. This case represents typical cold-start performance. Across 388 previously unseen IPs in our test set, SentinelGraph achieves 91.2% attribution accuracy with an average confidence of 0.883, demonstrating robust generalization to new entities.

5. Conclusion and Future Work

In this paper, we propose SentinelGraph, a temporal graph reasoning framework for sender group attribution from honeypot traffic. By constructing a honeypot traffic knowledge graph and jointly modeling coordination structures and temporal evolution, SentinelGraph effectively captures group-level behavioral patterns without reliance on external threat intelligence. Extensive experiments on real-world honeypot data demonstrate that explicitly reasoning over temporal relational structures leads to more accurate and robust sender group attribution.

While SentinelGraph achieves strong performance under the current modeling assumptions, several limitations also point to promising directions for future work. First, interaction relations in the current framework are derived through predefined abstraction rules, which discretize potentially continuous coordination intensity into a limited set of relation types. Exploring adaptive or continuous relation modeling may allow finer-grained characterization of evolving collaborative behaviors. Second, temporal reasoning currently relies on fixed snapshot granularity, which may not optimally capture coordination patterns occurring at different timescales. Although the temporal evolution patterns of sender groups are empirically and conceptually stable, extending the framework with multi-scale

or adaptive temporal modeling could further improve robustness under varying operational dynamics. Third, practical deployment faces scalability challenges from unbounded entity accumulation. Continuous operation would cause entity counts to grow significantly over time, straining memory and computational resources. Entity lifecycle management strategies, such as pruning inactive senders, hierarchical IP aggregation, and incremental graph updates, require further investigation for long-term deployment.

Author Contributions: S.W. carried out the majority of the research, including conceptualization, system implementation, experiments, data curation, analysis, visualization, and drafting the manuscript. C.T. contributed to methodology refinement, assisted with experiments, and reviewed the manuscript. M.Z. and P.X. provided supervision, resources, and feedback on manuscript revisions throughout the study. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The datasets presented in this article are not readily available because the data are derived from continuously operating honeypot systems that contain sensitive security information. Public disclosure of these datasets would compromise the operational effectiveness of our honeypot infrastructure for ongoing threat detection and analysis, and could enable adversaries to identify or evade our monitoring systems. Requests to access the datasets should be directed to the corresponding author; however, due to the sensitive nature of the data and ongoing security operations, access cannot be granted.

Acknowledgments: We would like to thank the anonymous reviewers for their valuable comments.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. NSFOCUS. *Botnet Trends 2023 Review and 2024 Predictions*; Technical Report; NSFOCUS Global: Beijing, China, 2024. Available online: <https://nsfocusglobal.com/wp-content/uploads/2024/04/Botnet-Trends-2023-Review-and-2024-Predictions.pdf> (accessed on 20 January 2025).
2. Karim, S.S.; Afzal, M.; Iqbal, W.; Al Abri, D. Advanced Persistent Threat (APT) and intrusion detection evaluation dataset for linux systems 2024. *Data Brief* **2024**, *54*, 110290.
3. CYFIRMA. *APT Quarterly Highlights: Q2 2024*; Technical Report; CYFIRMA: Singapore, 2024. Available online: <https://www.cyfirma.com/research/apt-quarterlyhighlights-q2-2024/> (accessed on 20 January 2025).
4. Cyberint. *Ransomware Groups Report 2024—Q3*; Technical Report; Cyberint: Redwood City, CA, USA, 2024. Available online: <https://e.cyberint.com/hubfs/Ransomware%20Q3%202024.pdf> (accessed on 20 January 2025).
5. Authors, M. BotDefender: A Collaborative Defense Framework Against Botnet Attacks Using Network Traffic Analysis and Machine Learning. *Arab. J. Sci. Eng.* **2023**, *49*, 3313–3329.
6. Zhang, J.; Han, D.; Lv, Z.; Lu, Y.; Duan, J.; Liu, Y.; Zhang, X. Bag2image: a multi-instance network traffic representation for network security event prediction. *Cybersecurity* **2025**, *8*, 31.
7. Prasad, N.; Diro, A.; Warren, M.; Fernando, M. A survey of cyber threat attribution: Challenges, techniques, and future directions. *Comput. Secur.* **2025**, *157*, 104606.
8. Egloff, F.J.; Smeets, M. Publicly attributing cyber attacks: a framework. *J. Strateg. Stud.* **2023**, *46*, 502–533.
9. Censys. 2015. Available online: <https://platform.censys.io/> (accessed on 15 December 2025).
10. Shodan. 2013. Available online: <https://www.shodan.io/> (accessed on 15 December 2025).
11. Erkan, U.; Toktas, F.; Toktas, A.; Lai, Q.; Zhou, S.; Lin, Y.; Gao, S. Multi-layer and multi-directional image encryption algorithm based on hyperchaotic 3D Xin-She Yang map. *Expert Syst. Appl.* **2025**, *304*, 130808.
12. Bennett, C.; Abdou, A.; van Oorschot, P.C. Empirical scanning analysis of Censys and Shodan. In Proceedings of the Workshop on Measurements, Attacks, and Defenses for the Web, Online, 25 February 2021.
13. Crochelet, P.; Neal, C.; Cuppens, N.B.; Cuppens, F. Attacker attribution via characteristics inference using honeypot data. In Proceedings of the International Conference on Network and System Security, Denarau Island, Fiji, 9–12 December 2022; pp. 155–169.

