

A Unified Knowledge-Graph Approach for Identifying APT Behaviors Across Logs and CTI Sources

1st Khaled Rahal

*Research Unit for Cyber Defence
Patrimony of the Royal Military Academy
Brussels, Belgium
k.rahal@cylab.be*

2nd Georgi Nikolov

*Research Unit for Cyber Defence
Patrimony of the Royal Military Academy
Brussels, Belgium
g.nikolov@cylab.be*

3rd Alexandre Vinovrski

*Université Libre de Bruxelles
Brussels, Belgium
Vinovrski.Alexandre@ulb.be*

4rd Arbia Riahi

*Research Unit for Cyber Defence
Patrimony of the Royal Military Academy
Brussels, Belgium
a.riahi@cylab.be*

5rd José Carresquel Vera

*Research Unit for Cyber Defence
Patrimony of the Royal Military Academy
Brussels, Belgium
j.carrasquelvera@cylab.be*

Abstract—The cyber domain expands rapidly with the continuous introduction of new techniques, tools, and systems. While this evolution benefits the management of private, public, and governmental infrastructures, it also broadens the threat landscape. Cyber Threat Intelligence (CTI) reports provide valuable insights, yet their unstructured nature and growing volume hinder effective utilization by security analysts. In this paper, we propose a unified approach leveraging Knowledge Graphs (KGs) to structure and integrate CTI with operational security logs. Our pipeline transforms unstructured CTI data into graph representations and aligns them with system logs through structural and semantic similarity analysis. This enables the identification of relationships between observed activities and known Advanced Persistent Threat (APT) behaviors. The proposed method enhances Cyber Situation Awareness (CSA) by supporting efficient data exploration and prioritizing suspicious events through similarity-based ranking. Experimental results demonstrate that our approach improves detection capabilities while reducing analyst workload, highlighting the effectiveness of graph-based representations for scalable and interpretable threat intelligence integration. The project code and dataset used is publicly available on Gitlab [1].

Index Terms—Knowledge Graph, Advanced Persistent Threat, Cyber Situation Awareness, Similarity, Data Exploration, Graph Comparison.

I. INTRODUCTION

Advanced Persistent Threats (APTs) represent some of the most challenging attacks for modern cybersecurity. Malicious actors, often state sponsored and well organised, conduct long and carefully orchestrated intrusion campaigns that can span months or even years while maintaining a remarkable level of stealth [2]. Beyond the inherent difficulty of securing and maintaining complex systems, organizations must also discover and mitigate vulnerabilities embedded in the hardware and software they rely on. APT groups frequently deploy custom-made tools, for which no signatures or known

detection methods exist. In the constantly changing cyber landscape, the idea of fully preventing large-scale cyberattacks has become increasingly unrealistic [3]. Logging systems play a central role in the Perception and Comprehension phases to gain a better Situation Awareness (SA). They are now ubiquitous across enterprise infrastructures and provide essential visibility into system activity. However, they generate overwhelming volumes of data, often comprising millions of log entries, and trigger hundreds of alerts, many of which turn out to be false positives. In environments where rapid incident response is critical, manually triaging massive and noisy information flows becomes a significant bottleneck, ultimately reducing the effectiveness of the overall mitigation strategy. The constant flow of security information and alerts can lead to alert fatigue and lower the SA of the analysts. The introduction of AI-based detection tools, while intended to improve threat identification, has added another layer of complexity. These systems inevitably produce additional false positives, and their limited explainability makes it difficult for analysts to quickly determine which alerts are trustworthy.

The work presented in this paper explores whether a formal representation of APT attacks can meaningfully improve cybersecurity analysis and investigation. Our first hypothesis is that structured modeling, particularly through graph-based representations, helps analysts better understand attack scenarios and improves detection accuracy while reducing investigation time compared to traditional approaches. A second hypothesis suggests that system logs can be transformed into this same structured representation with minimal information loss, while still preserving the semantic relationships necessary for meaningful analysis. Finally, we hypothesize that comparing formalized models of known APT campaigns with graph-based representations derived from logs enables the detection of

similar behaviors using measurable and reproducible criteria. The main contributions of this paper are fourfold:

- We introduce a CTI-to-KG transformation pipeline that extracts structured entities and relationships from threat intelligence sources such as MITRE ATT&CK and MISP.
- We design a log-to-KG modeling approach to represent system and network behaviors as graph structures capturing adversarial activities.
- We develop a CTI-log graph comparison framework that quantifies similarity between the two KGs for APT detection.
- We evaluate the proposed approach on real-world APT and benign datasets, demonstrating its effectiveness in distinguishing malicious from normal behavior.

