Entity Interactivity-Aware Graph Encoder-Decoder Networks for Explainable Relation Prediction against Advanced Persistent Threats

Xiao Yang*[§], Mianxiong Dong[†], Kaoru Ota[†], Gaolei Li*[§], and Chaofeng Zhang[‡]
*School of Electronics, Information and Electrical Engineering, Shanghai Jiao Tong University, Shanghai 200240, China
[†]Department of Information and Electronic Engineering, Muroran Institute of Technology, Muroran, Hokkaido 050-0071, Japan.
[‡] School of Information and Electronic Engineering, Advanced Institute of Industrial Technology, Tokyo 140-0011, Japan.
Email: youngshall@sjtu.edu.cn, mx.dong@csse.muroran-it.ac.jp, ota@csse.muroran-it.ac.jp, zhang-chaofeng@aiit.ac.jp

***Abstract*—Advanced persistent threats (APTs) are premeditatively implemented by malicious organizations via long-term monitoring, hidden vulnerabilities and social engineering attacks. To prevent homogeneous adversaries, APT relation prediction by scrutinizing the similarity between possible vulnerabilities of target systems and real behavior characteristics of threats, has gained increasing attention in recent years. Current studies solely focus on applying feature association to predict, disregarding the impact of feature propagation during entity interaction, which results in inadequate representation and difficulties in expressing unobserved entities. To address this limitation, in this paper, we propose a novel APT relation prediction method: GAP, which comprises three primary parts: 1) graph representation, standardizing the expression of threat information into a uniform graph structure via raw features and named entity recognition 2) encoder, iterative updating feature expression of graph entity through neighboring aggregation mechanism; and 3) decoder, predicting prospective relations for targeted entities in graph by the two-step classifier. Extensive experimental results, derived from real-world datasets, demonstrate the efficacy of the proposed GAP approach, with high prediction accuracy and stability across data of different sizes and relation types.**

***Index Terms*—advanced persistent threat, relation prediction, graph neural network, named entity recognition**

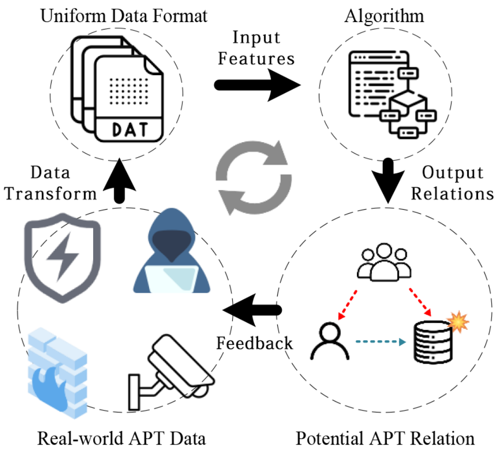


Fig. 1. The general process of APT relation prediction. Real-world APT data are transformed into uniform representation, which is subsequently utilized as input to an algorithmic model for predicting unknown relationships. Finally, the inferred relationships are fed back to the grassroots-level equipments for improving APT monitoring.

I. Introduction

Advanced persistent threat (APT) is a multi-step, long-term cyber attack carried out by highly coordinated adversaries, targeting organizations with high-value information or assets [1], [2]. To enable early detection of APTs and take proactive defense measures, researchers have recently emphasized the utilization of APT relation prediction as a promising approach to tackle this issue, which primarily leverages data characteristics and temporal features of APT-related entities to predict possible attack relations (i.e., forecast potential victims, attacks and adversaries) [3]–[5]. The general process of APT relation prediction is shown in Fig. 1.

Prediction methods based on mathematical probabilistic models or deep learning generally employ the extracted features to compute entity similarity or estimate the likelihood of associated entities or events. However, considering the implicit transmission of information during the interactions between entities (e.g., info. transfer, remote control and data sharing), the impact of the APT attack could extend to proximal entities associated with the targeted one. Besides, it is reasonable to expect that if one APT has been successfully carried out on a particular type of entity, another related APT may also direct its attention toward entities of a similar type. Hence, it is imperative to contemplate the interactions between entities during APT attack for more accurate relation prediction. Moreover, recent research studies solely presented prediction methods that apply identifiable features, and their ability to predict APT entity relation with unknown features is unsatisfactory.