14. Wu, Y.; Huang, C.; Zhang, X.; Zhou, H. GroupTracer: Automatic attacker TTP profile extraction and group cluster in Internet of Things. *Secur. Commun. Netw.* **2020**, *2020*, 8842539.
15. Kallitsis, M.; Prajapati, R.; Honavar, V.; Wu, D.; Yen, J. Detecting and interpreting changes in scanning behavior in large network telescopes. *IEEE Trans. Inf. Forensics Secur.* **2022**, *17*, 3611–3625.
16. Shamsi, Z.; Zhang, D.; Kyoung, D.; Liu, A. Measuring and clustering network attackers using medium-interaction honeypots. In Proceedings of the 2022 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW), Genoa, Italy, 6–10 June 2022; pp. 294–306.
17. Sheng, C.; Yao, Y.; Zhao, L.; Zeng, P.; Zhao, J. Scanner-hunter: An effective ICS scanning group identification system. *IEEE Trans. Inf. Forensics Secur.* **2024**, *19*, 3077–3092.
18. Ring, M.; Dallmann, A.; Landes, D.; Hotho, A. Ip2vec: Learning similarities between ip addresses. In Proceedings of the 2017 IEEE International Conference on Data Mining Workshops (ICDMW), New Orleans, LA, USA, 18–21 November 2017; pp. 657–666.
19. Gioacchini, L.; Vassio, L.; Mellia, M.; Drago, I.; Houidi, Z.B.; Rossi, D. DarkVec: Automatic analysis of darknet traffic with word embeddings. In Proceedings of the 17th International Conference on emerging Networking EXperiments and Technologies, Munich, Germany, 7–10 December 2021; pp. 76–89.
20. Hobert, K.; Lim, C.; Budiarto, E. Enhancing cyber attribution through behavior similarity detection on linux shell honeypots with att&ck framework. In Proceedings of the 2023 IEEE International Conference on Cryptography, Informatics, and Cybersecurity (ICoCICs), Bogor, Indonesia, 22–24 August 2023; pp. 139–144.
21. Sachidananda, V.; Patil, R.; Sachdeva, A.; Lam, K.Y.; Yang, L. APTer: Towards the Investigation of APT Attribution. In Proceedings of the 2023 IEEE Conference on Dependable and Secure Computing (DSC), Tampa, FL, USA, 7–9 November 2023; pp. 1–8.
22. Griffioen, H.; Koursiounis, G.; Smaragdakis, G.; Doerr, C. Have you syn me? characterizing ten years of internet scanning. In Proceedings of the 2024 ACM on Internet Measurement Conference, Madrid, Spain, 4–6 November 2024; pp. 149–164.
23. Hiesgen, R.; Nawrocki, M.; King, A.; Dainotti, A.; Schmidt, T.C.; Wählisch, M. Spoki: Unveiling a new wave of scanners through a reactive network telescope. In Proceedings of the 31st USENIX Security Symposium (USENIX Security 22), Boston, MA, USA, 10–12 August 2022; pp. 431–448.
24. Sen, Ö.; Ivanov, B.; Kloos, C.; Zöll, C.; Lutat, P.; Henze, M.; Ulbig, A.; Andres, M. Simulation of multi-stage attack and defense mechanisms in smart grids. *Int. J. Crit. Infrastruct. Prot.* **2025**, *48*, 100727.
25. Griffioen, H.; Doerr, C. Discovering Collaboration: Unveiling Slow, Distributed Scanners based on Common Header Field Patterns. In Proceedings of the NOMS, Budapest, Hungary, 20–24 April 2020; Volume 2020, pp. 1–9.
26. Durumeric, Z.; Adrian, D.; Stephens, P.; Wustrow, E.; Halderman, J.A. Ten years of zmap. In Proceedings of the 2024 ACM on Internet Measurement Conference, Madrid, Spain, 4–6 November 2024; pp. 139–148.
27. Wu, M.; Hong, G.; Chen, J.; Liu, Q.; Tang, S.; Li, Y.; Liu, B.; Duan, H.; Yang, M. Revealing the Black Box of Device Search Engine: Scanning Assets, Strategies, and Ethical Consideration. *arXiv* **2024**, arXiv:2412.15696.
28. Almazarqi, H.A.; Woodyard, M.; Marnerides, A.K. BotPro: Data-driven Tracking & Profiling of IoT Botnets in the Wild. *Comput. Secur.* **2025**, *162*, 104778.
29. Gao, M.; Mok, R.; Carisimo, E.; Li, E.; Kulkarni, S.; claffy, k. DarkSim: A similarity-based time-series analytic framework for darknet traffic. In Proceedings of the 2024 ACM on Internet Measurement Conference, Madrid, Spain, 4–6 November 2024; pp. 241–258.
30. Bou-Harb, E.; Debbabi, M.; Assi, C. A systematic approach for detecting and clustering distributed cyber scanning. *Comput. Netw.* **2013**, *57*, 3826–3839.
31. Wang, Z.; Zhou, Y.; Liu, H.; Qiu, J.; Fang, B.; Tian, Z. ThreatInsight: Innovating Early Threat Detection Through Threat-Intelligence-Driven Analysis and Attribution. *IEEE Trans. Knowl. Data Eng.* **2024**, *36*, 9388–9402. <https://doi.org/10.1109/TKDE.2024.3474792>.
32. Priya, V.D.; Chakkaravarthy, S.S. Containerized cloud-based honeypot deception for tracking attackers. *Sci. Rep.* **2023**, *13*, 1437.
33. Rossi, E.; Chamberlain, B.; Frasca, F.; Eynard, D.; Monti, F.; Bronstein, M. Temporal graph networks for deep learning on dynamic graphs. In Proceedings of the ICML 2020 Workshop on Graph Representation Learning, Virtual, 13–18 July 2020.
34. Xu, D.; Ruan, C.; Korpeoglu, E.; Kumar, S.; Achan, K. Inductive representation learning on temporal graphs. In Proceedings of the International Conference on Learning Representations, Addis Ababa, Ethiopia, 30 April 2020.
35. Seo, Y.; Defferrard, M.; Vandergheynst, P.; Bresson, X. Structured sequence modeling with graph convolutional recurrent networks. In Proceedings of the International Conference on Neural Information Processing, Siem Reap, Cambodia, 13–16 December 2018; pp. 362–373.
36. Cong, W.; Zhang, S.; Kang, J.; Yuan, B.; Wu, H.; Zhou, X.; Tong, H.; Mahdavi, M. Do We Really Need Complicated Model Architectures For Temporal Networks? In Proceedings of the International Conference on Learning Representations, Kigali, Rwanda, 1–5 May 2023.
37. Cai, L.; Chen, Z.; Luo, C.; Gui, J.; Ni, J.; Li, D.; Chen, H. Structural temporal graph neural networks for anomaly detection in dynamic graphs. In Proceedings of the 30th ACM International Conference on Information & Knowledge Management, Virtual, 1–5 November 2021; pp. 3747–3756.

