Knowledge-infused and Explainable Malware Forensics

Neha Mohan Kumar, Sheikh Rabiul Islam
Rutgers University - Camden
neha.m@rutgers.edu, sheikh.islam@rutgers.edu

Abstract

Despite considerable progress in malicious software forensics, the challenge of accurate attribution, formulation of appropriate response and mitigation strategies, and ensuring the interpretability of deep learning methods persists. While being less flexible and robust to noise compared to deep learning models, Knowledge Graphs are natively developed to be explainable and are a promising solution for exploring new features and relations, and enhancing understandability of decisions. In this work, we aim to develop an explainable malware classifier which can classify PE executable as malign or benign, by infusing external knowledge using Knowledge Graph (KG). We enrich our Knowledge Graph using MITRE Attack ontology (i.e., domain knowledge) and EMBER dataset and utilize Graph2Vec algorithm to embed KG knowledge into our classifier. We found that our classifier yields satisfactory results while maintaining a high level of explainability.

Introduction

The relentless evolution and increasing sophistication of malware in the cybersecurity landscape continually challenge the development of effective defense mechanisms. Traditional black-box models often fall short in providing transparent and understandable explanations for their decisions, necessitating innovative approaches to bridge this gap. Our research investigates the synergistic integration of Knowledge Graphs (KGs) and Explainable Artificial Intelligence (XAI) to conceive a Knowledge-Infused Malware Classifier.

Cybersecurity research has consistently highlighted the critical role of transparent and interpretable models in enhancing interpretability and trust of decision from it. Towards tackling this challenge, the MITRE Corporation's ATT&CK (Adversarial Tactics, Techniques, and Common Knowledge) framework (Roy et al. 2023) emerges as a foundational resource. MITRE ATT&CK provides a comprehensive and structured ontology of cyber threats, mapping out the tactics, techniques, and procedures (TTPs) employed by adversaries. By incorporating external knowledge, our Knowledge-Infused Malware Classifier gains a deeper understanding of the diverse strategies employed by malicious entities, enhancing its interpretability and performance in real-world scenarios.

To train and evaluate the robustness of our classifier, we utilize the EMBER dataset (Anderson and Roth 2018), a widely recognized and extensively studied repository of diverse malware samples. The EMBER dataset encompasses a diverse set of features extracted from a wide range of malicious and benign entities. Leveraging this dataset for training allows our model to generalize effectively across various malware types, ensuring its adaptability to emerging threats.

The Graph2Vec (Narayanan et al. 2017) algorithm serves as a cornerstone in our methodology, facilitating the embedding of knowledge from the MITRE ATT&CK framework into our classifier. Graph2Vec, a graph embedding technique, captures the structural information present in Knowledge Graphs, enabling the infusion of explicit knowledge into the decision-making process of the classifier. The resulting model not only excels in its ability to distinguish between malign and benign entities but also provides human-comprehensible explanations for its decisions, aligning with the growing need for transparency and accountability in the ever-evolving landscape of cybersecurity.

In summary, this research contributes to the advancement of malware classification by integrating Knowledge Graphs and Explainable AI. The incorporation of external knowledge from the MITRE ATT&CK framework and the utilization of the EMBER dataset enhance the interpretability and generalizability of our Knowledge-Infused Malware classifier. The subsequent sections of this paper delve into the background work, detailed methodology, experimental setup, and results, providing a comprehensive exploration of the contributions and implications of our approach in addressing the challenges posed by contemporary cyber threats.

