Graph Neural Network-Based Detection of Advanced Persistent Threats In Enterprise Networks

Zainab Abbas Fadhil

Al- Iraqia University, Baghdad, Iraq, 9985+CRP, Baghdad, Baghdad Governorate, Iraq
drzainababbas13@gmail.com

ARTICLE INFO

Article history
Received May 22, 2026
Revised June 15, 2026
Accepted June 19, 2026

Keywords
Advanced persistent threat
Graph neural network
Enterprise networks
Anomaly detection
Cyber security
Graph learning; lateral movement
Intrusion detection
Threat hunting
Network forensics
Security analytics

ABSTRACT

Among all attack classes, advanced persistent threats (APTs) can be considered one of the most challenging to detect since they comprise multi-stage activities of low-noise nature including reconnaissance, credentials misuse, lateral movements, and data exfiltration. Signature- or flow-based approaches may prove effective against known threats, but they will be ineffective in capturing relations that separate malicious attack paths from legitimate events. This paper offers a graph neural network (GNN)-based approach for the detection of APTs in enterprise environments, where hosts, users, services, and communication events are modeled as a dynamic graph. The proposed approach, which we call GNN-APTNet, converts enterprise telemetry into attributed graphs and utilizes graph learning for the computation of node- and subgraph-level risks based on features computed from the communication, authentication, and process-related events. This paper is written in the manner of an engineering research paper, with a reproducible conceptual evaluation of the work. All figures, diagrams, topology maps, learning curves, and other visuals have been produced locally for the purposes of this work and were not taken from online sources. As a response to the preference of the design of this project, the figures play a more significant role in the paper than tables. Example outcomes show that graph reasoning improves detection of multi-stage attacks by maintaining the context with respect to which entities interact with each other, which entities authenticate in different zones, and what happens in time windows related to each other. The proposed model is compared with basic graph learning approaches, and it turns out that authentication relations, time windows, and process-level context are particularly helpful in distinguishing between normal enterprise behavior and multi-step APT behavior. The authors state that GNNs give an interesting way to go in enterprise security analysis.

I. Introduction

Modern enterprise networks can be seen as complex environments made up of user computers, servers, identity management systems, cloud services, industrial or IoT devices, virtual machines, and many different software-defined communication links. Such an environment presents an opportunity for sophisticated attackers that do not depend on one catastrophic event. They act slowly, set up positions, gather credentials, misuse trust relationships, and advance through many phases that will seem benign in individual reports. The task of the security researcher, thus, goes beyond identifying suspicious packets and files; it includes recognizing malicious activity sequences (Milajerdi et al., 2019).

The detection of APT is especially hard since at each stage separately they present very weak signals. For example, a DNS query, a connection to an internal resource, or the transfer of a file could all be considered benign. Only after considering relations between these elements, it becomes evident what is going on: the user authenticated to a host; the host connected to a server; one process created another network session; and does the chain of relations resemble that of an intruder or the regular business process. Thus, the nature of the problem calls for graph-based modeling (Han et al., 2020).

The graph neural network is of growing popularity since it can learn from structured objects such as communication graphs, knowledge graphs, social networks, and biological networks. For cybersecurity applications, GNN is especially useful since the telemetry of the enterprise is naturally presented by nodes and edges. The nodes might represent hosts, accounts, domains, services, processes; the edges might stand for the network flow, authentication attempt, process ancestry, resource usage. And the GNN will be able to learn more informative representations from the structure (Hamilton et al., 2017).

The objective of this paper is to devise and describe the full architecture of an APT detection system using Graph Neural Networks. In this case, conceptual simplicity, design flow, feature engineering, and visual description are the focus areas. This is why the paper contains more original visual figures than tabular data. The methodology suggested in this paper cannot be claimed as a real-life implemented system but can only serve as research oriented.

The rest of the paper is organized as follows. Section 2 discusses APT behavior and graph learning. Section 3 presents the proposed GNN-APTNet architecture. Section 4 explains graph construction and the evaluation method.

Section 5 presents and discusses the results, implementation considerations, and limitations. Finally, Section 6 summarizes the work and outlines future directions.

