From Fine-grained to Refined: APT Malware Knowledge Graph Construction and Attribution Analysis Driven by Multi-stage Graph Computation*

Rongqi Jing[1,2], Zhengwei Jiang[1,2], Qiuyun Wang[1,2](✉), Shuwei Wang[1,2], Hao Li[1,2], and Xiao Chen[1,2]

[1] Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China
{jingrongqi, jiangzhengwei, wangqiuyun, et al.}@iie.ac.cn
[2] School of Cyber Security, University of Chinese Academy of Sciences, Beijing 100049, China

Abstract. In response to the growing threat of Advanced Persistent Threat (APT) in network security, our research introduces an innovative APT malware attribution tool, the APTMalKG knowledge graph. This knowledge graph is constructed from comprehensive APT malware data and refined through a multi-stage graph clustering process. We have incorporated domain-specific meta-paths into the GraphSAGE graph embedding algorithm to enhance its effectiveness. Our approach includes an ontology model capturing complex APT malware characteristics and behaviors, extracted from sandbox analysis reports and expanded intelligence. To manage the graph's granularity and scale, we categorize nodes based on domain knowledge, form a correlation subgraph, and progressively adjust similarity thresholds and edge weights. The refined graph maintains crucial attribution data while reducing complexity. By integrating domain-specific meta-paths into GraphSAGE, we achieve improved APT attribution accuracy with an average accuracy of 91.16%, an F1 score of 89.82%, and an average AUC of 98.99%, enhancing performance significantly. This study benefits network security analysts with an intuitive knowledge graph and explores large-scale graph computing methods for practical scenarios, offering a multi-dimensional perspective on APT malware analysis and attribution research, highlighting the value of knowledge graphs in network security.

Keywords: APT malware · attribution analysis · graph clustering · graph embedding · ensemble machine learning.

1 Introduction

In the realm of cybersecurity, Advanced Persistent Threats (APTs) are a major concern. APTs are intricate and covert threats, designed to infiltrate networks

* Supported by Youth Innovation Promotion Association,CAS (No.2023170)

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63749-0_6

for extended periods with the goal of stealing sensitive information or executing destructive attacks. These threats often employ sophisticated malware to bypass traditional security defenses [33, 10]. Understanding and mitigating APT attacks, especially APT malware, are crucial.

Researchers have employed various techniques like machine learning, behavioral analysis, code signature recognition, and malware sandbox testing to enhance APT malware detection and attribution [15, 13]. However, cybersecurity faces challenges due to the evolving nature of APT attacks and the complexity of large-scale networks and datasets [5].

Knowledge graphs have proven effective in network security analysis. They provide a comprehensive way to understand malware, attacker behavior, and attack chains by integrating data from various sources [11, 17, 26]. However, constructing and analyzing complex knowledge graphs pose challenges, such as the scale of nodes and edges in large networks [20]. Traditional knowledge graphs derived from abstract data sources struggle to support multidimensional attribution analysis.

Therefore, we propose a refined representation of an APT malware knowledge graph: APTMalKG. Based on in-depth analysis of malware, covering static and dynamic features, rule-based TTPs, and communication behavior, among other information. Through the construction of the APTMalKG knowledge graph, we capture multidimensional features and behaviors in a structured manner, providing a more comprehensive view for network security analysts. However, due to the fine-grained nature of knowledge graphs, their vast scale and complexity pose challenges for efficient graph computation. To address this issue, we introduce a multi-stage graph clustering method, which organizes nodes effectively into related heterogeneous subgraphs based on node attributes and structural similarity, thereby achieving efficient node fusion and graph refinement. Additionally, we consider the optimization of graph representation, defining malware aggregation features and abstract features as two dimensions of graph representation, gradually reducing graph complexity while preserving critical attribution information. Finally, we apply domain-defined meta-path-enhanced GraphSAGE graph embedding methods, guided by domain knowledge-defined malware behavior paths, to provide a streamlined and efficient APT attribution classification model. In summary, our research offers several key advantages:

- Firstly, the construction of knowledge graphs enables us to capture various features and behaviors of malware in a structured manner, providing a more comprehensive view.
- Secondly, the multi-stage graph clustering method refines knowledge graphs and optimizes graph representations, reducing graph complexity while retaining important information.
- Lastly, domain-defined meta-path-enhanced GraphSAGE graph embedding enhances attribution classification accuracy and efficiency, enabling us to better understand and respond to APT threats.

Our study aims to delve into how to construct and optimize knowledge graphs from the perspective of APT malware, providing robust support for attribution

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63749-0_6

analysis. Simultaneously, we provide a new multidimensional visualization graph for APT malware analysis and attribution research, aiding security teams in better understanding and responding to APT threats, and showcasing the potential value of knowledge graphs in the field of network security.

2 Related Work

In recent years, APT malware has emerged as a major concern in cybersecurity. The in-depth analysis and accurate attribution of such attacks are crucial. Researchers have been focusing on leveraging machine learning and deep learning to enhance the identification and attribution of APT attacks. Methods like deep learning algorithms and transfer learning have been instrumental in classifying APT malware, particularly in understanding API behaviors [31]. Techniques such as sequence pattern feature mining and combining BERT with LSTM have also been explored for improved detection [4, 2].