This paper is organized as follows: Section II provides a comprehensive review of prior work. Section III presents the overall framework, covering the complete pipeline from data transformation into KGs to the final comparison stage. Section IV reports the experimental setup and evaluation results and analyzes the effectiveness, scalability, and practical relevance of the proposed approach in realistic cybersecurity environments. Finally, Section V presents our conclusions and future research directions in the domain.

II. LITERATURE REVIEW

A. Enhancing SA through Visual Analytics

Visual Analytics offers a means to enhance CSA by enabling analysts to detect and interpret system changes more effectively. Well-designed visualizations support data exploration, filtering, and reporting, significantly improving user awareness [4]. However, the hybrid nature of cyber environments poses challenges: unlike physical systems, security analysts often act as passive observers, relying on data collected by distributed tools [5]. Furthermore, sensor heterogeneity and data quality critically influence CSA [6].

To address these challenges, visualization strategies can either specialize in a specific data type or abstract heterogeneous data into generalized representations. This need becomes more pronounced when analyzing sophisticated attacks targeting critical infrastructures [7]. KGs provide a novel representation of entities and their relationships, enabling intuitive visual exploration and serving as a robust foundation for machine learning (ML) applications.

B. Detection of APTs

The increasing sophistication of cyberattacks has made their detection a critical challenge for security professionals. In [8], the authors provide a comprehensive survey of APT detection techniques, categorizing them into three groups: (i) detection by life-cycle stage, (ii) correlated detection, and (iii) detection through information flow and audit logs.

Life-cycle stage detection relies on expert knowledge of APT behavior and corresponding indicators. Commercial Intrusion Detection Systems (IDS) and Intrusion Prevention Systems (IPS) solutions typically target the initial intrusion phase, employing AI and ML to identify entry attempts [9] [8].

However, failure at this stage grants adversaries unrestricted access. Recent research addresses later stages, such as Persistence [10] [11] and Command & Control (C2) [12] [13].

Correlated detection aggregates signals from diverse sources, including network and system logs, IDS/IPS alerts, and user events, to identify coherent attack sequences. Techniques include Bayesian networks [14], Markov chains [15], SIEM-based alert correlation [16], causal graphs, multi-sensor fusion, and attack scenario reconstruction. While effective against sophisticated attacks, these methods demand significant computational resources and large APT datasets for accurate parameter tuning. Despite these challenges, they are widely adopted in commercial solutions.

III. PROPOSED APPROACH

The novelty of our approach lies in a fully integrated pipeline spanning CTI ingestion, log transformation, KG construction, and embedding generation. The system produces KGs from CTI and operational logs, then derives embeddings capturing both semantic and structural dimensions. These embeddings are compared using multiple similarity algorithms in order to evaluate behavioral proximity to documented attacks as shown in Fig. 1.

A. Transforming CTI to KG

Transforming publicly available CTI into a graph-based model begins with ingesting raw STIX 2.1 JSON files sourced from open platforms such as MITRE ATT&CK [17] and the MISP platform [18]. These files are parsed and normalized by a dedicated Python script [19], standardizing threat intelligence objects from heterogeneous sources into a uniform format.

The pipeline proceeds in three stages. First, the *Json Handler* ingests and parses the STIX files, extracting structured threat objects. Second, the parsed data is stored in a SQLite database, which serves as an intermediate relational layer consolidating CTI records alongside the dataset database. Third, this structured dataset is consumed by the Neo4j pipeline, which maps entities and relationships into graph nodes and edges, yielding the KG used throughout the framework for embedding generation and similarity-based comparison.