To overcome the aforementioned limitations and inspired by graph learning, graph neural networks (GNN) will be introduced. GNNs update the representation of node by leveraging an aggregation process that utilizes the information

[§]These authors contributed to the work equllly and should be regarded as co-first authors.

of the vicinity to augment its current state. Thereafter, the updated node data could be employed to make predictions on unobserved connections [6]. By representing APT attack information as graph structure, where nodes correspond to APT entities and edges represent the relationships between them, predicting APT entity relationships can be framed as an edge-level task (edge prediction) in GNNs, which entails leveraging the aggregated APT node vector to predict previously unobserved connections.

Following the analysis above, we present a novel **G**raph Neural Network-based **A**PT Relation **P**rediction method, GAP, built upon link prediction-based GNN and named entity recognition (NER). The proposed method is composed of three core steps, 1) graph representation, 2) encoding and 3) decoding. In particular, graph representation extracts entity features and topologies from APT information, and employs NER to adaptively represent all entities as graph. Subsequently, the encoding component exploits the graph vector to update the node embedding vector of APT entities, while the decoder component utilizes two classifiers to accurately predict the relationship between entity nodes.

The proposed GAP leverages the information from nearby nodes, thereby accounting for the interactions between entities. Additionally, feature expressions from aggregation enable a more comprehensive representation of emerging entities, enhancing predicting accuracy for correlations of unobserved entities.

The contributions of this paper are summarized as follows:

- We propose a novel approach for predicting APT relatioships, which represents APT data as graph via NER and identifies potential relations through APT node feature aggregation and two-step prediction.
- We present an entity feature representation method based on NER, which enables a more comprehensive depiction of feature information.
- Through experiments on real-world datasets and under various experimental settings, the proposed method demonstrates notable accuracy and stability across varying dataset sizes.

II. Related Work

A. Attack Relation Prediction

[7] introduced an attack graph-based prediction model, which represents network states as nodes and changes in the attacker's actions as edges. The model aims to verify whether a given attack satisfies specific features, and this work is one of the first to use graph knowledge for attack relationship prediction. [8] leverages natural language processing to extract relevant information from unstructured APT reports and provide corresponding relations. [9] highlighted the challenges of traditional vulnerability analysis techniques, which can be time-consuming and error-prone due to manual inspection. They presented a graph-based approach to represent the network as a directed graph and identify potential attack paths using various algorithms, prioritizing remediation efforts. To address the limitations of manual analysis in APT prediction, [10] proposed a scalable approach that models entities as graphs and identifies potential attack paths. [11] analyzed the feature relations between entities and utilized GRU to infer the corresponding relations. [12] presented an APT analysis method specifically for the industrial internet of things, while [13] proposed a framework that extends traditional attack prediction models to account for modern network structures and protocols. [14] stated the availability of statistical model and used Bayesian-based LSTM to predicate entity relations. [15] presented an automatic attack graph generation framework that combines system topology and bug database to create an accurate representation of an attacker's potential paths through a system. A comprehensive survey of attack relation prediction was provided in [3].

B. Graph Neural Network

GNNs have gained considerable attention for their ability to represent various types of graph data, including social media, networks, and knowledge graphs. The concept of GNNs was first proposed in [16] to handle node-linked data. In order to make GNNs more suitable for semi-supervised learning, [17] designed the graph convolution network for data prediction through node aggregation and data readout. The propagation of node data is usually influenced by its nearby neighbors, leading to high feature similarity. Therefore, an attention mechanism was investigated to properly aggregate the surrounding influences [18]. Practical graph tasks often require the ability to handle large-scale graphs, and to implement GNNs in big graphs, [19] considered using node sampling for aggregation to reduce training complexity and improve model generalization. [20] discussed the possibility of solving the problems of node representation in heterogeneous networks and proposed a heterogeneous attention network framework to capture the semantic aggregation of nodes. Graph isomorphism is widely present in graphs, and [21] proposed a graph isomorphism network method to handle this effectively. [22] gives a comprehensive survey about GNN.