Background

The digital era has brought unprecedented advancements in technology, transforming the way individuals, businesses, and governments operate. Alongside these innovations, however, has emerged a formidable and dynamic threat landscape characterized by the proliferation of sophisticated and evolving malware. In response, the field of cybersecurity has become a paramount concern, with a constant need for innovative strategies to detect, analyze, and mitigate the impact

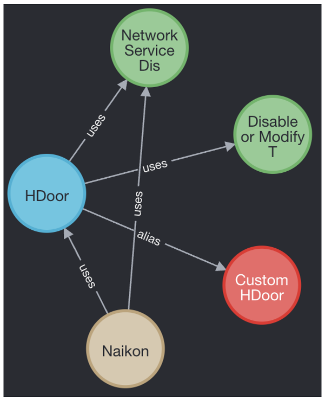

Figure 1: Software node (Light Blue) from MITRE Ontology associated with techniques used (Light Green), Alias (Red) and respective cyber Group (Beige)

of malicious software.

The traditional reliance on signature-based detection methods is increasingly insufficient as malware authors employ polymorphic and obfuscation techniques to evade detection. As a result, cybersecurity researchers and practitioners have shifted towards leveraging advanced machine learning models to keep pace with the dynamic nature of malware. However, the widespread use of complex black-box models poses a significant challenge, as their decision-making processes lack transparency and interpretability, hindering the ability to understand and trust their outputs.

The importance of transparent and interpretable models in cybersecurity has been emphasized by numerous studies (Islam et al. 2019). Transparent models not only foster trust among users but also enable collaboration and knowledge-sharing within the cybersecurity community. Interpretable models are essential for understanding the rationale behind decisions, aiding in the identification of false positives and facilitating the refinement of detection strategies.

Knowledge Graphs (KGs) have proven to be powerful tools for representing and organizing complex relationships within a domain. KGs encode knowledge in a structured format, connecting entities through defined relationships, allowing for a rich representation of semantic information. The integration of KGs into machine learning models has shown promise in enhancing interpretability by incorporating explicit knowledge. This synergy between KGs and machine learning has been explored in various domains, including natural language processing and cybersecurity (Sikos 2023), demonstrating the potential for improving model transparency and performance.

Explainable Artificial Intelligence (XAI) is another key area of research that seeks to demystify complex machine learning models and make their decisions more understandable to end-users. Numerous XAI techniques, such as LIME (Local Interpretable Model-agnostic Explanations) (Zafar and Khan 2021) and SHAP (SHapley Additive exPlanations) (Lundberg and Lee 2017), have been developed to provide post-hoc explanations for black-box models. These techniques generate simplified and human-comprehensible explanations for model predictions, contributing to the overall transparency of the decision-making process.

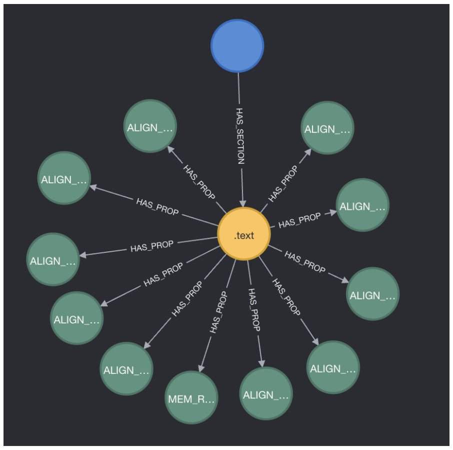

Figure 2: Each data point(Blue) associated with its sections (Yellow) and each section associated with it's properties (Green)

The research presented in this paper aims to address the challenges posed by evolving malware threats through the integration of Knowledge Graphs and Explainable AI. By leveraging external knowledge from the MITRE ATT&CK framework and employing the EMBER dataset, our approach seeks to enhance the interpretability and performance of malware classification models.

Methodology

The proposed approach consists of three major stages: A) Data Integration and Knowledge Graph Construction, B) Graph Representation Learning, and C) Building the Malware Classifier:

1. Data Integration and Knowledge Graph Construction

Our methodology begins with the integration of two distinct datasets – the MITRE dataset and the Ember dataset. Simultaneously, the Ember dataset, focused on executable files, is ingested into the knowledge graph, capturing structural elements such as sections, imports, exports, and data directories. The Neo4j Graph Data Science library aids in predicting new relationships between nodes, enhancing the inter-connectedness of our Knowledge Graph.