This three-stage design cleanly separates ingestion, storage, and graph construction, ensuring that CTI from heterogeneous sources is consistently normalized before being represented as a graph, and providing a reproducible and extensible foundation for both statistical and graph-based threat analysis. By transforming CTI reports into a graphical representation, it is easier to view the various relations between the different entities and follow the chain of events that represent the Cyber Kill Chain [20].

We also include the tools used by the APT groups, as documented in the SolarWinds campaign [21] and Operation Ghost [22], based on their respective CTI reports cross-referenced with MITRE ATT&CK. By further integrating IoCs from the MISP platform, the KG is enriched with concrete operational artifacts.

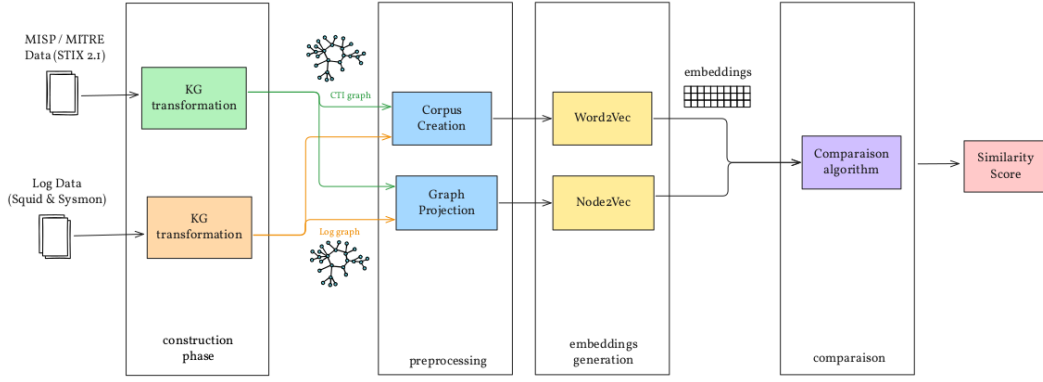


Fig. 1. Overall Architecture of the Proposed Framework

Incorporating IoCs is essential, as they bridge high-level threat knowledge with the low-level observables found in security logs, such as IP addresses, port numbers, protocols, and resource usage metrics. Directly linking these indicators to known TTPs enables more precise detection of suspicious behavior and strengthens the overall cybersecurity analysis.

B. APT Datasets to KG

We employ two complementary datasets to evaluate our framework. The first is a purpose-built laboratory dataset, constructed over 28 days of data collection, to address the scarcity of labeled APT telemetry [23]. It deliberately focuses on persistence mechanisms and C2 activity, two behavioral dimensions central to this work. This dataset was generated from realistic simulated APT scenarios reproducing the trade-craft of groups such as APT41, APT29, and APT28, using adversary simulation frameworks including Cobalt Strike and PowerShell Empire, both post-exploitation C2 frameworks, as well as CALDERA for automated adversary emulation. In total, the dataset comprises 527,301 OS Sysmon log entries and 24,450 proxy log entries, hereafter referred to as APT-Datasets. The generated dataset is publicly available on the framework’s GitLab repository [1].

The second dataset, denoted Benign-Datasets, consists of benign network and endpoint telemetry drawn from LMD-2022, comprising approximately 80,000 events [24], and provides a representative baseline for normal system behavior. To rigorously evaluate graph construction under adversarial telemetry, we focus on two complementary and behaviorally grounded data sources: *Sysmon* endpoint logs and *Squid* proxy logs. For Sysmon logs, we construct the graph using *ProcessGuid*, *ProcessId*, and process names as primary entity identifiers, enabling consistent representation of process execution chains across log entries. For Squid logs, IP addresses, domain names, and URLs constitute the core network entities. A key design decision is the *cross-source correlation mechanism*, which links endpoint and network perspectives through three complementary event types: Sysmon Event ID 3 (NetworkConnect), which captures outbound TCP/UDP connection attempts at the process level; Sysmon

Event ID 22 (DNSEvent), which records DNS query activity and resolved hostnames attributed to initiating processes; and Squid HTTP access entries, which log proxied web requests at the application layer. Together, these sources establish a principled, multi-layer alignment between process-level execution context and external communication behavior, preserving the full causal chain from host-side activity through name resolution to outbound C2 traffic.