III. Methodology

In this section, our proposed APT relationship prediction method, GAP, will be elaborated. GAP comprises three functional parts: graph representation, encoder, and decoder. The initial part, graph representation, is to extract graph of entity features and topology from collected APT information via NER. Next, the encoder is responsible for aggregating the ultimate feature representation from graph. And finally, the decoder is then tasked with employing the aggregated results to make predictions regarding the relevant unmarked relations by a two-step classification mechanism. The general framework of GAP is shown in Fig. 2.

In the rest part, graph representation will first be introduced, followed by a detailed description of the encoder and the decoder.

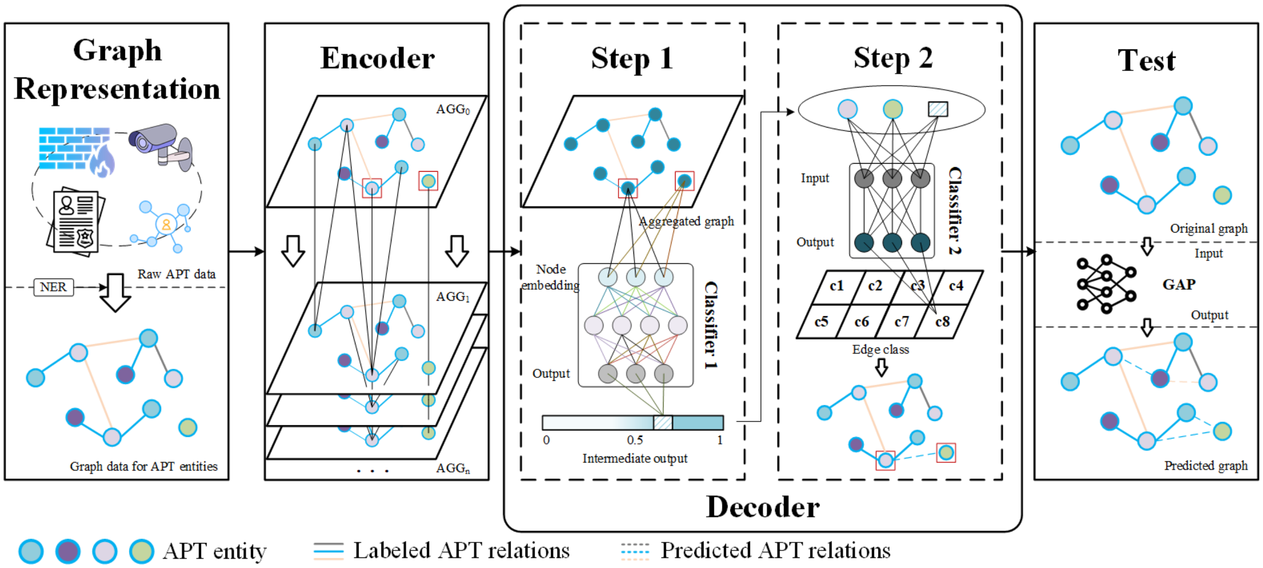


Fig. 2. Illustration for the proposed APT relation prediction scheme: GAP. Initially, the real-world APT data is transformed into a graph-structured dataset. Subsequently, the graph data is used as input to the GAP model, which leverages the encoder and decoder modules to infer the missing APT relationships. Finally, the obtained output results are utilized to construct the comprehensive APT relationship graph for the entire dataset.

A. *Graph Representation*

The raw APT information will undergo a conversion process to generate graph structure $\boldsymbol{G} = (\boldsymbol{A}, \boldsymbol{X})$ comprising of two main parts: feature representation $\boldsymbol{X}$ and adjacency matrix $\boldsymbol{A}$.