APT attribution research confronts challenges like data scarcity and attacker deception. To combat these, various methodologies, including attack pattern analysis and cross-domain data fusion, are being utilized to gain more precise insights [25, 23, 24]. Different feature dimensions like static, dynamic, and network features each provide unique perspectives for malware analysis.

The construction of knowledge graphs, representing malware attributes and relationships, has become a vital tool in APT malware analysis [8, 17, 22]. These graphs use standardized malware descriptions covering functionality, behavior, and targets to support analysis and detection tasks [1].

Graph representation learning, converting data into vector representations, has advanced in malicious software classification and attribution. Techniques like graph clustering, graph embedding, and graph convolution are being applied to improve the precision of malware classification and attribution [30, 32, 14, 3, 19].

In summary, current research highlights the importance of a multidimensional approach, knowledge graph construction, and attribution research in addressing APT threats. However, challenges in constructing high-quality APT malware attribution graph data and selecting appropriate models persist. Addressing these issues is crucial for the development of graph representation learning in malware classification and attribution. This study aims to construct a refined APT malware knowledge graph through multi-stage graph clustering, enhancing the effectiveness of attribution research.

3 Proposed Method

Fig. 1 illustrates the construction of APTMalKG, which involves the analysis of malware in both static and dynamic dimensions, incorporating Tactics, Techniques, and Procedures (TTPs) as well as location data. The process employs a multi-stage graph clustering approach to refine the graph, reduce complexity, and enhance the GraphSAGE algorithm with domain-specific metapaths to enable efficient APT attribution.

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63749-0_6

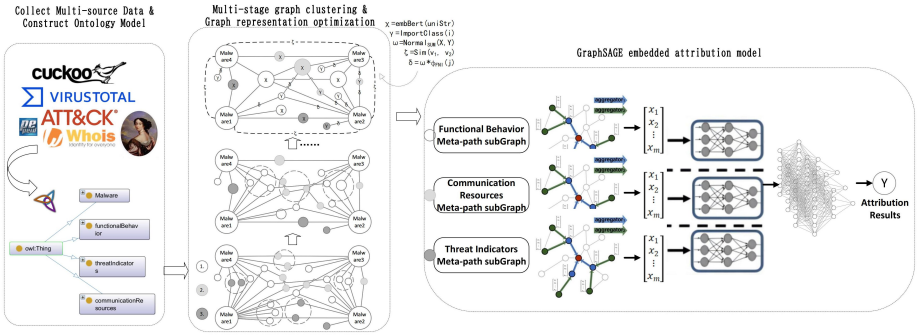


Fig. 1. System architecture of APTMalKG attribution model.

Initially, the research extracts multidimensional malware data and constructs a comprehensive knowledge graph. Subsequently, the multi-stage graph clustering method creates node-associated subgraphs based on attribute and structural similarity, optimizing the graph's representation while simplifying its complexity. The article distinguishes between two graph dimensions: aggregate and abstract malware features, further enhancing information processing efficiency. Lastly, the incorporation of malware metapaths enhances GraphSAGE, resulting in a more effective APT attribution model tailored for network security analysis.

3.1 APT Malware Ontology Mode

In the evolving field of network security, constructing and applying APT knowledge graphs has become vital. This approach organizes complex information about APT malware and its activities to enhance network security analysis. We introduce the APT malware ontology model, refining existing APT knowledge graphs. Our new framework categorizes entities into static properties, functional behaviors, communication resources, and threat indicators. We select 20 feature dimensions from malware analysis, including static, dynamic, and intelligence aspects. These dimensions help form a comprehensive ontology model, as depicted in Fig. 2, providing insights into APT attacks and improving defense strategies.

In this model, we define four core classes: APT Malware, Functional Behaviors, Communication Resources, and Threat Indicators. APT Malware captures static analysis information, Functional Behaviors describe system-level behaviors, Communication Resources cover network-related information, and Threat Indicators highlight signatures and classifications. Associations between entities are categorized into interactions with causality and process-centered calls, outputs, and mapping edges. This ontology model enhances our understanding of APT attacks and aids network security analysts.

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63749-0_6

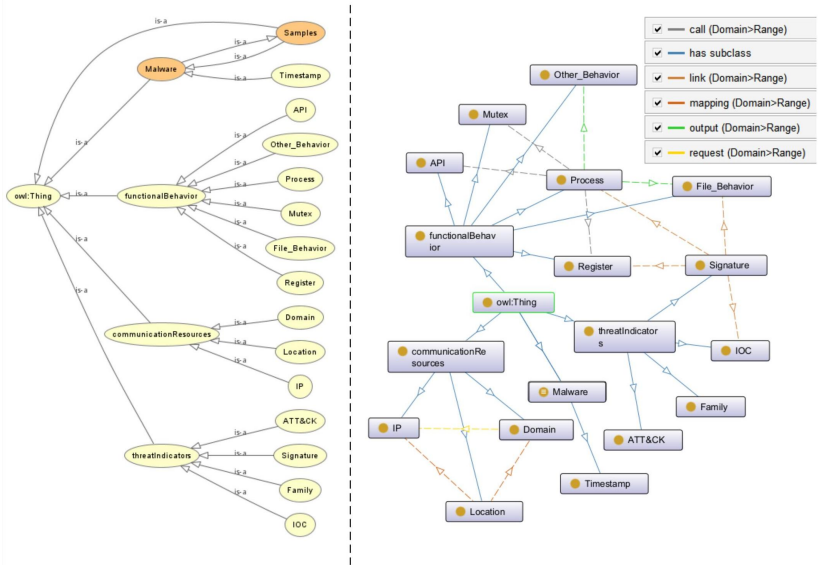


Fig. 2. APT Malware Ontology Model.