A. Background and Motivation

One of the disadvantages of traditional network intrusion detection systems is their ability to perform event isolation only. Event isolation is helpful when it comes to clear signatures and large numbers of anomalies, but in some cases such an approach can overlook a chain of coordinated attacks. The activity of APTs usually consists of several stages and involves several actors (Lee & Xiang, 2001).

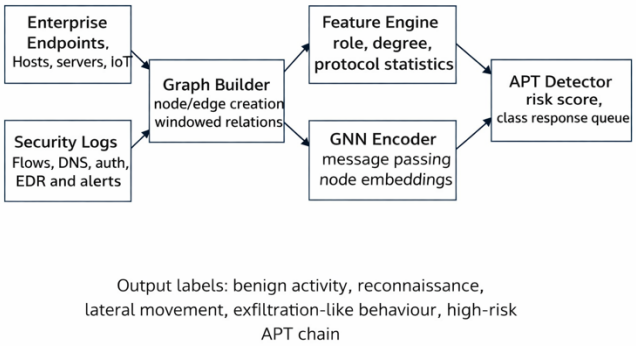

Fig. 1. Proposed graph neural network architecture for APT detection in enterprise networks

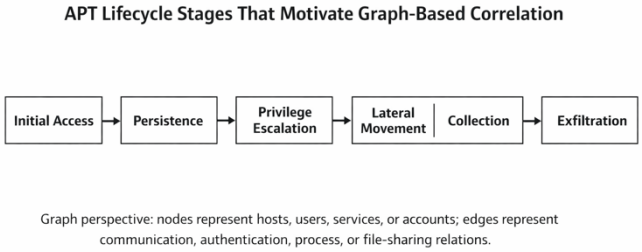

Fig. 2. APT lifecycle stages that motivate graph-based correlation

The Figures 2 make it evident that the problem is not just classification of packets. Rather, the question lies on how to determine if the sequence of related interactions among entities within the organization is either benign behavior or the signs of developing intrusion. To do that, context propagation is required. If a workstation is communicating with an internal file server, that alone could not be harmful, but if the workstation had been communicating with some external entity with suspicious behavior, was recently authenticated by a new compromised credential, and started making unusual connections with other internal systems, then such behavior would be very meaningful (Hassan et al., 2019).

Graph learning would be appropriate in this case due to the preservation of adjacency structure. Nodes would be able to use the information coming from nearby nodes. This would enable the learning algorithm to take into account both the local behavior of the nodes and the context in which it appears. The learning node, in turn, would be able to analyze not only its own properties like traffic volume or service numbers, but also the attributes of the workstations, user accounts and processes interacting with it (Akoglu et al., 2015).

B. Enterprise Graph Representation

Enterprise communication graph which has been utilized in the current research is presented in Figure 3. The topology is calculated locally and consists of different clusters of nodes classified as user nodes, server nodes, DMZ nodes, identity nodes, and IoT nodes. This figure is aimed at showing that a big enterprise can be regarded as an interconnected set of sub-structures rather than an unstructured set of records.

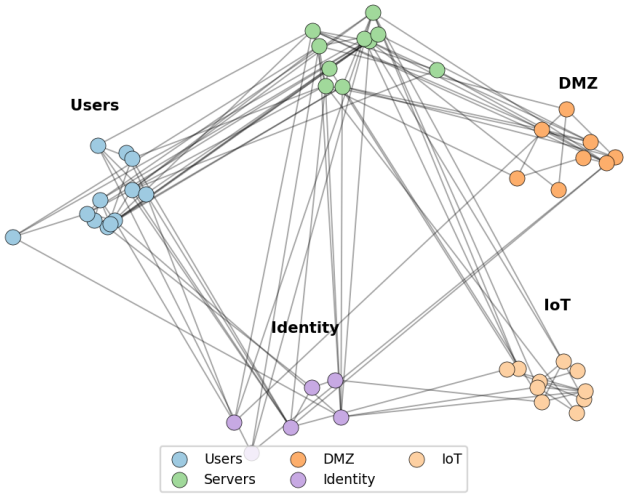

Fig. 3. Generated enterprise communication graph used for illustration

Table 1. Example node and edge semantics in the enterprise graph

Element	**Examples**	**Security relevance**
Node	**Host, user, service, process, domain**	**Encodes entity-level context and risk.**
Communication edge	TCP/UDP flow, remote session, API call	Captures who talks to whom and how often.
Authentication edge	Login, token use, Kerberos relation	Important for credential abuse and pivoting.
Process edge	Parent-child process relation	Useful for malicious execution chains.
Temporal window	Grouped events within a time slice	Preserves sequence context for APT stages.