1) Feature Representation: To achieve improved depiction of APT entity, we employ NER tool to annotate entity types in APT informations. NER labels entity in accordance with established textual content, which better specifies the relevant classes to which the APT entity belongs [23]. By designating certain classes frequently employed in APT reports, NER algorithm could be applied to detect and classify entities within APT related logs. While entity types may vary across different datasets, most of them should encompass essential categories, e.g., APT, organization, company, person, time, and location.

Specifically, for one APT entity, a universal feature representation approach was employed while refraining from constraining the feature dimensionality, and it is shown as follows:

$$\begin{aligned} feat &= (attr, N_{type}) \\ s.t. \quad attr &= (\rho_1,\ \rho_2,\ ...,\ \rho_k), \end{aligned} \tag{1}$$

where $attr$ is the one hot expression of the APT-related features, ρ_i is one specific feature (e.g., system version, port num. and certain software patch) and $N_{type} \in \{n_1, n_2, ..., n_K\}$ is the NER classification of the entity. After obtaining the distinct feature vector expressions for each entity, we concatenate them to form the feature matrix $\boldsymbol{X}$, which is shown in Eq. 2.

$$\boldsymbol{X} = \begin{pmatrix} feat_1 \\ feat_2 \\ ... \\ feat_n \end{pmatrix} \tag{2}$$

This feature representation enables the system to independently determine the extent of the feature expression in accordance with the present entities, and specifies APT NER type for more precise relation classification.

2) Adjacent Matrix: We further construct the adjacency (topology) matrix $\boldsymbol{A}$ by leveraging the entities implicated in the APT information (i.e., observed APT relation info.):

$$\boldsymbol{A} = \begin{pmatrix} adj_{11}, & adj_{12}, & ..., & adj_{1n} \\ adj_{21}, & adj_{22}, & ..., & adj_{2n} \\ & ... & & \\ adj_{n1}, & adj_{n2}, & ..., & adj_{nn} \end{pmatrix}, \tag{3}$$

where $adj_{ij} = 1$ indicating the observed presence of a relationship between entity node i and j (e.g., attack relation, subordination and partnership) and vice versa representing unrecognized relationship.

B. *Encoder*

Based on the graph representations derived from the preceding section, we employed GNN to learn the relationships between APT entities in acquired data, with the aim of using observed knowledge to infer potential APT relationships in unknown information.

The GNN designed in our study comprises two fundamental modules, namely encoder and decoder. We will illustrate the encoder module first.

The main function of the encoder in GNNs is to perform feature aggregation (or information aggregation), which is achieved through GNN's information propagation (message passing) mechanism. On the basis of the propagation, each node updates its state according to its neighbors' information, thereby gradually disseminating and aggregating information (node features) across the entire graph. The aggregation process computes the updated expression of the node by iterations, and the equation is shown as follow:

$$\boldsymbol{H}^{(s)} = Activation(\tilde{\boldsymbol{D}}^{-\frac{1}{2}}\tilde{\boldsymbol{A}}\tilde{\boldsymbol{D}}^{-\frac{1}{2}}\boldsymbol{H}^{(s-1)}\boldsymbol{W}^{(s-1)}), \tag{4}$$

where $\boldsymbol{H^{(s)}}$ is the updated expression of the feature matrix of all APT entities in s-th iteration, whose initial expression is $\boldsymbol{X}$ in Eq. 2, and $\boldsymbol{W}$ is weight parameter matrix. $\tilde{\boldsymbol{A}}$ is the updated adjacent matrix for APT entities, which is shown as follow:

$$\tilde{\boldsymbol{A}} = \boldsymbol{A} + \boldsymbol{I}, \tag{5}$$

where $\boldsymbol{I}$ is identity matrix. $\tilde{\boldsymbol{D}}$ is the improved degree matrix, which is calculated by

$$\tilde{\boldsymbol{D}} = \boldsymbol{D} + \boldsymbol{I}, \tag{6}$$

where $\boldsymbol{D}$ is the degree matrix of $\boldsymbol{A}$.