1.1 MITRE Dataset Integration The MITRE dataset, known for its comprehensive cyber threat intelligence, is seamlessly integrated into a Neo4j knowledge graph. Entities representing Groups, Software, Tools, and Techniques form the backbone of our graph, interconnected to capture contextual dependencies as seen in Figure 1. This process captures not only individual entities but also the intricate relationships among Groups, Software, Tools, and Techniques. The resulting Knowledge Graph becomes a comprehensive repository of cyber threat intelligence.

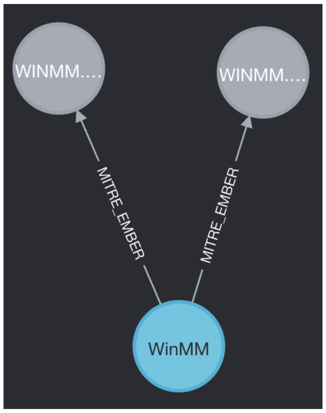

Figure 3: Software (Light Blue) imported (Grey) by one of the executables in EMBER dataset linked by the link prediction algorithm

1.2 Ember Dataset Integration The seamless amalgamation of the Ember dataset, which centers on executable files, into our Knowledge Graph is facilitated by a meticulously crafted integration script. This script adeptly captures pivotal structural components intrinsic to executable files, encompassing aspects such as segments, dependencies, exports, and data directories. This infusion of structural intricacies enhances the graph's depth with vital details essential for the nuanced classification of malware.

Within the training dataset of our classifier, each discrete data point finds its place in the Knowledge Graph as a unique node. It is noteworthy that the visual representation in Figure 2 encapsulates these data points through distinct nodes denoted by a distinct color palette. Subsequent to the assimilation of these data points, we proceed to encompass all JSON fields originating from the dataset within the fabric of the Knowledge Graph. This meticulous integration ensures that the intricate JSON structure finds its reflection in the interconnected relationships among nodes in the Knowledge Graph. This holistic approach not only amplifies the comprehensiveness of the graph but also lays the groundwork for a robust and nuanced malware classification framework.

1.3 Predicting New Relationships Between the Nodes Upon augmenting our Knowledge Graph (KG) with both EMBER and MITRE datasets, the subsequent imperative involves establishing connections between nodes from the EMBER dataset and their counterparts in the MITRE dataset. This pivotal step constitutes the initial stride in integrating external knowledge into our classifier. To facilitate this process, we meticulously prepare our training set, which serves as the foundation for training our link prediction model. The creation of the training set involves assessing the similarity between the data of individual nodes within the EMBER dataset and those within the MITRE dataset.

The computation of similarity is achieved through the application of the Dice (Sorensen) coefficient ((Dice 1945), (Sørensen 1948)). Specifically, given two strings A and B, the Dice coefficient is calculated as follows (Formula 1).

$$D(A,B) = \frac{2 * |A \cap B|}{|A| + |B|} \tag{1}$$

We establish a threshold at 0.67 as they demonstrated robust compatibility and add connections between nodes that exhibit similarity scores surpassing this defined threshold.

Subsequently, our link prediction model undergoes training, employing a Multi-Layer Perceptron (MLP) (Yaghi et al. 2020). In the feature engineering step, node properties existing in the input graph or added during the pipeline process are utilized. For each node in a potential link, the node embeddings' values (as discussed in Section 2.1) are concatenated in the pre-configured order, forming a vector 'f.' This process entails combining the feature vector of the source node, denoted as s = [s1, s2, ..., sd] with the feature vector of the target node, denoted as t = [t1, t2, ..., td]

The trained model then uses the *Approximate Search* strategy which leverages the K-Nearest Neighbors algorithm with our model's prediction function as its similarity measure to trade off lower runtime for accuracy. Accuracy in this context refers to how close the result is to the very best new possible links according to our models predictions, i.e. the best predictions that would be made by exhaustive search.