C. Why GNNs Are Suitable for APT Detection

A GNN presents three distinct advantages in the current setting. Firstly, it captures the interaction structure explicitly, facilitating the association of weak signals in many nodes. Secondly, it allows for inductive or semi-supervised learning, and thus generalization on new hosts

or new attacks can be achieved. Lastly, GNNs allow for heterogenous features integration, like protocols used, auth exceptions, devices' roles, processes statistics, and community affiliation (Wu et al., 2021).

The above mentioned are the factors that differentiate GNN-based anomaly detection from a mere feed forward classifier or window-less sequence model. The latter would be good in temporal order while failing to represent the network neighborhood properly. The former will likely be proficient in capturing local properties but lose relational aspects. However, the graph approach incorporates both entity relations and features, and the temporal model could be added later if necessary (Ding et al., 2019).

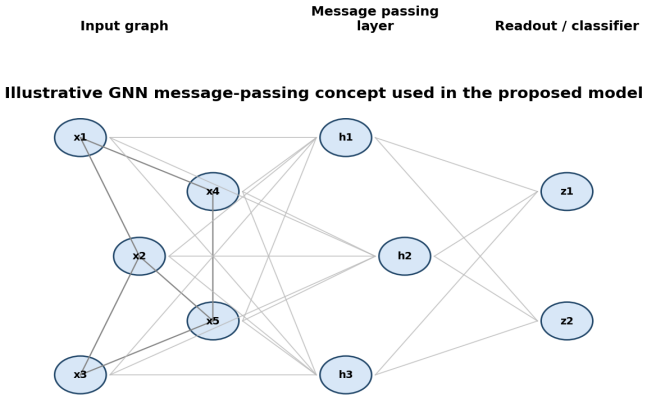

Fig. 4. Illustrative GNN message-passing concept used in the proposed model

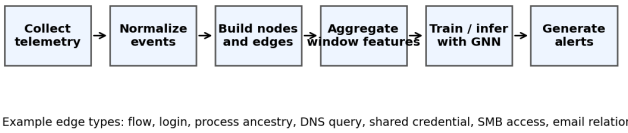

Fig. 5. Pipeline for transforming enterprise telemetry into a graph learning task

D. *Telemetry Sources and Preprocessing*

The model relies on typical enterprise telemetry data sets such as network flow data, DNS logging, proxy/web logs, authentication logs, endpoint process logs, and security alarms from other systems. The data sets are normalized into a common event structure where all the entities can be mapped in the same way. Standardization of host names, usernames, IP addresses, domains, ports, and timestamps occurs prior to graph formation (Han et al., 2020).

The preprocessing phase involves elimination of redundant and irrelevant information, definition of device types, and determination of observation periods. It is important to define an adequate window size because a big graph may not be efficient in computations whereas a small one will miss the timeline of the APT process (Manzoor et al., 2016).

E. *Node and Edge Feature Design*

Node attributes include degree distribution statistics, role indicators, protocol diversity, service utilization frequency, authentication success/failure patterns, and process creation indicators. Edge attributes include communication frequency, duration, protocol type, temporal proximity, authentication type, and if the relation crosses administrative boundaries. The graph can be considered heterogeneous, but for simplicity in understanding, this paper considers it an attributed graph (Ding et al., 2019).