During each iteration, the input feature for each node is subjected to a transformation process to generate a low-dimensional representation. It is then passed through a fully connected layer that incorporates an activation function. Afterwards, contextual representation information of each node is obtained by aggregating the representation information of its neighboring nodes. By employing iterative aggregation as outlined above, we are able to get more comprehensive and holistic representation of node updates.

C. Decoder

By employing the aggregated (encoded) information acquired in the preceding step, the decoder is responsible for predicting the APT relationship.

Since APT messages often contain a considerable volume of normal information, the majority of relationships between APT entities tend to be normal relationships. To minimize redundancy, we have devised a two-step decoder. Firstly, the model determines the existence of a relationship between two APT entities. Then, in the second step, the class of entity relationship is identified. If a singular classifier is employed to facilitate the two-step function, the resultant complexity of the classifier would increase. However, by utilizing one classifier for each single step, it becomes feasible to enhance the classification accuracy with regard to their respective tasks. Hence, two classifiers are combined to implement the aforementioned two-step function.

The result of the first classifier for entity node i and j is shown as follows:

$$\boldsymbol{Z} = Sum(\boldsymbol{H_i} * \boldsymbol{H_j}), \tag{7}$$

where $\boldsymbol{H_i}$ denotes the vector corresponding to entity i in $\boldsymbol{H}$, output of the first step, and $\boldsymbol{H_j}$ is the vector for corresponding entity j. Sum is the summation function for row vectors. The primary function of the first classifier involves utilizing similarity between every unconnected pair of nodes to predict whether or not they should be connected (have relations). This prediction applies to all unconnected pairs in the graph.

In accordance with the first classifier's output results, the subsequent task involves assigning a distinct category to the edges connecting two APT entities. To accomplish this, a deep neural network is employed for multi-classification, with the initial features of both nodes on the edges and the output of the first classifier serving as inputs. The network outputs the precise relationship type between the two entities.

The second classifier is expressed as follow:

$$y = DNN(feat_i, feat_j, Z), \tag{8}$$

where y is the final APT relation prediction result; $feat$ is the processed feature from Eq. 1; while Z represents the output of the first classifier predicated on these two entities.

Building upon the GAP method, the input APT information is transformed into a unified feature representation, which is subsequently processed by the encoder and decoder components. The resulting output captures the edge relationships between nodes, enabling the construction of the APT entity relationship graph, i.e., predicting potential APT relations.

IV. Experiment

In this section, we present a detailed assessment of the efficacy of the proposed method. Our experimental evaluation is divided into two major parts, comparison experiments and ablation experiments. The former entails benchmarking our approach against several APT relational prediction methods, while the latter tests the performance of our algorithm under various model configurations. For comparison experiments, methods in [14] (bayesian LSTM-based relationship prediction model) and [11] (multi-headed attention GRU model) are employed as baselines to evaluate the performances.

A. Dataset Settings

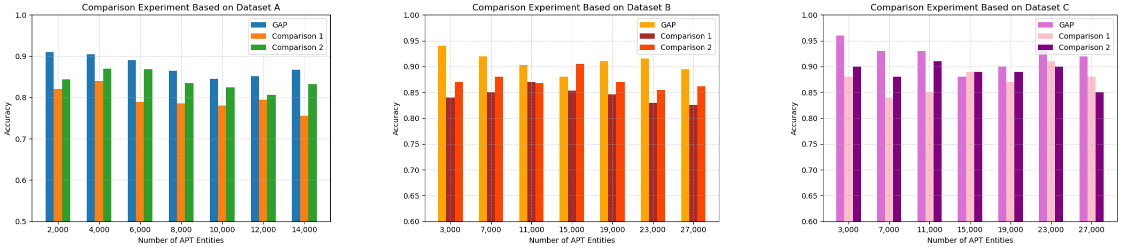
The effectiveness of our proposed algorithm will be evaluated via using three different real-world APT datasets collected from APT detection devices. Detailed information about the datasets are listed in the following Table I.