38. Wang, J.; Qiu, G.; Wu, C.; Li, X.; Lu, X. Federated learning for network attack detection using attention-based graph neural networks. *Sci. Rep.* **2024**, *14*, 19088.
39. Milajerdi, S.M.; Gjomemo, R.; Eshete, B.; Sekar, R.; Venkatakrishnan, V. Holmes: Real-time apt detection through correlation of suspicious information flows. In Proceedings of the 2019 IEEE Symposium on Security and Privacy (SP), San Francisco, CA, USA, 19–23 May 2019; pp. 1137–1152.
40. Hassan, W.U.; Bates, A.; Marino, D. Tactical provenance analysis for endpoint detection and response systems. In Proceedings of the 2020 IEEE Symposium on Security and Privacy (SP), San Francisco, CA, USA, 18–21 May 2020; pp. 1172–1189.
41. Huang, S.; Poursafaei, F.; Rabbany, R.; Rabusseau, G.; Rossi, E. UTG: Towards a unified view of snapshot and event based models for temporal graphs. *arXiv* **2024**, arXiv:2407.12269.
42. Bahar, A.A.M.; Ferrahi, K.S.; Messai, M.L.; Seba, H.; Amrouche, K. CONTINUUM: Detecting APT Attacks through Spatial-Temporal Graph Neural Networks. *arXiv* **2025**, arXiv:2501.02981.
43. Fu, C.; Li, Q.; Xu, K. Detecting unknown encrypted malicious traffic in real time via flow interaction graph analysis. *arXiv* **2023**, arXiv:2301.13686.
44. Trivedi, R.; Farajtabar, M.; Biswal, P.; Zha, H. Dyrep: Learning representations over dynamic graphs. In Proceedings of the International Conference on Learning Representations, New Orleans, LA, USA, 6–9 May 2019.
45. Jin, W.; Qu, M.; Jin, X.; Ren, X. Recurrent event network: Autoregressive structure inference over temporal knowledge graphs. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Online, 16–20 November 2020; pp. 6669–6683.
46. Schlichtkrull, M.; Kipf, T.N.; Bloem, P.; Van Den Berg, R.; Titov, I.; Welling, M. Modeling relational data with graph convolutional networks. In Proceedings of the European Semantic Web Conference, Crete, Greece, 3–7 June 2018; pp. 593–607.
47. Cho, K.; Van Merriënboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; Bengio, Y. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv* **2014**, arXiv:1406.1078.
48. Shang, C.; Tang, Y.; Huang, J.; Bi, J.; He, X.; Zhou, B. End-to-end structure-aware convolutional networks for knowledge base completion. In Proceedings of the AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019; Volume 33, pp. 3060–3067.
49. Wang, S.; Tu, C.; Zhang, Y.; Zhang, M.; Xue, P. Mapping Cyber Bot Behaviors: Understanding Payload Patterns in Honeypot Traffic. *Sensors* **2025**, *26*, 11.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.