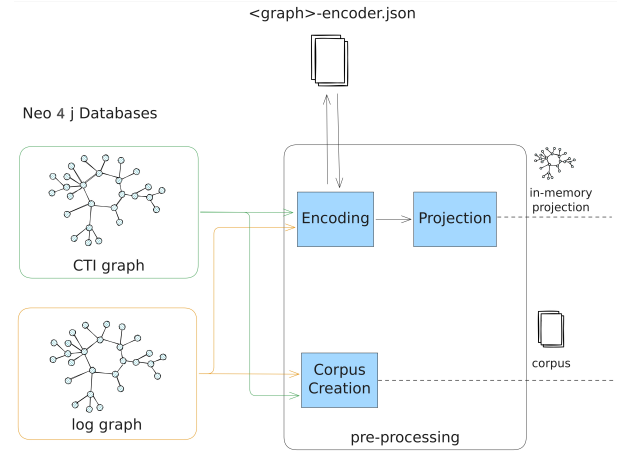


Fig. 2. Pipeline for converting CTI and log-derived graphs into in-memory projections for Node2Vec and a corpus for Word2Vec.

C. Graph Encoding and Projection Strategy

As illustrated in Fig. 2, the strategy is organized around three core components: **Corpus Creation**, **Encoding**, and **Projection**. Together, these stages convert heterogeneous graph data into a suitable form for large-scale embedding generation and cross-graph analysis. The encoding stage extracts all non-numeric node properties and assigns each a unique integer identifier within a global dictionary. These encoded values are then reinjected into their respective nodes under standardized, slightly modified property names to distinguish them from the original attributes.

After the encoding step, the transformed properties are written back into Neo4j using a clear and consistent naming

scheme, ensuring that each encoded attribute remains easy to track and interpret. With the graph now fully numeric and structurally coherent, Neo4j’s native GDS projection can be applied directly without any additional preprocessing [25]. This seamless transition enables fast, scalable construction of in-memory graph projections for both the CTI and log datasets, laying the groundwork for efficient embedding generation and downstream similarity analysis.

D. Embedding Generation

Embedding generation is a core component of the proposed framework, enabling graph comparison in a continuous vector space rather than at the individual entity level. To capture both structural and semantic properties, we employ two complementary strategies: Node2Vec and Word2Vec.

1) *Node2Vec: Embedding Heterogeneous Threat Graphs:* Node2Vec operates directly on the graph topology through biased random walks, requiring no node feature vectors. This makes it particularly suited to our setting, where the KG is built from heterogeneous log sources and node attributes may be incomplete or absent.

Node2Vec is a framework that treats graph traversal as a guided narrative [26], uncovering latent structural patterns within projected graphs. Unlike static topological descriptors, Node2Vec performs biased random walks that “read” the graph, converting complex connectivity into sequence-based representations suitable for downstream ML models.

Fig. 3 illustrates a representative directed graph constructed from system telemetry. Nodes represent heterogeneous entities including processes, files, domains, IPs, hostnames, user accounts, and indicators, while edges encode operational or adversarial relationships such as *EXECUTED BY*, *CONNECTED TO*, and *RESOLVE TO* as shown in Table I. This rich semantic structure forms the foundation for Node2Vec’s walk-based sampling, enabling the model to capture both behavioral proximity and structural equivalence.

TABLE I
FORMAL SCHEMA OF NODE TYPES AND EDGE TYPES IN THE
CONSTRUCTED KG.