TABLE I
DATASET INFORMATION

Dataset	APT Info.	Entities	Features	NER Types	Relations
A	8463	15063	4179	7	5
B	5251	28762	3752	13	6
C	6745	27542	5190	21	9

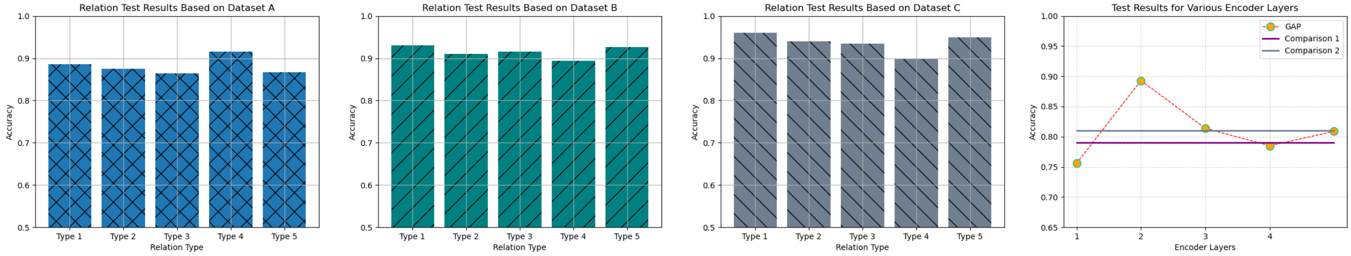
B. Model Parameters

To help better reproduce the GAP method, we provide a comprehensive description of the experimental parameters utilized in our study. The detailed information regarding them is presented in Table II.



(a) Comparison experiment under dataset A (b) Comparison experiment under dataset B (c) Comparison experiment under dataset C

Fig. 3. Illustration of the test results under different entity sizes in dataset A, B and C. Dataset A was partitioned as ranging from 2,000 to 14,000; Dataset B was segmented with a range of 3,000 to 27,000; and Dataset C was split as ranging from 3,000 to 27,000.



(a) Relation test results for dataset A (b) Relation test results for dataset B (c) Relation test results for dataset C (d) Encoder layer test results

Fig. 4. Illustrations of the relation test results based on dataset A, B and C, and model performances for various encoder layers. The proposed GAP method will be employed to categorize five types of edges, namely 1) APT to company, 2) APT to person, 3) APT to location, 4) APT to tool, and 5) other types, to evaluate the performance of GAP.

TABLE II
MODEL PARAMETERS

Parameter	Value
Encoder Layers	2
Encoder Hidden Dimension	128
Encoder Output Dimension	128
Encoder Activation Function	ReLU
Decoder Hidden Dimension	10

C. Comparison Experiment Results

To improve the effectiveness of model training, testing, and evaluation, and to minimize overfitting, we partitioned each dataset into multiple subsets of different sizes, to thoroughly evaluate the predictive performance of our proposed GAP approach. To ensure objectivity in the results, each test was conducted ten times and the average value was calculated.

The experimental results based on different sizes of data can be seen in Fig. 3.

1) Dataset A: As could be caught from Fig. 3(a), the accuracy of GAP fluctuates slightly with the increase in the amount of APT entities, but it generally remains within 84% and 91% with little overall fluctuation (average accuracy 87.6%), which also indicates that the proposed algorithm is effective for datasets of varying sizes. In contrast, the comparison algorithm 1 fluctuates between 75% and 84% (average accuracy 79.5%), and the comparison algorithm 2 fluctuates between 80% and 87% (average accuracy 83.9%).

2) Dataset B: As depicted in Fig. 3(b), GAP got an average accuracy of 90.9% (with a maximum 94% and a minimum 88%) across multiple sizes of data. Meanwhile, the average accuracy of comparison algorithms 1 and 2 is around 84.5% and 88.7%. The observed fluctuations in all three methods did not reach statistical significance.

