Integrating Ontology and Graph Neural Network for Explainable Malware Detection

Monday Onoja*

Department of Applied Informatics, Comenius University in Bratislava, Mlynská dolina, 842 48 Bratislava, Slovakia

Abstract

Modern machine learning (ML) models for malware detection offer high predictive power but often lack transparency, hindering trust and interpretability. Ontology-driven representation provides a formal, structured way to model malware behavior that is both machine-readable and human-interpretable. This research proposes a dynamic malware ontology leveraging standard vocabularies from Malware Attribute Enumeration and Characterization (MAEC) and Structured Threat Information Expression (STIX), aimed at capturing behavioral features extracted via dynamic analysis. The structured dataset resulting from the ontology will serve as input to Graph Neural Networks (GNNs) and DeepProbLog to produce explainable detection results. This work addresses key challenges in explainability, semantic representation, and robust malware classification, contributing a novel dataset, ontology, and interpretability framework for cybersecurity applications.

Keywords

Malware detection, Malware ontology, Explanability, Graph Neural Network

1. Introduction

Malware continues to pose a significant threat in our increasingly interconnected world. Modern machine learning (ML) techniques have shown great promise in detecting malicious code, but these models often operate as black boxes, making their decisions difficult to interpret and trust. As a result, there is a growing demand for explainable malware detection systems that not only achieve high accuracy but also provide human-understandable insights. Ontologies provide a structured and semantically rich means to represent domain knowledge. In the context of malware detection, an ontology-based approach offers a consistent, machine- and human-interpretable representation of malware behaviors, artifacts, and threat patterns. Existing malware ontologies, however, often rely solely on static analysis features and lack coverage of dynamic behaviors that are crucial for detecting sophisticated malware employing evasion techniques. Static malware features (data obtained through static analysis) have been useful for malware detection, but their limitations with respect to obfuscation [1], evasion tactics [2], and lack of understanding of malware behaviors [3] underscore the need for dynamic features (data obtained through dynamic malware analysis) in mitigating malware attacks. There exists a standard called MAEC[1] (Malware Attribute Enumeration and Characteristics), created for sharing these analysis results. This standard is starting to be used among experts and is considered a fundamental standard for sharing. However, to the best of our knowledge, there has not yet been an ontological representation created for the MAEC standard that would be suitable for data sharing. Several efforts have been made to develop ontologies for the cybersecurity domain [4, 5], but these initiatives have remained largely outdated and either do not incorporate the MAEC standard or only do so at a very superficial level. For data obtained through static analysis, the PE Malware Ontology [6] was created, which partially uses the taxonomy from the MAEC standard but also adds its own attributes. The version of the ontology presented here includes a taxonomy for storing data obtained through both dynamic and static analysis and fully utilises the MAEC taxonomy. This study combines MAEC

Doctoral Consortium of the 22nd International Conference on Principles of Knowledge Representation and Reasoning (KR 2025 DC), November 11-17, 2025, Melbourne, Australia

monday.onoja@fmph.uniba.sk (M. Onoja)

0000-0003-2119-170X (M. Onoja)

[1] https://maecproject.github.io/releases/5.0/MAEC_Core_Specification.pdf

and STIX languages for describing and representing malware actions, artifacts, and threat patterns for ontology-based knowledge representation that are suitable for the integration of results obtained from static and dynamic malware analysis (hybrid features). Aligned with the MITRE ATT&CK[2] framework, this ontology enables the definition of malware techniques and tactics.

This research aims to bridge that gap by developing an integrated malware ontology enriched with dynamic features derived from live malware analysis. The ontology will leverage standard vocabularies from MAEC and STIX to formalize the representation. By integrating this structured representation with Graph Neural Networks (GNNs) and DeepProbLog as explainable models, the work will advance both detection performance and interpretability.