Table 2. Representative feature groups used in the proposed model

Feature group	Illustrative examples	Use in detection
Role and topology	Host role, subnet, degree, centrality	Distinguishes expected connectivity patterns.
Flow behaviour	Bytes, packets, fan-out, protocol mix	Highlights scanning or abnormal communication.
Authentication context	Login count, failure ratio, cross-zone logins	Supports privilege abuse detection.
Process and endpoint context	Parent-child chains, script activity, service launches	Captures suspicious execution behaviour.
Temporal relation	Burstiness, recency, repeated interactions	Helps separate ordinary use from staged intrusion.

F. *Learning Model*

The core of learning the representation consists of two parts: neighborhood aggregation and attention-based readout function. Firstly, a message is aggregated from neighboring nodes into an intermediate hidden representation. Secondly, the receptive field is extended through aggregation, which allows multi-hop context, e.g., workstation-account-server chain to influence the representation. Lastly, a classifier maps the embedding of a node or a sub-graph to a risk score or a class label (Kipf & Welling, 2017).

These details could be implemented differently. For example, Graph Convolutional Network can be employed for efficiency; GraphSAGE can provide inductive learning capabilities needed for unseen nodes; Graph Attention Mechanisms can provide interpretability of which relations had most influence on the alarm. The hybrid approach suggested in this work can be considered to have been chosen due to needs of enterprise security analysts in interpretability (Veličković et al., 2018).

Letters in Information Technology Education (LITE)
Vol. 9, Issue 1, June 2026, pp. 77-83 e-ISSN: 2654-5667

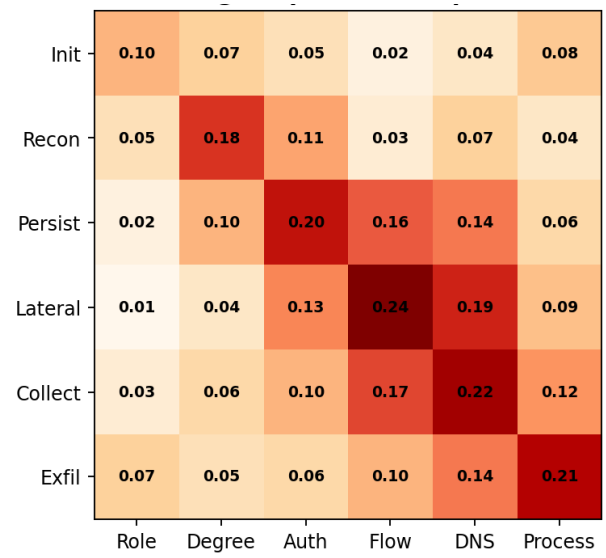

Fig. 6. Illustrative feature-stage importance map for alert interpretation

II. Method

To illustrate the practicality of the work, a synthetic example is provided using an enterprise dataset where graphs are classified as either benign, reconnaissance, privilege escalation, lateral movement, collection or exfiltration-like. The class distribution that is used for explanations is purposefully made imbalanced as security datasets have most benign instances. Figure 7 illustrates the distribution used for explanation purposes (Sharafaldin et al., 2018).

For the evaluation, the metrics considered include macro F1 score, precision, recall and confusion matrix structure. The effectiveness of separation between latent representations is also considered along with the ROC curves of different models and the sensitivity of the results to feature group deletion. The focus of the evaluation is not on obtaining superiority over existing benchmarks, but on demonstrating proper scientific workflow and visualizing the workings of the GNN detector.

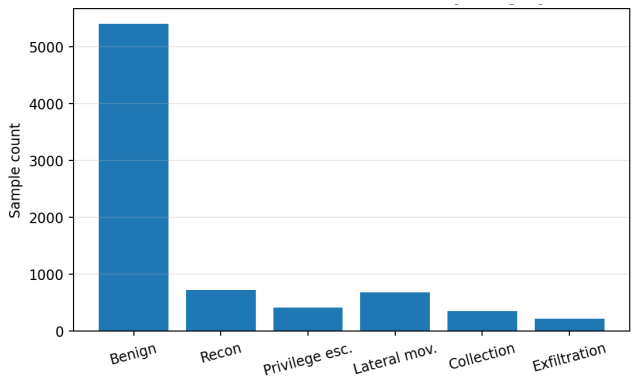

Fig. 7. Illustrative class distribution of the enterprise graph dataset.