Type	Name	Description
Node	process	System process identified by GUID/PID/image path
	file	Artifacts identified by image path and hash
	user_account	User or service account executing a process
	network_traffic	Network event capturing protocol/ports/timestamp
	ip	Source or destination IP address observed in logs
	domain_name	DNS domain name extracted from proxy traffic
	url	Full URL observed in proxy logs
Edge	indicator	MITRE ATT&CK technique referenced by a Sysmon rule
	EXECUTES	Process executes a file
	EXECUTED_BY	Process is executed by a user account
	CHILD_OF	Process is spawned by a parent process
	CONNECT_TO	Process or network event initiates a connection
	SRC_REF	Network event references a source IP address
	DST_REF	Network event references a destination IP address
	RESOLVE_TO	Domain name resolves to an IP address
	CONTAINS	URL contains a domain name
	REFERENCE	Process references a known ATT&CK indicator

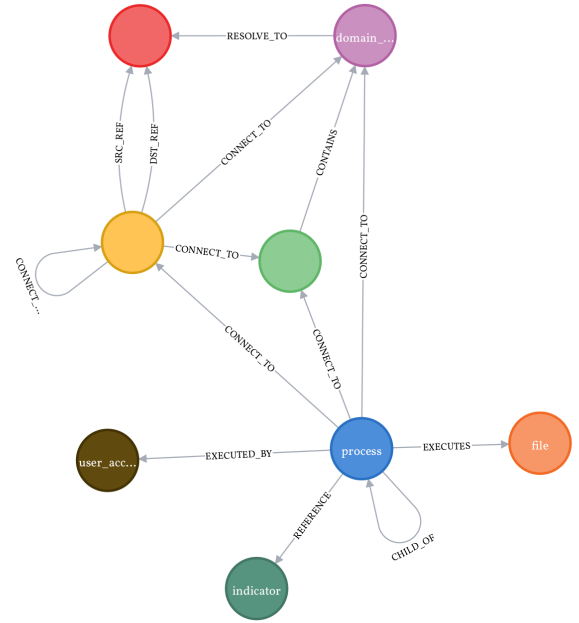


Fig. 3. Directed threat graph constructed from system logs, showing heterogeneous entities and labeled relationships.

2) *Word2Vec [27]: Semantic Graph Representation:* Graph paths are linearized into sentence-like sequences, where each entity (process, file, registry key, network connection) and each relationship (e.g., *launches*, *writes*, *connects_to*) is treated as an ordered token. Word2Vec complements graph traversal done by Node2Vec, by embedding sequential behavioral traces, such as command-line arguments and process execution chains, where the skip-gram model effectively captures co-occurrence patterns characteristic of adversarial tradecraft.

3) *Cosine Similarity:* a widely used metric for quantifying the alignment between two non-zero vectors in an inner-product space [28]. Rather than comparing magnitudes, it captures the angular closeness of vectors, making it notably suitable for embedding-based semantic analysis.

In our setting, we compute cosine similarity directly from the normalized dot product of embedding pairs.

IV. EXPERIMENTATION

This section presents a pairwise similarity among embedding representations in latent vector space, without performing supervised detection or classification. No ground-truth labels are available to indicate which behaviors should be separated or assigned to predefined categories. The analysis instead focuses on the relative proximity of embeddings, allowing structurally and behaviorally coherent patterns to be identified through clustering and neighborhood relationships.

For Node2Vec, cosine similarity is the most suitable comparison metric, as the learned representations are intrinsically dependent on graph topology and are generated with respect to the node set and embedding dimensionality of a given graph instance. By contrast, Word2Vec embeddings are produced in a fixed-dimensional continuous vector space, which permits the

use of a broader set of similarity and association measures. In addition to cosine similarity, these representations can be evaluated using normalized Euclidean distance, Pearson correlation, and Spearman rank correlation [29], each capturing complementary aspects of vector alignment, magnitude variation, or rank-order consistency.

A. Experiment 1: Comparative Embedding Similarity Between CTI and APT-Datasets

As outlined in Section III-D, we first evaluate the baseline parameter configurations for both embedding models. The objective of this experiment is to compare how effectively Word2Vec and Node2Vec capture semantic and structural similarities between CTI graphs and APT-Datasets. Table II summarizes the resulting similarity scores and highlights the performance differences observed under the selected configurations. The results indicate that Word2Vec achieves higher cosine similarity scores than Node2Vec, suggesting stronger alignment in terms of contextual semantics, whereas Node2Vec captures graph topology more conservatively.

TABLE II
SIMILARITY METRICS BETWEEN APT-DATASETS AND CTI EMBEDDINGS
USING NODE2VEC AND WORD2VEC.