3) Dataset C: As illustrated in Fig. 3(c), the accuracy attains a peak of 96%, while the mean remains approximately 92.2%. Comparison algorithm 1 demonstrates a maximum value of 91% and an average of roughly 87.4%, while comparison algorithm 2 displays a maximum value of 91% and an average of 88.8%.

D. Ablation Experiment Results

To further explore the performance of GAP, ablation experiments were implemented on GAP. we first observe the prediction accuracy for different APT relationship categories in three datasets to determine whether the relation categories hold an impact on the model performances. Five types of APT relationships, 1) APT to company, 2) APT to person, 3) APT to location, 4) APT to tool, and 5) other types, are selected to analyze the accuracy of the GAP. All the three datasets are exploited to conduct the experiments, and each test is repeated ten times to calculate the average value so as to obtain objective results. The test results are shown in Fig. 4

1) Dataset A: As illustrated in Fig. 4(a), GAP achieves an average result of about 88.1% (with a maximum of 91% and a minimum of 86%) on all kinds of edge predictions. Overall, GAP achieves type-agnostic results for APT relationship prediction, and has preferable prediction performances on all five types of relations.

2) Dataset B: Fig. 4(b) displays the results of relation test results of dataset B, which exhibits that GAP performs relatively consistent performance (maximum 93%, minimum 89%, average 91.5%) and did only minor accuracy fluctuations.

3) Dataset C: The result of relation test for dataset C is shown in Fig. 4(c) and it indicates the highest accuracy of 96% and an average accuracy of 93.6%.

Besides relation types, to examine the potential impact of the number of encoder layers on the model's performance, we varied the number of encoder layers and evaluated the resulting performance of GAP. Dataset A was selected as the experimental subject and conducted ten repetitions of each test ten times to compute the average outcome. The result is shown in Fig. 4(d).

As can be seen from the figure, the optimal performance of the model is achieved when the encoder architecture comprises two layers, leading other results by about 8% to 14% in accuracy performance. The accuracy differences could be attributed to the inadequate interaction (i.e., resulting in insufficient features for model to predict) among the data features when one layer is employed, and the redundant data interactions (i.e., leading to redundant feature representation) resulting from the excessive number of layers.

Conclusion

In this study, we address the need for predicting APT relationships and identify the limitations of current research in this area, including 1) the failure to consider entity feature interactions and 2) the challenge of predicting attacks on unknown entities. To overcome these limitations, we propose a GNN-based APT relation prediction method called GAP, which comprises 1) graph representation, 2) encoder and 3) decoder. Representation employs NER to obtain a uniform feature and topology representation of APT entities. Afterwards, the encoder component aggregates features and generates comprehensive graph entity representations, while the decoder component exploits two classifiers to identify relationships. Our experiments on real-world datasets demonstrate that the proposed algorithm achieves preferable predictions on datasets of varying sizes and APT relationship types. Future research will focus on enhancing the algorithm's ability to predict multi-step APT relationships.

Acknowledgement

This research is supported by Topsec Technologies Group, National Nature Science Foundation of China (No. 62202303, 62202302, and U20B2048), Shanghai Sailing Program (No. 21YF1421700), and JSPS KAKENHI (No. JP22K17884).

References

[1] T. Ye, G. Li, I. Ahmad, C. Zhang, X. Lin, and J. Li, "FLAG: few-shot latent dirichlet generative learning for semantic-aware traffic detection," *IEEE Trans. Netw. Serv. Manag.*, vol. 19, no. 1, pp. 73–88, 2022.

[2] G. Li, J. Wu, S. Li, W. Yang, and C. Li, "Multitentacle federated learning over software-defined industrial internet of things against adaptive poisoning attacks," *IEEE Trans. Ind. Informatics*, vol. 19, no. 2, pp. 1260–1269, 2023.

[3] M. Husák, J. Komárková, E. Bou-Harb, and P. Čeleda, "Survey of attack projection, prediction, and forecasting in cyber security," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 1, pp. 640–660, 2018.