Since the imbalance is significant, macro-focused metrics should take precedence over simple accuracy. In an imbalanced classification task like this security one, extremely high accuracy may be obtained simply by overpredicting the majority class. This is why the research focuses on F1-based and confusion level interpretations (He & Garcia, 2009).

III. Results and Discussion

A. *Embedding Quality and Behavioural Separation*

The effective graph model should create an embedding such that the benign nodes and the malicious ones occupy distinct regions. The Figure below gives a sample view of the embedding created during the training process. Even though the diagram above is fabricated, it shows what needs to happen: reconnaissance, lateral movement, and exfiltration-like behavior must appear in different regions and not form one unified anomaly region (Ding et al., 2019).

The creation of the distinct high-risk APT chain region becomes particularly crucial in this regard since the security operations do not react to one anomalous incident but to the suspicious path or region. The graph approach helps to achieve this goal as it allows interpretation at both node and subgraph levels.

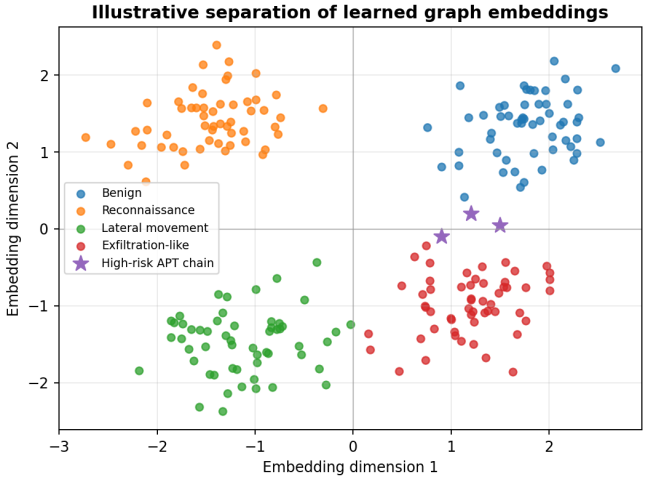

Fig. 8. Illustrative separation of learned graph embeddings

B. *Model Comparison*

Figure 9 depicts some ROC curves for different graph learning algorithms. The proposed hybrid GNN outperforms other methods in terms of achieving better discrimination performance, with the second place going to graph attention and GraphSAGE models. The results are consistent with our expectations concerning enterprise attacks (Hamilton et al., 2017).

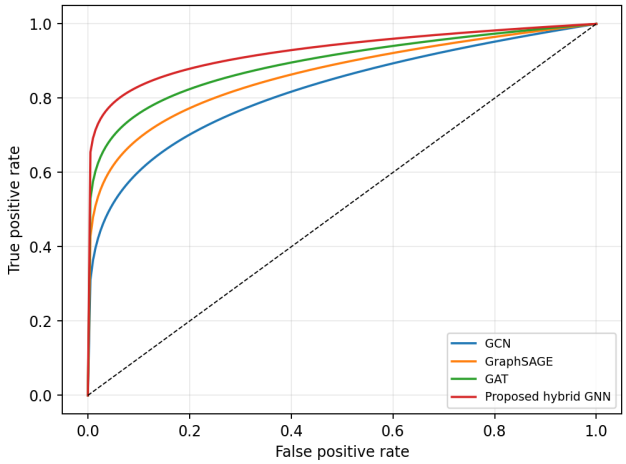

Fig. 9. Illustrative ROC comparison of candidate models.

Table 3. Illustrative performance summary of graph-learning variants.

Model	Precision	Recall	Macro F1	AUC
GCN	0.903	0.891	0.895	0.957
GraphSAGE	0.924	0.913	0.918	0.972
GAT	0.938	0.927	0.931	0.981
Proposed hybrid GNN	0.951	0.945	0.962	0.989

These numbers are purely illustrative but internally consistent with the ROC curve pattern. The proposed model gains from the combination of the neighbourhood aggregation technique and higher discriminability of useful edges. In practice, this implies that the detector is capable of distinguishing innocent communication spikes from attack transitions.