3.2 Multistage Graph Clustering & Graph Representation Optimization

In APT malware attribution analysis, the typical method involves examining shared code functionality, attack resources, or TTPs families to uncover the groups and intentions behind APT attacks. Our study constructs a knowledge graph using various behaviors and resource associations from malware analysis, aiming to identify key association dimensions to pinpoint the APT malware group. To handle the complexity of APT malware hiding among other behaviors, we use a multi-stage graph clustering method to reduce graph volume and complexity, improving computational efficiency while maintaining analysis quality. We streamline the graph volume and optimize representation post-refinement to enhance the accuracy and efficiency of APT malware attribution analysis.

Since there is no dynamic interaction between static features, this paper considers static properties as APT Malware node attributes in the graph calculation. In addition to the Malware node, we subdivided sample information into three categories: functional behaviors, communication resources, and threat indicators, assigning a unified unique identifier (uniStr) to each node type for efficient data preprocessing. Then, we introduced a multi-stage graph clustering method and defined two graph representation dimensions: aggregation features and abstract features. The aggregation dimension simplifies the graph by merging similar nodes, revealing malware macrostructure and behavioral patterns. The abstract dimension focuses on extracting abstract characteristics, improving graph representation efficiency. By dividing malware feature dimensions into

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63749-0_6

Algorithm 1 Multi-Stage Graph Clustering for Aggregation Dimension

Require: Original fine-grained graph $G_{\text{fine-grained}}$, number of stages S, similarity thresholds $\{\theta_s\}_{s=1}^{S}$

Ensure: Refined graph G_{new}

Preprocessing: Initialize $G_{\text{clustering\_dimension}}$ with node embeddings from $G_{\text{fine-grained}}$: $\chi \leftarrow \text{embBert}(\text{uniStr})$

Stage 1 - Pairwise Similarity Calculation:

Get pair (v_i, v_j) based on shared neighbors.

for each node pair (v_i, v_j) in $G_{\text{fine-grained}}$ **do**

  Calculate $Sim(v_i, v_j) = \mu_1 \text{Pro}_{\text{sim}}(v_i, v_j) + \mu_2 \text{Stru}_{\text{sim}}(v_i, v_j), \mu_1 + \mu_2 = 1$

  Form edges in $G_{\text{clustering\_dimension}}$ using these similarities: $\zeta \leftarrow Sim(v_i, v_j)$

end for

Stage 2 - Multi-Stage Community Clustering:

for stage $s = 1$ to S **do**

  Use θ_s to filter edges ζ in $G_{\text{clustering\_dimension}}$.

  Apply graph community detection algorithms on $G_{\text{clustering\_dimension}}$.

  Merge nodes and edges based on communities to update $G_{\text{clustering\_dimension}}$.

end for

Stage 3 - Final Normalization:

Normalize weights from nodes X to Y based on the statistical frequency of records pointing to the aggregated node using the formula: $Normal_{sum}(X, Y) = \frac{\sum w_{X \to Y}}{\sum w_{X*}}$

Update the edge weight $\omega \leftarrow Normal_{sum}(X, Y)$ on $G_{\text{clustering\_dimension}}$

return $G_{\text{new}} \leftarrow \text{RefineGraph}(G_{\text{clustering\_dimension}})$

aggregate and abstract dimensions, we reduce computational resource requirements and enhance information processing efficiency.

Aggregate Dimension Represention Optimization. In the aggregation dimension, our focus is on handling nodes characterized by random strings and large quantities, which are often challenging to obtain through simple traversal. To reduce the graph's complexity, we cluster and merge nodes that are similar or strongly related, thus preserving essential structural and informational features. We calculate the similarity between node pairs, considering both structure and attributes, and introduce new edges and weights to construct a refined heterogeneous graph. We perform multiple stages of graph clustering calculations to enhance the graph's structure. To ensure the unsupervised graph clustering's credibility, we employ various graph community detection algorithms. During these stages, we carry out node fusion and replacement within the same community. We also define the attributes and edge weights for the merged nodes, creating a detailed and enriched APT malware knowledge graph, as illustrated in Algorithm 1. The optimized graph representations are stored in the graph structure, with χ and γ denoting node feature representations in different aggregation and abstract dimensions. The weight representations ω and δ are used to quantify the strength and relevance of connections between nodes.

Where, embBert(uniStr) represents the embedding vector of a node obtained by processing uniStr using the BERT model. Prosim(v_i, v_j) indicates the attribute similarity determined by the node's embedding vector, while Strusim(v_i, v_j)

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63749-0_6

signifies the structural similarity influenced by the graph's topology. The term $\sum w_{X \to Y}$ represents the cumulative original weights of edges originating from node X and pointing towards the merged node Y, whereas $\sum w_{X*}$ denotes the total weight of all edges originating from node X and directed towards any other node. In this approach, the community clustering algorithm integrates various clustering techniques [18], which are instrumental in revealing the community structure and behavioral patterns of malware in APT malware analysis.