[4] G. Li, Y. Zhao, W. Wei, and Y. Liu, "Few-shot multi-domain knowledge rearming for context-aware defence against advanced persistent threats," *CoRR*, vol. abs/2306.07685, 2023.

[5] H. Li, J. Wu, H. Xu, G. Li, and M. Guizani, "Explainable intelligence-driven defense mechanism against advanced persistent threats: A joint edge game and AI approach," *IEEE Trans. Dependable Secur. Comput.*, vol. 19, no. 2, pp. 757–775, 2022.

[6] J. Zhou, G. Cui, S. Hu, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, and M. Sun, "Graph neural networks: A review of methods and applications," *AI open*, vol. 1, pp. 57–81, 2020.

[7] O. Sheyner, J. Haines, S. Jha, R. Lippmann, and J. M. Wing, "Automated generation and analysis of attack graphs," in *Proceedings 2002 IEEE Symposium on Security and Privacy*, pp. 273–284, IEEE, 2002.

[8] Y. Park and T. Lee, "Full-stack information extraction system for cybersecurity intelligence," in *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: EMNLP 2022 - Industry Track, Abu Dhabi, UAE, December 7 - 11, 2022* (Y. Li and A. Lazaridou, eds.), pp. 531–539, Association for Computational Linguistics, 2022.

[9] P. Ammann, D. Wijesekera, and S. Kaushik, "Scalable, graph-based network vulnerability analysis," in *Proceedings of the 9th ACM Conference on Computer and Communications Security*, pp. 217–224, 2002.

[10] X. Ou, W. F. Boyer, and M. A. McQueen, "A scalable approach to attack graph generation," in *Proceedings of the 13th ACM conference on Computer and communications security*, pp. 336–345, 2006.

[11] Y. Cai, Q. Yang, W. Chen, G. Wang, T. Liu, and X. Liu, "A new knowledge inference approach based on multi-headed attention mechanism," in *2022 IEEE 6th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, pp. 825–829, IEEE, 2022.

[12] H. Mouratidis and V. Diamantopoulou, "A security analysis method for industrial internet of things," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 9, pp. 4093–4100, 2018.

[13] O. Stan, R. Bitton, M. Ezrets, M. Dadon, M. Inokuchi, Y. Ohta, T. Yagyu, Y. Elovici, and A. Shabtai, "Extending attack graphs to represent cyber-attacks in communication protocols and modern it networks," *IEEE Transactions on Dependable and Secure Computing*, vol. 19, no. 3, pp. 1936–1954, 2020.

[14] Y. Fang, Y. Zang, and J. Ge, "Research on relation extraction method based on similar relations and bayesian neural network," in *Journal of Physics: Conference Series*, vol. 1792, p. 012011, IOP Publishing, 2021.

[15] C. Hankin, P. Malacaria, *et al.*, "Attack dynamics: An automatic attack graph generation framework based on system topology, capec, cwe, and cve databases," *Computers & Security*, vol. 123, p. 102938, 2022.

[16] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," *IEEE transactions on neural networks*, vol. 20, no. 1, pp. 61–80, 2008.

[17] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, OpenReview.net, 2017.

[18] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," *arXiv preprint arXiv:1710.10903*, 2017.

[19] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," *Advances in neural information processing systems*, vol. 30, 2017.

[20] X. Wang, H. Ji, C. Shi, B. Wang, Y. Ye, P. Cui, and P. S. Yu, "Heterogeneous graph attention network," in *The world wide web conference*, pp. 2022–2032, 2019.

[21] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, "How powerful are graph neural networks?," *arXiv preprint arXiv:1810.00826*, 2018.

[22] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and S. Y. Philip, "A comprehensive survey on graph neural networks," *IEEE transactions on neural networks and learning systems*, vol. 32, no. 1, pp. 4–24, 2020.

[23] J. Li, A. Sun, J. Han, and C. Li, "A survey on deep learning for named entity recognition," *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 1, pp. 50–70, 2022.