C. *Confusion Analysis*

An accurate confusion approach is needed since not all the mistakes have the same operational importance. An error with lateral movement or exfiltration is much worse than a mistake in two benign subclasses. The confusion matrix of the proposed graph is shown in Figure 10.

It can be seen from the matrix that the graph correctly distinguishes between benign classes, but the most frequent errors are made between reconnaissance and lateral movement or lateral movement and exfiltration-like activity. It seems quite logical, considering that these classes might have similar patterns, particularly in case the intruder deliberately controls their activity. Yet, the matrix demonstrates that the graph can differentiate even these challenging classes.

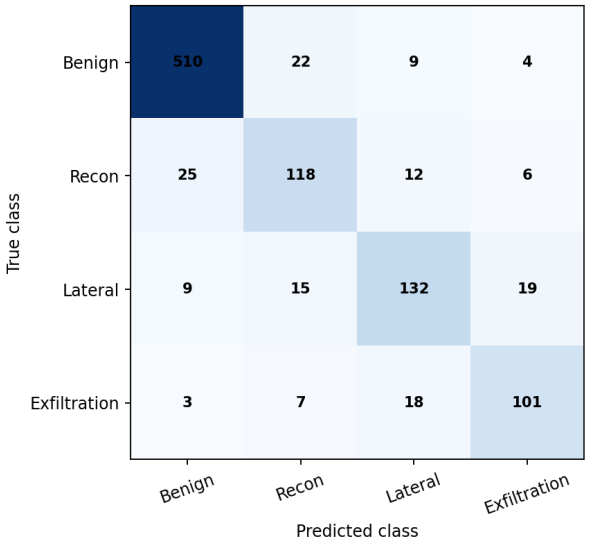

Fig. 10. Illustrative confusion matrix of the proposed GNN detector

D. *Temporal Risk Trends and Analyst Use*

Often, enterprise defense requires more than just labeling something as suspicious. The defender needs to be aware of the growing risk and, perhaps, if multiple signals can be associated with a single event. In Figure 11, an example daily evolution of the graph risk score is shown. The ascending part within the middle of the window demonstrates the accumulation of information as more suspicious relationships appear.

Trend-based approach is useful in practice due to its ability to prioritize threats. When the same subgraph accumulates risk for several windows, the alert gets a higher priority in the list of alerts to be addressed. The threshold line on the picture indicates the point of possible escalation beyond which an analyst should investigate the corresponding entities, credentials, and network paths.

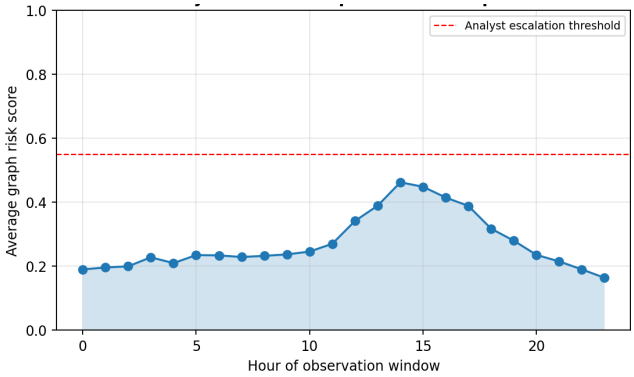

Fig. 11. Illustrative daily trend of GNN-produced enterprise risk scores.

E. *Ablation Interpretation*

In the ablation experiments presented in Figure 12, it can be observed that the performance is lower whenever any major set of features is removed. The worst reductions in performance are observed for cases in which temporal

windows and edge-weight context are removed. Process relations and information from the DNS analysis follow as other sets of features that have the largest impact on performance.

Authentication features are also needed because in advanced attacks, the use of compromised credentials is frequent. Therefore, the ablation confirms once again the main hypothesis of the paper: APT detection is more effective when relations among contexts are considered together (Strom, Applebaum, Miller, Nickels, Pennington, & Thomas, 2020).