Abstract Dimension Representation Optimization. Unlike the aggregation dimension, the abstract dimension deals with types that have clear naming conventions and a limited number of nodes. These nodes can be traversed within a certain time frame, allowing for deeper and more specific analysis. In the abstract dimension, we enhance the quality and usability of the knowledge graph by precisely capturing and expressing the key features of these limited nodes.

At the same time, for measuring the importance of terms this paper introduces the Feature Node Importance (FNI) metric. FNI is used to measure the uniqueness of node features for certain categories, reducing the influence of features that appear in almost all categories in subsequent community partitioning. The specific formula for calculating FNI is as follows:

$$\varphi_{FNI}(j) = 1 - \left(\frac{\sum (g_j) - 1}{\sum (G)} \right)^2, \quad g_j = \text{Uniq}\,(R_{ij}) \tag{1}$$

$\sum(G)$ refers to the total count of all cluster categories across the entire graph, and $\sum(g_j)$ refers to the count of distinct cluster categories associated with a particular feature node after deduplication.

In equation (1), $\sum(G)$ represents the total number of cluster categories included, and R_{ij} represents the set of relevant edges connecting samples in cluster i to feature node j. g_j represents the set of cluster associations contained in feature node j after deduplication based on feature node j. $\sum(g_j)$ represents the number of cluster categories associated in feature node j. To capture the trend of decreasing influence as features become more general, a non-linear decreasing function $\varphi_{FNI}(j)$ is fitted. Based on the importance indicator, we define edge weights in the abstract dimension as follows:

$$\delta : \omega \cdot \varphi_{FNI}(j) \tag{2}$$

Furthermore, nodes evaluate their impact on classification by computing the attribute γ for node i, as expressed by the equation (3). Where $F(i)$ represents the total frequency of feature node i in the sample set. $\sum_{C \in \text{class All}}$ represents the summation over all categories *class* C. $f(i, C)$ represents the frequency of feature node i included in category C within the sample set.

$$\gamma : ImportClass(i) = \sum_{C \in \text{class All}} \frac{f(i, C)}{F(i)} \tag{3}$$

This formula accounts for node i's influence across different classifications, considering both its frequency within the sample set and its distribution among

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63749-0_6

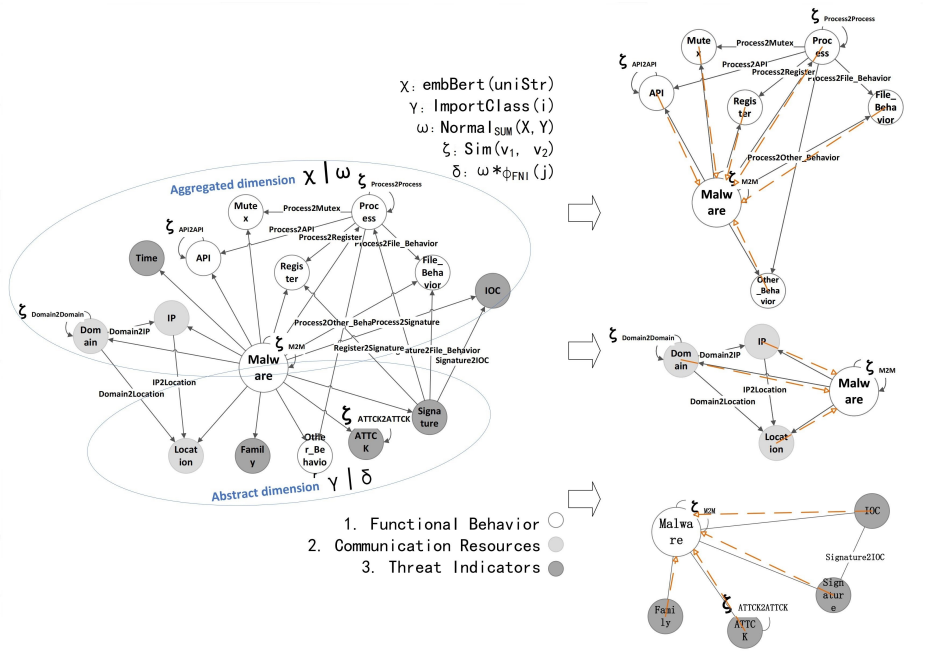


Fig. 3. Graph Embedding Representation Based on Metapath.

various categories. The resulting node attribute γ quantifies the extent of nodes' impact on different classifications.

3.3 Threat Actor Groupal Attribution Based on Meta-Path Graph Embedding

In this research, we propose a graph embedding method based on meta-paths, which improves upon the GraphSAGE algorithm to construct an accurate and efficient APT attribution classification model. This method leverages malware behavior paths defined by domain knowledge, effectively focusing on the most relevant structures in the graph, and captures critical information about nodes and their neighbors through specific aggregation functions. This guided GraphSAGE strategy contributes to the achievement of a more precise and efficient APT attribution classification model.