The initial set of considered links for each node is picked at random and then refined in multiple iterations based of previously predicted links. The algorithm returns the probability of a link for each node pair. We specify threshold to include only predictions with probability greater than 55%.

Noteworthy is our observation that the trained link prediction model attains a state-of-the-art performance, effectively integrating MITRE knowledge into the EMBER dataset. Figure 3 shows a sample of new links created by the trained model.

2. Graph Representation Learning

We employ the Graph2Vec algorithm to generate embeddings from the integrated knowledge graph. The generated embeddings serve as a dual-layered representation, capturing both structural insights derived from the graph's topology and semantic understanding learned from the underlying cybersecurity intelligence derived from MITRE dataset. Graph2Vec traverses the entire knowledge graph and generate embeddings for every data point in the dataset. These embeddings serve as one of the features for training our classifier.

3.Building the Malware Classifier

Employing the Gradient Boosting Machine Framework, LightGBM (Ke et al. 2017), we conduct experiments encompassing diverse tree-based machine learning techniques. LightGBM's distinctive advantage lies in its adoption of leaf-wise tree growth as it uses an algorithm which involves selecting the leaf with the maximum difference in loss for tree expansion. Consequently, the utilization of leaf-wise or Best-first algorithms empowers the learning to attain lower loss compared to traditional level-wise algorithms.

Feature Set	**Accuracy**	**Recall**	**F1 Score**	**Precision**
Without KG embeddings	0.93	0.92	0.93	0.95
Before MITRE Integration	0.92	0.91	0.93	0.95
After MITRE Integration	0.93	0.91	0.93	0.95

Table 1: Comparison of performance metrics

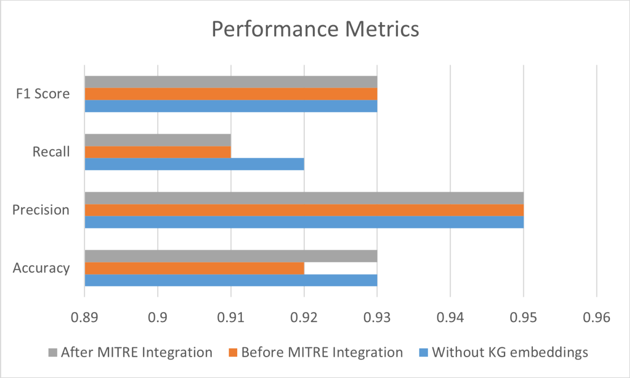

Figure 4: Comparison of performance metrics

Experimentation & Results

The training phase of experiment was executed in three distinct phases:

1. Initial Training without Node Embeddings: In the first phase, we trained the classifier exclusively on numerical fields from the EMBER Dataset, omitting the incorporation of node embeddings.
2. Node Embeddings Before Knowledge Graph Enrichment: The second phase involved the inclusion of node embeddings calculated prior to enhancing the Knowledge Graph with the MITRE dataset.
3. Node Embeddings After Knowledge Graph Enrichment: The third phase encompassed the calculation of node embeddings subsequent to the Knowledge Graph being enriched with the MITRE dataset.

Conducting experiments on approximately 18,500 rows of the EMBER dataset, with an 80-20% train-test split, our results, summarized in Table 1 (also in Figure 4), present accuracy and other performance metrics across all phases. Throughout the process of hyperparameter tuning, we explored diverse techniques related to data sampling, learning rates, the number of leaves, and types of boosting. Notably, our observations indicated that Gradient Boosted Decision Trees consistently outperformed Random Forests and CART algorithms in terms of accuracy. Importantly, we successfully constructed a highly explainable model using the Knowledge Graph, with little to no compromise in accuracy and other metrics.