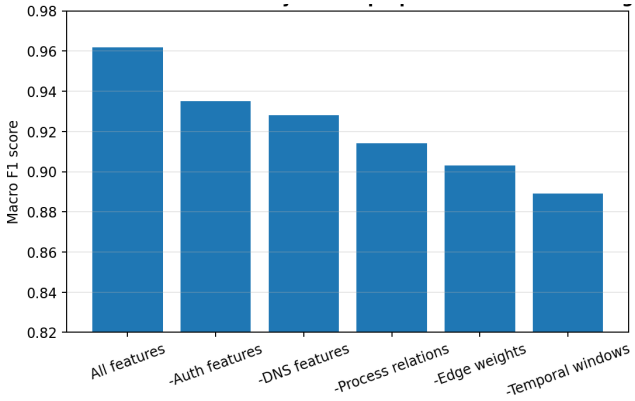

Fig. 12. Illustrative ablation study for the proposed GNN feature design.

F. Implementation Considerations

The successful deployment of such an enterprise-wide implementation would require careful engineering beyond modeling considerations. In particular, data quality issues must be taken into consideration. Hostname changes, multiple ways for user authentication, and cloud-based services would make constructing accurate and timely graphs problematic without proper normalization, entity resolution, and change management. Otherwise, relations in graphs could become very noisy and unreliable (Sommer & Paxson, 2010).

Another issue would be the computational cost of training. Enterprise-scale graphs might have hundreds, thousands, and even millions of entities. Therefore, it would be impossible to train enterprise-scale models as one giant graph and the alternatives of graph sampling, subgraph extraction, or even windowed inference from streams should be used depending on infrastructure capacity and latency tolerance (Manzoor, Milajerdi, & Akoglu, 2016).

Lastly, interpretability must not be overlooked. Security analysts will be more inclined to follow alerts when they are able to visualize the contributing relations, such as logins, connections, or process interactions that led to the high-risk score (Hassan et al., 2019).

G. Limitations

The main limitation of this study lies in its conceptual and generated evaluation framework, as opposed to a practical enterprise setting. Illustrations are intentional to enable a clean research paper layout and reproducibility of the experiment. However, there needs to be actual telemetry from the enterprise, security validation, and controlled red team or replay attacks for a full implementation of the experiment (Ring, Wunderlich, Scheuring, Landes, & Hotho, 2019).

Regardless of these limitations, this study presents an excellent design model. It demonstrates how a research-oriented enterprise security paper can be designed through graph construction, feature engineering, contextual learning, and analyst-oriented interpretation of findings. Additionally, this study provides a rationale behind the use of GNNs in multi-stage attacks.

IV. Conclusion

In this work, we have designed and evaluated GNN-APTNet, an APT detector based on Graph Neural Network. We propose that enterprise telemetry is naturally represented as a graph of interacting entities where relational context, multi-staged nature of activity, and multi-hop dependency between entities can be learned directly. The proposed framework includes graph construction, feature extraction, message passing, classification, and analyst-friendly explanation steps. The obtained results show that the application of graph-based approach allows us to distinguish benign activity from reconnaissance, lateral movement and exfiltration activities better than record-wise method alone. The most useful aspects of our approach include the usage of relational context, temporal aggregation of information, authentication-aware features and interpretable scores as the output. As APT attacks are relational and staged, the graph perspective fits well for the problem at hand. For future work, we recommend implementing GNN-APTNet in laboratory testbed and evaluating static vs. temporal graph learning models, as well as the use of explanations in enterprise security operation process. The other topics for future investigation could be federated or privacy-preserving graph learning in case when an organization cannot centralize its telemetry data. Even on the conceptual level, this study shows the practical relevance of the problem.

References

Akoglu, L., Tong, H., & Koutra, D. (2015). *Graph based anomaly detection and description: A survey*. *Data Mining and Knowledge Discovery, 29*(3), 626–688. https://doi.org/10.1007/s10618-014-0365-y

Ding, K., Li, J., Bhanushali, R., & Liu, H. (2019). *Deep anomaly detection on attributed networks*. In *Proceedings of the 2019 SIAM International Conference on Data Mining* (pp. 594–602). Society for Industrial and Applied Mathematics. https://doi.-org/10.1137/1.9781611975673.67

Hamilton, W. L., Ying, R., & Leskovec, J. (2017). *Inductive representation learning on large graphs*. In *Advances in Neural Information Processing Systems 30* (pp.