As shown in Fig. 3, the representation of the symbols corresponds to the descriptions provided in the previous section. It should be noted that ζ is employed for multi-stage graph clustering and is not part of the graph embedding calculations. By utilizing domain meta-paths defined for Functional Behavior, Communication Resources, and Threat Indicators, we partition the graph into different subgraphs aligned with these meta-paths. Subsequently, we apply the

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63749-0_6

GraphSAGE algorithm to extract information surrounding Malware center nodes as described below.

$$h_v^k = \sigma\left(W^k \cdot \text{CONCAT}\left(h_v^{k-1}, \text{AGG}\left(\{h_u^{k-1}, \forall u \in N(v)\}\right)\right)\right) \tag{4}$$

Here, h_v^k is the hidden state of node v at layer k, $N(v)$ is the set of neighboring nodes of v, W^k is the weight matrix at layer k, σ is the activation function, and AGG is the aggregation function.

To combine the advantages of different aggregation strategies and comprehensively capture information from neighboring nodes, we use a hybrid aggregation function that combines direct concatenation, pooling, long short-term memory networks (LSTM), and graph convolutional networks (GCN) to generate different embeddings as follows:

$$\text{AGG}(S) = (\text{SPLICE}(S), \text{POOL}(S), \text{LSTM}(S), \text{GCN}(S)) \tag{5}$$

Here, S is the feature set of neighboring nodes. This hybrid aggregation approach preserves the integrity of the original features while enhancing the extraction of information from neighboring nodes through different strategies.

Next, we combine these graph embedding results with ensemble machine learning and use an auto-sklearn [9] for ensembled model selection and parameter tuning. We explore different machine learning models and parameter configurations to find the best combination. GraphSAGE primarily focuses on structural feature extraction in this process, while ensemble learning is responsible for model selection and parameter optimization. This ensemble method combines multiple machine learning techniques to improve prediction accuracy and generalization.

With this optimization approach, our model retains the information of the graph data structure while achieving effective model selection and optimization. This combination of meta-path-guided graph embedding and ensemble machine learning techniques provides an innovative and efficient solution for APT attribution classification.

4 Evaluation & Discussion

In this chapter, we compare our ontology-based APT malware knowledge graph with existing research on APT-related knowledge graphs. We focus on evaluating whether graph refinement negatively impacts classification performance and comparing different graph computation methods using meta-path graph embeddings. Our goal is to demonstrate the advantages of our domain graph embedding approach based on the refined knowledge graph in APT group recognition tasks.

We collected APT malware from public intelligence sources [12, 6, 21] using VirusTotal [29], excluding samples that were either reused pieces of malware or could not be analyzed in sandboxes. We focused on 10 groups from different countries and regions for experimental verification and obtained labeled samples that were manually verified by security experts. This study also encompasses

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63749-0_6

Table 1. APT Malware Distribution and Corresponding Threat Actor Groups.

Country	Threat Actor Groups	Number
Iran	APT-C-07	139
USA	Equation	464
Turkey	PROMETHIUM	203
Vietnam	APT32	1243
North Korea	Lazarus Group	4028
Russia	BlackEnergy	101
South Korea	Darkhotel	525
India	Patchwork	1380
India	APT-C-35	379
Russia	Carbanak	557
COUNT	10	9019

Table 2. Comparisons with APT Malware Classification Approaches.

Node Type	Finegrained	Refined Knowledge Graph		
		Abstract	*Aggregation*	*All*
API	208061		3935	3935
ATTCK	143	164		164
Location	56262	49722		49722
Domain	3605		1761	1761
Family	3142	717		717
File_Behavior	982223		12039	12039
IOC	27846		107	107
IP	3109741		291169	291169
Malware	9019			9019
Mutex	3161		1655	1655
Other_Behavior	2053	27		27
Process	15516		8633	8634
Register	82104		3850	3850
Signature	481	477		477
All Nodes	4503357			383276
Related Edge	14085227			1249510

the analysis of APT groups with imbalanced samples, ranging from just over 100 samples for the least represented group to over 4,000 samples for the most represented group. The specific distribution of APT malware and their corresponding group lists are shown in Tab. 1.

The fine-grained knowledge graph constructed through the ontology model in this paper contains various types of nodes, such as API, ATT&CK, location, Domain, etc., with a total of approximately 4.5 million nodes. Through multi-stage graph clustering methods, these nodes are significantly reduced in the aggregation dimension, ultimately reducing the total number of nodes in the "Refined Knowledge Graph" to approximately 380,000, as shown in Tab. 2.

4.1 Discussion1: Whether the Key Information of APT Malware Knowledge Graph Attribution is Reduced After Refining

We visualize the similarity relationship between two malware samples by considering the associated nodes around them, using a relationship-based layout as depicted in Fig. 4. The results show significant associations within the same group. This highlights the effectiveness of the ontology model we propose for constructing the APTMalKG, especially in classifying APT groups.

Furthermore, the analysis indicates that the Refined Knowledge Graph's performance is on par with that of the Fine-grained Knowledge Graph, showing a slight advantage in differentiating certain groups. For instance, the distinction between PROMETHIUM (green) and Lazarus Group (pink) is more distinct, and the separation of Darkhotel (blue) from other groups is clearer. This suggests that while we reduced the graph's complexity, essential features for groupal analysis have been effectively retained.