The infusion of MITRE knowledge contributed to a slightly higher accuracy compared to scenarios without such knowledge infusion. This suggests that external knowledge incorporation holds the potential to enhance the model's interpretability without sacrificing accuracy. Through simple Knowledge Graph queries, we derived insightful observations. For instance, we identified that many correctly predicted malign executables shared similar *imports*, and a majority of malwares exhibited no *exports*. These findings contribute to the classifier's high explainability, emphasizing the utility of Knowledge Graphs in infusing external knowledge and extracting meaningful insights from the model.

Conclusion

In addressing the challenges of interpretability in the field of malware software forensics, this research looks to build a Malware Classifier which offers high explainability with little or no compromise to the model accuracy. Leveraging Knowledge Graphs (KG) as a foundational element, our approach involves the infusion of external knowledge through the integration of the MITRE Attack ontology and the EMBER dataset. The enriched KG serves as a comprehensive repository of cyber intelligence, capturing intricate relationships and dependencies among entities, thereby enhancing the model's contextual understanding. We look to establish link between MITRE and EMBER data through exploration of new relationships between nodes from the both datasets. Leveraging the Graph2Vec algorithm, embeddings generated from the integrated KG contribute dual-layered representations, incorporating both structural insights and semantic understanding derived from cybersecurity intelligence.

This novel approach not only enhances model accuracy but also ensures a high level of explainability, a critical facet often lacking in deep learning methodologies. Furthermore, the ability to derive insightful observations through simple Knowledge Graph queries reinforces the utility of Knowledge Graphs in extracting meaningful insights. This research not only contributes to the development of a sophisticated and interpretable malware classifier but also sheds light on the promising avenues of leveraging external knowledge through Knowledge Graphs for improved cyber threat intelligence and decision-making.

Future Work

Future work could focus on enhancing the scalability of the Knowledge Graph, incorporating additional cybersecurity intelligence sources, and exploring the integration of temporal aspects for dynamic malware detection.

References

Anderson, H. S., and Roth, P. 2018. Ember: An open dataset for training static pe malware machine learning models.

Dice, L. R. 1945. Measures of the amount of ecologic association between species. *Ecology* 26(3):297–302.

Islam, S. R.; Eberle, W.; Ghafoor, S. K.; Siraj, A.; and Rogers, M. 2019. Domain knowledge aided explainable artificial intelligence for intrusion detection and response. *arXiv preprint arXiv:1911.09853*.

Ke, G.; Meng, Q.; Finley, T.; Wang, T.; Chen, W.; Ma, W.; Ye, Q.; and Liu, T.-Y. 2017. Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems* 30:3146–3154.

Lundberg, S., and Lee, S.-I. 2017. A unified approach to interpreting model predictions.

Narayanan, A.; Chandramohan, M.; Venkatesan, R.; Chen, L.; Liu, Y.; and Jaiswal, S. 2017. graph2vec: Learning distributed representations of graphs. *arXiv preprint arXiv:1707.05005*.

Roy, S.; Panaousis, E.; Noakes, C.; Laszka, A.; Panda, S.; and Loukas, G. 2023. Sok: The mitre attck framework in research and practice.

Sikos, L. F. 2023. Cybersecurity knowledge graphs. *Knowledge and Information Systems* 1–21.

Sørensen, T. 1948. *A Method of Establishing Groups of Equal Amplitude in Plant Sociology Based on Similarity of Species Content and Its Application to Analyses of the Vegetation on Danish Commons*. Biologiske skrifter. Munksgaard in Komm.

Yaghi, R. I.; Faris, H.; Aljarah, I.; Al-Zoubi, A. M.; Heidari, A. A.; and Mirjalili, S. 2020. Link prediction using evolutionary neural network models. *Evolutionary Machine Learning Techniques: Algorithms and Applications* 85–111.

Zafar, M. R., and Khan, N. 2021. Deterministic local interpretable model-agnostic explanations for stable explainability. *Machine Learning and Knowledge Extraction* 3(3):525–541.