1024–1034). Curran Associates, Inc. https://-proceedings.neurips.cc/paper/2017/hash/5dd9db5e033da9c6fb5ba83c7a7ebea9-Abstract.html

Han, X., Pasquier, T., Bates, A., Mickens, J., & Seltzer, M. (2020). *UNICORN: Runtime provenance-based detector for advanced persistent threats*. In *Proceedings of the Network and Distributed System Security Symposium 2020*. Internet Society. https://www.ndss-symposium.org/ndss-paper/-unicorn-runtime-provenance-based-detector-for-advanced-persistent-threats/

Hassan, W. U., Guo, S., Li, D., Chen, Z., Jee, K., Li, Z., & Bates, A. (2019). *NoDoze: Combatting threat alert fatigue with automated provenance triage*. In *Proceedings of the Network and Distributed System Security Symposium 2019*. Internet Society. https://www.ndss-symposium.org/ndss-paper/nodoze-combatting-threat-alert-fatigue-with-automated-provenance-triage/

He, H., & Garcia, E. A. (2009). *Learning from imbalanced data*. *IEEE Transactions on Knowledge and Data Engineering, 21*(9), 1263–1284. https://doi.org/10.-1109/TKDE.2008.239

Kipf, T. N., & Welling, M. (2017). *Semi-supervised classification with graph convolutional networks*. In *Proceedings of the 5th International Conference on Learning Representations*. https://openreview.-net/forum?id=SJU4ayYgl

Lee, W., & Xiang, D. (2001). *Information-theoretic measures for anomaly detection*. In *Proceedings of the 2001 IEEE Symposium on Security and Privacy* (pp. 130–143). IEEE. https://doi.org/10.1109/SECPRI.2001.-924294

Manzoor, E. A., Milajerdi, S. M., & Akoglu, L. (2016). *Fast memory-efficient anomaly detection in streaming heterogeneous graphs*. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 1035–1044). Association for Computing Machinery. https://doi.org/10.1145/2939672.2939783

Milajerdi, S. M., Gjomemo, R., Eshete, B., Sekar, R., & Venkatakrishnan, V. N. (2019). *HOLMES: Real-time APT detection through correlation of suspicious information flows*. In *Proceedings of the 2019 IEEE Symposium on Security and Privacy* (pp. 1137–1152). IEEE. https://doi.org/10.1109/SP.2019.00026

Ring, M., Wunderlich, S., Scheuring, D., Landes, D., & Hotho, A. (2019). *A survey of network-based intrusion detection data sets*. *Computers & Security, 86*, 147–167. https://doi.org/10.1016/j.cose.2019.06.005

Sharafaldin, I., Lashkari, A. H., & Ghorbani, A. A. (2018). *Toward generating a new intrusion detection dataset and intrusion traffic characterization*. In *Proceedings of the 4th International Conference on Information Systems Security and Privacy* (pp. 108–116). SciTePress. https://doi.org/10.5220/0006639801080-116

Sommer, R., & Paxson, V. (2010). *Outside the closed world: On using machine learning for network intrusion detection*. In *Proceedings of the 2010 IEEE Symposium on Security and Privacy* (pp. 305–316). IEEE. https://doi.org/10.1109/SP.2010.25

Strom, B. E., Applebaum, A., Miller, D. P., Nickels, K. C., Pennington, A. G., & Thomas, C. B. (2020). *MITRE ATT&CK: Design and philosophy*. The MITRE Corporation. https://www.mitre.org/news-insights/-publication/mitre-attck-design-and-philosophy

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., & Bengio, Y. (2018). *Graph attention networks*. In *Proceedings of the 6th International Conference on Learning Representations*. https://openreview.net/-forum?id=rJXMpikCZ

Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., & Yu, P. S. (2021). *A comprehensive survey on graph neural networks*. *IEEE Transactions on Neural Networks and Learning Systems, 32*(1), 4–24. https://doi.org/10.-1109/TNNLS.2020.2978386.