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63749-0_6

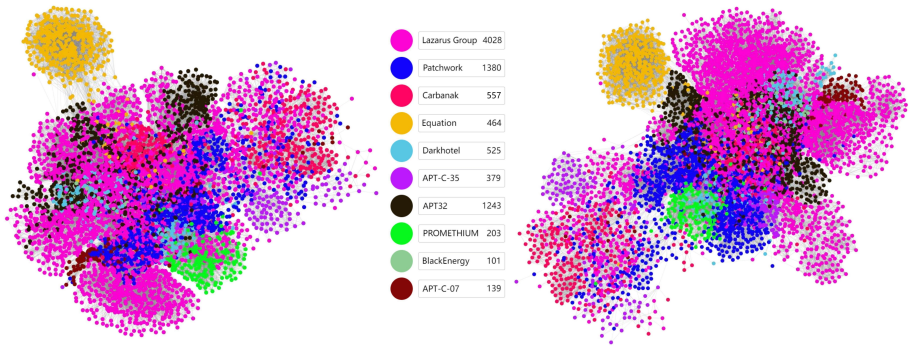


Fig. 4. Comparative Analysis of Association Effects in APT Threat Actor Groups: Fine-Grained vs. Refined Knowledge Graphs of Malware Samples. (left: the fine-grained knowledge graph, right: refined knowledge graph, nodes: individual malware samples, colors: 10 group categories.)

4.2 Discussion2: What is the Effect of Refined APT Malware Knowledge Graph Attribution Analysis

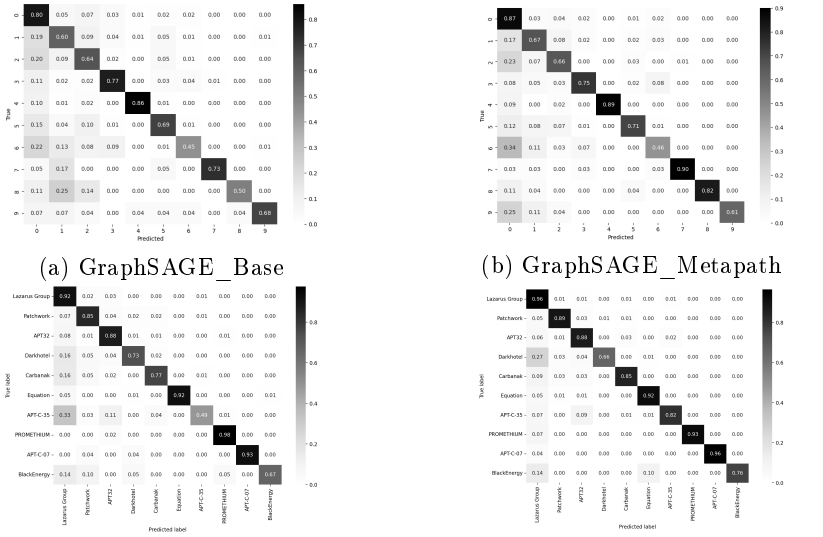
In this section, we delve into the attribution analysis of refined APT malware knowledge graphs. Through a comprehensive comparison of various graph computation methods and ensemble machine learning techniques, we showcase their performance in APT malware attribution analysis.

Table 3. Comparison with Other Graph Calculation Methods on F1 score.

Group	RGCN	Hetro-GNN	Graph-SAGE	SAGE-Metapath	Ensemble (our)
Lazarus Group	0.8151	0.8351	0.7984	0.834	0.9373
Patchwork	0.6909	0.705	0.6277	0.6996	0.9274
APT32	0.7	0.692	0.6186	0.6891	0.8722
Darkhotel	0.5161	0.5579	0.7089	0.7264	0.8128
Carbanak	0.6827	0.6698	0.9278	0.9326	0.8768
Equation	0.8432	0.8508	0.6712	0.7199	0.9348
APT-C-35	0.609	0.5854	0.5979	0.5998	0.8050
PROMETHIUM	0.8648	0.8266	0.8919	0.8981	0.9512
APT-C-07	0.84	0.8727	0.6455	0.8679	1
BlackEnergy	0.6444	0.5589	0.7331	0.7947	0.8648
$F1_{Macro}$	0.72062	0.71542	0.7221	0.77621	0.8982
Test ACC	0.7533	0.7572	0.7641	0.8021	0.9116
Test AUC	0.9253	0.9205	0.9506	0.9612	0.9899

First, Tab. 3 data illustrates the significant performance of our specific method, which combines GraphSAGE embeddings based on metapath with an ensemble of machine learning models (denoted as SAGE_Metapath+Ensemble), across all APT groups. Particularly, in the cases of "Lazarus Group" and "Equation," this combined method outperforms others with F1 scores of 0.9373 and 0.9348,

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63749-0_6



(a) GraphSAGE_Base (b) GraphSAGE_Metapath

(c) SAGE_Base+Ensemble (d) SAGE_Metapath+Ensemble(*)

Fig. 5. Confusion Matrices for Test Data Set with Model Selection (* indicates the final chosen model)

respectively. This indicates the effectiveness of our composite approach in identifying complex and covert APT activities compared to traditional methods like RGCN, Hetero-GNN, and standalone GraphSAGE.

Table 4. Comparison with Other Integrated Machine Learning Methods.