Similarity Metric	Node2Vec	Word2Vec
Cosine Similarity	0.4175	0.5292
Cosine Similarity (chunked)	0.4175	0.5292
Euclidean Similarity	–	0.0386
Pearson Correlation	–	0.3737
Spearman Correlation	–	0.3837

B. Experiment 2: Comparative Embedding Similarity Between CTI and APT-Datasets and Benign Datasets

In the second experiment, we introduce an additional benign dataset, as mention in section III-B to complement the existing CTI and APT-Datasets. This extension allows us to evaluate the behavior of both embedding models under a more realistic and balanced setting that includes normal system activity alongside malicious traces. Word2Vec and Node2Vec are then systematically tested with different parameter configurations.

As shown in Table III, the semantic analysis conducted using Word2Vec demonstrates a strong capacity to discriminate between benign activity and behavior associated with APT. Cosine similarity values between log-derived embeddings and CTI-derived embeddings are consistently elevated for infected datasets, while remaining near zero for benign datasets. This clear separation indicates that Word2Vec effectively captures semantically meaningful relationships between operational log entities such as processes, domains, file system artifacts, and STIX-encoded threat indicators and the behavioral patterns documented in CTI repositories. From an operational detection standpoint, this behavior confirms that semantic similarity constitutes a robust and actionable signal for identifying activity aligned with known APT tradecraft.

As illustrated in Fig. 4, the optimal configuration for Word2Vec is obtained with a vector dimension of 400, trained

TABLE III
COMPARISON OF SEMANTIC (WORD2VEC) AND STRUCTURAL (NODE2VEC) SIMILARITY METRICS FOR EMBEDDINGS ON APT-INFECTED AND BENIGN DATASETS.

Dataset Type	Word2Vec (Cosine)	Node2Vec (Cosine)
APT Dataset	0.55	0.75
Benign dataset	0.06	0.75

for 7 epochs and using a window size of 6. This configuration consistently yields the highest similarity scores across all evaluated metrics, reinforcing its suitability for the semantic component of the framework.

Beyond the statistical properties of the similarity metrics, the performance results highlight a pronounced operational contrast between the two embedding strategies. Word2Vec remains computationally lightweight, completing the full similarity computation in approximately 1.62 seconds. Its low memory footprint and rapid execution make it well suited for large-scale or near-real-time detection pipelines, where responsiveness and efficiency are critical.

Node2Vec, in contrast, imposes a substantially higher computational burden. Its reliance on biased random walks and repeated graph traversals results in extended processing times, elevated memory consumption, and significant CPU load. These characteristics raise legitimate concerns regarding its scalability and practical deployment in production environments, particularly those handling high-volume telemetry or requiring continuous monitoring. Consequently, within this framework, the structural embedding component is best positioned as a supplementary validation mechanism rather than a primary detection signal.

V. CONCLUSION AND FUTURE WORK

Our work presents a unified framework for comparing CTI knowledge graphs with system log-derived graphs, detecting adversarial behavior through semantic and structural similarity. By encoding heterogeneous CTI and telemetry sources into consistent numerical representations, the framework enables direct graph-level comparison, eliminating the dependency on exact indicator matching and manual rule engineering. The combination of Word2Vec and Node2Vec captures both contextual relationships among threat entities and latent structural patterns within attack graphs, effectively bridging high-level threat intelligence with low-level system activity. The results prove that semantic graph comparison is a necessary condition for reliable APT detection, given the inherently context-dependent nature of sophisticated adversarial behavior.

Several directions remain open for future work. We aim to extending the framework toward multi-stage APT modeling, through temporal graph segmentation and sliding-window embeddings, may enable the capture of transitions across attack lifecycle phases. Extending detection from initial compromise through persistence establishment to outbound command-and-control activity offers support for earlier detection of APT campaigns based on partial behavioral overlap with CTI-derived graph representations.