2. Research Plan and Objectives

The proposed research seeks to create a pipeline that connects dynamic malware analysis, ontology-based representation, and explainable AI (XAI) methods for malware detection. The following key objectives guide this research:

2.1. Ontology Development

We will create a semantically rich and comprehensive ontology that formally represents malware behavior, actions, and threat patterns. The ontology will incorporate dynamic malware attributes such as runtime behaviors and system-level interactions using standardized vocabularies from MAEC and STIX. This structured representation will enable both humans and machines to understand the context of malware operations more clearly and provide a basis for interpretable reasoning. MAEC (Malware Attribute Enumeration and Characterization) and STIX (Structured Threat Information Expression) are standardized vocabularies for malware and threat intelligence. MAEC provides a structured language for describing low-level malware behaviors, while STIX adds higher-level context and granularity, thereby enriching MAEC classes. Used together, they enhance the ontology's expressiveness and interoperability, and their wide adoption and support across analysis tools make them particularly suitable for our approach.

2.2. Dynamic Malware Analysis

To enhance the ontology with real-world behavioral data, this step involves performing dynamic analysis on live malware samples. Using Cuckoo Sandbox, the malware is executed in a controlled environment to capture runtime features such as API calls, file manipulations, registry changes, and network activity. These behaviors offer valuable insights into how malware interacts with a system, especially those that evade static detection through obfuscation or encryption.

2.3. Ontology-based Dataset Construction

The dynamically extracted features will be mapped to the ontology and used to generate a structured dataset. This dataset will contain instances represented as graph-like structures suitable for input into machine learning models. The process involves transforming unstructured behavioral logs into semantically labeled data points, thus enabling downstream algorithms to learn from meaningful, domain-grounded representations.

2.4. Explainable Detection Models

In the final stage, the structured dataset will be used to train Graph Neural Networks (GNNs) and DeepProbLog, a probabilistic logic programming framework. GNNs are well-suited for learning from graph-structured data (e.g., ontology instances) [7], while DeepProbLog allows for symbolic reasoning

[2]https://attack.mitre.org

with probabilistic inference [8]. Together, these models will support interpretable classification decisions by associating predictions with human-understandable rules, dependencies, or paths in the graph making the malware detection process more transparent and trustworthy.

This integration of the symbolic structure of the ontology and sub-symbolic learning enables transparent and logically grounded explanations for model predictions. Through this combination, our approach not only aims to improve malware classification performance but also provides rule-based explanations for why a sample was labeled as malicious or benign thus enhancing the trust, auditability, and human interpretability of AI-driven security systems [9, 10].

This research also aim to address the following questions:

- RQ1: Which dynamic features are most suitable (or unsuitable) for representing malware behavior?
- RQ2: To what extent does integrating these features into a formal ontology produce a structured and expressive dataset?
- RQ3: How effective are GNNs and DeepProbLog in providing interpretable decisions based on ontological data, and what are their limitations?

3. Current Progress

We have developed MAECO, an ontology constructed using classes derived from the vocabularies of MAEC and STIX languages. This section introduces key classes in the ontology, primarily derived from MAEC's top-level objects, along with selected object properties that define relationships between these classes, and relevant data properties. We formalize our ontology in OWL, one of the most widely used ontology languages for knowledge representation. [11]. It is a more expressive language for designing complex ontologies which support the extension of RDFS with richer semantics such as property restriction, equivalence and disjointness and enhances reasoning [12, 13].

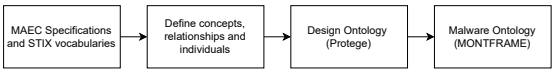

Figure 1: MAECO Process Workflow

Figure 1 illustrates our systematic approach to designing the Malware Ontology (MAECO). We began by reviewing and analyzing the MAEC and STIX [14] languages to understand existing standards in malware characterization. Based on this analysis, we defined relevant concepts, relationships, and individuals, along with their associated properties. Utilizing Protégé [15], we structured the ontology according to these defined classes and properties. The culmination of this process is the development of the standardized Malware Ontology Framework (MAECO). Table 1 present the metrics of the MAECO ontology in comparison with the PE Malware Ontology. [6]. The metrics which include Total Axioms of 2,929, Logical Axiom Count of 1,702, Declaration Axiom Count of 1,227 with Class Count of 195 and Object Property Count 107. Data Property Count is 194 with Individual Count of 393. SubClassOf Axioms of 510 with EquivalentClasses Axioms equall to 1. These figures indicate a moderately expressive ontology, with a balanced representation of schema-level constructs (classes, properties) and instance-level assertions (individuals). The high count of SubClassOf axioms suggests a well-structured class hierarchy over the existing PEFile Ontology [6]