Model	ACC	AUC	F1_Macro
GraphSAGE_XGBoost	0.8620	0.9802	**0.8286**
GraphSAGE_CatBoost	0.8242	0.9731	0.7652
GraphSAGE_LightGBM	**0.8624**	**0.9823**	0.8274
SAGEMetapath_XGBoost	**0.9082**	0.9874	**0.8927**
SAGEMetapath_CatBoost	0.9054	0.9880	0.8856
SAGEMetapath_LightGBM	0.9060	**0.9884**	0.8884
Our	**0.9116**	**0.9899**	**0.8982**

Table 5. Comparisons with APT Malware Classification Approaches.

Model	ACC	AUC	F1_Macro
Shudong et al. [16]	0.7401	0.9497	0.7610
Adem et al. [28]	0.8286	0.9414	0.8092
Hrishabh et al. [27]	0.8054	0.9038	0.7850
Do Xuan et al. [7]	**0.8398**	**0.9541**	**0.8217**
Our	**0.9116**	**0.9899**	**0.8982**

We further compare the performance of our ensemble machine learning method in Tab. 4. Our method, which is a combination of SAGE_Metapath and Ensemble learning, excels in terms of accuracy (ACC), area under the curve (AUC), and macro-average F1 score ($F1_{Macro}$), highlighting its efficiency and accuracy when dealing with complex datasets.

The confusion matrices in Fig. 5 further confirms the superiority of our method. The SAGE_Metapath+Ensemble model performs best in correctly classifying APT groups, demonstrating its highly precise classification capabilities.

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63749-0_6

Our research indicates that a comprehensive approach combining the strengths of multiple advanced algorithms enhances overall prediction accuracy and robustness. The SAGE-Metapath algorithm improves feature representation of malicious software samples, effectively capturing key features and intricate behavioral patterns of APT malware, thus enhancing classification accuracy. Moreover, our method exhibits outstanding performance across diverse APT group samples, demonstrating its broad applicability and robustness. This is particularly crucial in the context of the increasing diversity and complexity of APT malware.

4.3 Discussion3: How meaningful is our research in comparison with similar studies in the field of APT malware analysis

Nowadays, existing research on related knowledge graphs primarily relies on manual intelligence sources, requiring manual processing and often lacking in-depth behavioral characteristics. We have completed the data collection on the malware side through APT malware analysis. This connects detailed low-level behavioral analysis with broader infrastructure and threat intelligence information. Despite the raw nature of the data obtained from samples, which has not undergone manual processing, our graph data encompasses a more comprehensive range of dimensions available at the sample level. After constructing an extensive dataset, we refined it to capture critical groupal classification features.

In the realm of APT group classification research, we conducted experiments on our dataset using feature processing techniques from established multi-classification models in malware studies. As shown in Tab. 5, this highlights the comparison models' limitations in classifying high-level APT groups, emphasizing the need for more targeted research. And our methodology proves effective in APT group classification.

Due to constraints like the experimental environment and parameters, our study focused on basic model replication without parameter tuning or addressing unknowns, potentially not reaching the reference models' optimal performance. Nevertheless, considering the highly unbalanced nature of the APT group dataset, these results remain credible, especially given their reliance on superficial malware features, possibly missing complex APT attack behavior patterns. Additionally, these methods may struggle with malware's semantic and structural information, limiting their effectiveness in complex APT scenarios. Our approach, integrating deep graph networks and ensemble learning techniques, thoroughly analyzes APT malware's multidimensional features, showcasing its potential in complex APT attack classification.

5 CONCLUSION

In summary, refining the knowledge graph reduces complexity while preserving attribution information. This research aims to enhance APT malware analysis and attribution, aiding network security analysts in countering threats ef-

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63749-0_6

ficiently. The APTMalKG is constructed through graph clustering and optimization, utilizing metapath-enhanced graph embedding methods. It integrates various data dimensions, enhancing attribution accuracy by capturing malware behaviors comprehensively. In the future, we plan to enhance the model's ability to generalize to new categories by combining multiple approaches such as embeddings and attribute learning. This approach empowers analysts with a more refined knowledge graph, bolstering their capabilities against APT threats and strengthening network security.