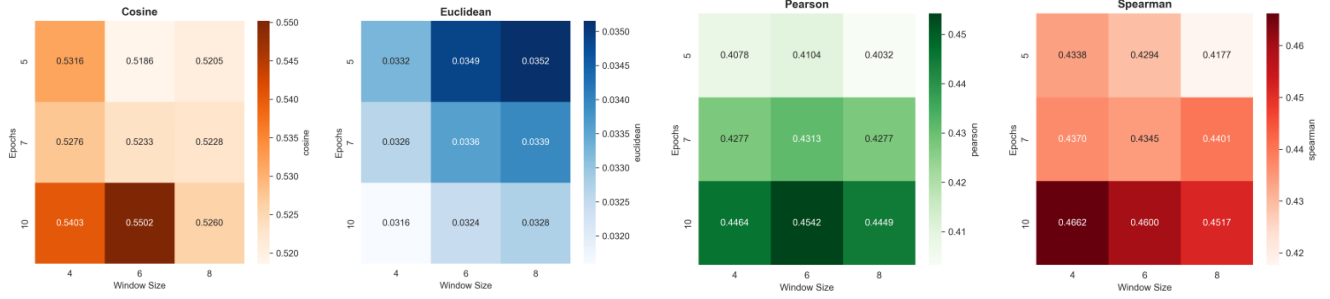


Fig. 4. Similarity heatmaps for Word2Vec metrics.

ON THE USE OF AI IN THIS PAPER.

All content presented in this paper is the original work of the authors. Large Language Models (LLMs) were used exclusively for minor spelling corrections and for restructuring sentences to improve readability and alignment with scientific writing standards. No generative content or substantive contributions were produced by LLMs.