References

1. Malware attribute enumeration and characterization (maec). https://maecproject.github.io/, accessed: 2023-11-11
2. Balan, G., Gavriluţ, D.T., Luchian, H.: Using api calls for sequence-pattern feature mining-based malware detection. In: Information Security Practice and Experience. pp. 233–251. Springer (2022)
3. Busch, J., Kocheturov, A., Tresp, V., Seidl, T.: Nf-gnn: Network flow graph neural networks for malware detection and classification. In: Proceedings of the 33rd International Conference on Scientific and Statistical Database Management. p. 121–132. Association for Computing Machinery (2021)
4. Chang, H.Y., Yang, T.Y., Zhuang, C.J., Tseng, W.L.: Ransomware detection by distinguishing api call sequences through lstm and bert models. The Computer Journal **13**, 5439 (2023)
5. Cremer, F., Sheehan, B., Fortmann, M., Kia, A.N., Mullins, M., Murphy, F., Materne, S.: Cyber risk and cybersecurity: a systematic review of data availability. The Geneva Papers on Risk and Insurance - Issues and Practice **47**, 698–736 (2022)
6. CyberMonitor, Robert Haist, k., et al.: Apt & cybercriminals campaign collection. GitHub repository (2022), https://github.com/CyberMonitor/APT_CyberCriminal_Campagin_Collections
7. Do Xuan, C., Huong, D.: A new approach for apt malware detection based on deep graph network for endpoint systems. Applied Intelligence **52**(12), 14005–14024 (2022)
8. Dutta, S., Rastogi, N., Yee, D., Gu, C., Ma, Q.: Malware knowledge graph: A comprehensive knowledge base for malware analysis and detection. In: 2021 IEEE Network Security and Privacy Protection International Conference (NSPW) (2021)
9. Feurer, M., et al.: auto-sklearn: Automated machine learning toolkit (2023), https://automl.github.io/auto-sklearn/master/, gitHub repository
10. Hasan, M.M., Islam, M.U., Uddin, J.: Advanced persistent threat identification with boosting and explainable ai. SN Computer Science **4**, 271–279 (2023)
11. Kiesling, E., Ekelhart, A., Kurniawan, K., Ekaputra, F.: The sepses knowledge graph: An integrated resource for cybersecurity. In: The Semantic Web - ISWC 2019. pp. 198–214. Springer (2019)
12. Kiran Bandla, S.C.: Aptnotes data. GitHub repository (2021), https://github.com/aptnotes/data
13. Lee, K., Lee, J., Yim, K.: Classification and analysis of malicious code detection techniques based on the apt attack. Applied Sciences **13**, 2894 (2023)
14. Li, S., Zhou, Q., Zhou, R., Lv, Q.: Intelligent malware detection based on graph convolutional network. The Journal of Supercomputing **78**, 4182–4198 (2022)

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63749-0_6

15. Li, S., Zhang, Q., Wu, X., Han, W., Tian, Z.: Attribution classification method of apt malware in iot using machine learning techniques. Security and Communication Networks **2021** (2021)
16. Li, S., Zhang, Q., Wu, X., Han, W., Tian, Z., Yu, S.: Attribution classification method of apt malware in iot using machine learning techniques. Security and Communication Networks **2021**, 12 (2021)
17. Li, Z., Zeng, J., Chen, Y., Liang, Z.: Attackg: Constructing technique knowledge graph from cyber threat intelligence reports. In: Computer Security - ESORICS 2022. pp. 589–609. Springer (2022)
18. MLG at Neo4j: Community detection (2022), https://neo4j.com/docs/graph-data-science/current/algorithms/community/
19. Moon, H.J., Bu, S.J., Cho, S.B.: Directional graph transformer-based control flow embedding for malware classification. In: International Conference on Artificial Neural Networks. pp. 506–518. Springer (2021)
20. Peng, C., Xia, F., Naseriparsa, M., Osborne, F.: Knowledge graphs: Opportunities and challenges. Artificial Intelligence Review **56**, 13071–13102 (2023)
21. RedDrip7: Apt_digital_weapon: Indicators of compromise (iocs) collected from public resources and categorized by qi-anxin. GitHub repository (2022)
22. Ren, Y., Xiao, Y., Zhou, Y., Zhang, Z., Tian, Z.: Cskg4apt: A cybersecurity knowledge graph for advanced persistent threat organization attribution. IEEE Transactions on Knowledge and Data Engineering **35**, 5695–5709 (2023)
23. Renz, M., Kröger, P., Koschmider, A., Landsiedel, O., de Sousa, N.T.: Cross domain fusion for spatiotemporal applications: taking interdisciplinary, holistic research to the next level. Informatik Spektrum **45**, 271–277 (2022)
24. Sahoo, D.: Cyber Threat Attribution with Multi-View Heuristic Analysis, pp. 271–277. Springer (2022)
25. Sharma, A., Gupta, B.B., Singh, A.K., Saraswat, V.K.: Advanced persistent threats (apt): evolution, anatomy, attribution and countermeasures. Journal of Ambient Intelligence and Humanized Computing **14**, 9355–9381 (2023)
26. Sikos, L.F.: Cybersecurity knowledge graphs. Knowledge and Information Systems **65**, 3511–3531 (2023)
27. Soni, H., Kishore, P., Mohapatra, D.P.: Opcode and api based machine learning framework for malware classification. In: 2022 2nd International Conference on Intelligent Technologies (CONIT). pp. 1–7 (2022)
28. Tekerek, A., Yapici, M.M.: A novel malware classification and augmentation model based on convolutional neural network. Computers & Security **112**, 102515 (2022)
29. VirusTotal: Virustotal: Analyse suspicious files and urls to detect malware. Website (2022), https://www.virustotal.com/
30. Wai, F.K., Thing, V.L.L.: Clustering based opcode graph generation for malware variant detection. In: 2021 18th International Conference on Privacy, Security and Trust (PST). pp. 1–11 (2021)
31. Wei, C., Li, Q., Guo, D., Meng, X.: Toward identifying apt malware through api system calls. Security and Communication Networks **2021**, 8077220 (2021)
32. Wu, X.W., Wang, Y., Fang, Y., Jia, P.: Embedding vector generation based on function call graph for effective malware detection and classification. Neural Computing and Applications **34**, 8643–8656 (2022)
33. Xuan, C.D., Dao, M.H.: A novel approach for apt attack detection based on combined deep learning model. Neural Computing and Applications **33**, 13251–13264 (2021)

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63749-0_6