REFERENCES

- [1] "A unified knowledge-graph approach for identifying apt behaviors across logs and cti sources," <https://gitlab.cylab.be/cylab/graphctivlogscomparison>.
- [2] A. Alshamrani, S. Myneni, A. Chowdhary, and D. Huang, "A survey on advanced persistent threats: Techniques, solutions, challenges, and research opportunities," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 2, pp. 1851–1877, 2019.
- [3] M. S. d. Araujo, B. A. S. Machado, and F. U. Passos, "Resilience in the context of cyber security: A review of the fundamental concepts and relevance," *Applied Sciences*, vol. 14, no. 5, p. 2116, 2024.
- [4] M. Varga, C. Winkelholz, S. Träber-Burdin, P. Bivall, and K. Kullman, "Chapter 7 cyber situation awareness, exploratory visual analytics," NATO, Tech. Rep. 10.14339/STO-TR-IST-141, February 2023.
- [5] M. Husák, T. Jirsák, and S. J. Yang, "Sok: Contemporary issues and challenges to enable cyber situational awareness for network security," in *Proceedings of the 15th International Conference on Availability, Reliability and Security*, 2020, pp. 1–10.
- [6] H. Alavizadeh, J. Jang-Jaccard, S. Y. Enoch, H. Al-Sahaf, I. Welch, S. A. Camtepe, and D. S. Kim, "A survey on threat situation awareness systems: Framework, techniques, and insights," *arXiv preprint arXiv:2110.15747*, 2021.
- [7] G. Nikolov, M. Varga, A. R. Panganiban, K. Kullman, and V. Lavigne, "Enhancing cyber situation awareness: Visualizing advanced persistent threats as complex systems," in *International Conference on Availability, Reliability and Security*. Springer, 2025, pp. 90–107.
- [8] S. Krishnapriya and S. Singh, "A comprehensive survey on advanced persistent threat (apt) detection techniques," *Computers, Materials & Continua*, vol. 80, no. 2, 2024.
- [9] S. Singh, P. K. Sharma, S. Y. Moon, D. Moon, and J. H. Park, "A comprehensive study on apt attacks and countermeasures for future networks and communications: challenges and solutions," *The Journal of Supercomputing*, vol. 75, no. 8, pp. 4543–4574, 2019.
- [10] Q. Liu, M. Shoaib, M. U. Rehman, K. Bao, V. Hagenmeyer, and W. U. Hassan, "Accurate and scalable detection and investigation of cyber persistence threats," 2024. [Online]. Available: <https://arxiv.org/abs/2407.18832>
- [11] K. Rahal, A. Riahi, G. Nikolov, T. Debatty, and J.-M. Dricot, "Uncovering malicious persistence: Machine learning-based detection of windows scheduled tasks," *Adv. Know. Base. Syst. Data Sci. Cyber.*, vol. 2, no. 3, pp. 310–336, 2025.
- [12] G. Nikolov and W. Mees, "Detection of previously unknown advanced persistent threats through visual analytics with the masfad framework," in *2023 International Conference on Military Communications and Information Systems (ICMCIS)*, 2023, pp. 1–10.
- [13] R. Z. Haider, B. Aslam, H. Abbas, and Z. Iqbal, "C2-dnswatch: Endpoint framework for detecting command and control (c2) connection of advanced persistent threats (apts)," in *2024 13th International Conference on Communications, Circuits and Systems (ICCCAS)*. IEEE, 2024, pp. 64–69.
- [14] Y. Kim, I. Lee, H. Kwon, K. Lee, and J. Yoon, "Ban: Predicting apt attack based on bayesian network with mitre att&ck framework," *IEEE Access*, vol. 11, pp. 91 949–91 968, 2023.
- [15] G. Ioannou, P. Louvieris, N. Clewley, and G. Powell, "A markov multi-phase transferable belief model: An application for predicting data exfiltration apts," in *Proceedings of the 16th International Conference on Information Fusion*. IEEE, 2013, pp. 842–849.
- [16] A. AwangLah, R. Akmal, and M. Hadri, "Proposed framework for network lateral movement detection based on user risk scoring in siem," in *2018 2nd International Conference on Telematics and Future Generation Networks (TAFGEN)*, pp. 149–154.
- [17] (2024) Mitre att&ck. [Online]. Available: <https://attack.mitre.org/>
- [18] C. Wagner, A. Dulaunoy, G. Wagener, and A. Iklody, "Misp: The design and implementation of a collaborative threat intelligence sharing platform," in *Proceedings of the 2016 ACM on workshop on information sharing and collaborative security*, 2016, pp. 49–56.
- [19] "A unified knowledge-graph approach for identifying apt behaviors across logs and cti sources," <https://gitlab.cylab.be/cylab/mitreattack-dataset-to-json>.
- [20] T. Yadav and A. M. Rao, "Technical aspects of cyber kill chain," in *International symposium on security in computing and communication*. Springer, 2015, pp. 438–452.
- [21] E. D. Wolff, K. M. GroWIEy, M. O. Lerner, M. B. Welling, M. G. Gruden, and J. Canter, "Navigating the solarwinds supply chain attack," *Procurement Law.*, vol. 56, p. 3, 2021.
- [22] M. Faou, M. Tartare, and T. Dupuy, "Operation ghost," *ESET Research White papers, October 2019*, 2019.
- [23] Q. Liu, M. Shoaib, M. U. Rehman, K. Bao, V. Hagenmeyer, and W. U. Hassan, "Accurate and scalable detection and investigation of cyber persistence threats," 2024. [Online]. Available: <https://arxiv.org/abs/2407.18832>
- [24] C. Smiliotopoulos, G. Kambourakis, and K. Barmatsalou, "On the detection of lateral movement through supervised machine learning and an open-source tool to create turnkey datasets from sysmon logs," 04 2023.
- [25] S. Kolla, "Neo4j graph data science (gds) library: Advanced analytics on connected data," *INTERNATIONAL JOURNAL OF ADVANCED RESEARCH IN ENGINEERING & TECHNOLOGY*, vol. 11, pp. 1077–1086, 08 2020.
- [26] A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks," 2016. [Online]. Available: <https://arxiv.org/abs/1607.00653>
- [27] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," 2013. [Online]. Available: <http://arxiv.org/abs/1310.4546>
- [28] S. Quer, T. Madeo, A. Calabrese, G. Squillero, and E. Carraro, "Node embedding and cosine similarity for efficient maximum common subgraph discovery," *Applied Sciences*, vol. 15, no. 16, 2025. [Online]. Available: <https://www.mdpi.com/2076-3417/15/16/8920>
- [29] D. Koutra, A. Parikh, A. Ramdas, and J. Xiang, "Algorithms for graph similarity and subgraph matching," in *Proc. Ecol. inference conf*, vol. 17, 2011, pp. 1–50.