4. Investigating the Suitability of GNNs on Ontology-Based Datasets

We constructed PyTorch Geometric (PyG) graph data from ontology-based knowledge graphs created by Daniel et al. [16], derived using the malware ontology of Švec et al. [17] from the EMBER dataset (1,000 binary-labeled samples). To assess the suitability of graph neural networks (GNNs), we conducted four experiments:

Table 1
Structural and Evaluation metrics for PEFile and MAECO Ontologies

Structural Metrics		
Metric	**PEFile Ontology [6]**	**MAECO**
Total Axioms	874	2,929
Logical Axiom Count	231	1,702
Declaration Axiom Count	210	1,227
Class Count	195	535
Object Property Count	6	107
Data Property Count	10	194
Individual Count	0	393
Annotation Property Count	3	0
SubClassOf Axioms	189	510
EquivalentClasses Axioms	0	1
Evaluation Metrics		
Attribute Richness	0.051	0.36
Inheritance Richness	0.97	0.95
Relationship Richness	0.032	0.21
Annotation Coverage	0.015	0.00
Class Richness	1.000	0.58
Average Population	0.00	0.73
Axiom Richness	2.96	3.03
Logical Axiom Density	0.264	0.58
Declaration Axiom Density	0.240	0.42

1. **GCN1** – Graph Convolutional Network
2. **GCN2** – Graph Convolutional Network with edge reversal
3. **RGCN1** – Relational Graph Convolutional Network
4. **RGCN2** – Relational Graph Convolutional Network with edge reversal

The goal was to evaluate the effect of bidirectional relations (edge reversal) when learning from numeric feature subsets. The results are summarized in Table 2.

Table 2
Performance of GCN and RGCN variants on ontology-based dataset.

Model	**Accuracy (%)**	**TPR (%)**
GCN1	67	55
GCN2	67	87
RGCN1	67	87
RGCN2	**98**	**98**

The results show that bidirectional relations significantly improve performance. In particular, RGCN with edge reversal (RGCN2) achieved 98% accuracy and TPR, compared to 67% in baseline models. This demonstrates that relational GNNs are highly suitable for ontology-based datasets, where relational structures are central.

In our envisaged neuro-symbolic pipeline, the ontology structures domain knowledge into relational graphs of malware samples. A relational graph convolutional network (RGCN) learns embeddings and probabilistic predictions from these graphs, which are then passed into DeepProbLog and combined with logical rules derived from the ontology. This integration allows the ontology to constrain the feature space, the RGCN to capture statistical relational patterns, and DeepProbLog to provide probabilistic reasoning with symbolic explanations.

5. Related Work

Ontology-based approaches for malware detection have been investigated in various forms, aiming to formalize malware behaviors and characteristics for better representation and reasoning. The earliest ontology [18]; a core ontology to model suspicious malware behaviour, lacks grounding in formal standards such as MAEC or STIX, reducing reusability and interoperability.

Chowdhury and Bhowmik [19] introduced a knowledge graph-based approach to model malware behavior, capturing relationships between malware indicators and behavior patterns. While their system provided graphical interpretations, it did not explicitly rely on standardized ontology vocabularies and lacked dynamic feature integration. More recently, Svec et al. [20] presented a PE malware ontology using the MAEC vocabulary to model features from static analysis. Although their model included explainability considerations, it was limited to Windows PE binaries and static features leaving out dynamic behaviors which are essential in uncovering evasive malware traits. Balogh and Galko [21] addressed integration of both static and dynamic analysis results into an ontological model using MAEC. Their work is foundational to this proposal, as it confirms the feasibility of unifying hybrid malware attributes into a single semantic representation. However, it stops short of applying explainable machine learning models to the resulting ontology-enhanced datasets. In a broader AI context, research on explainability has progressed rapidly. [22] outlines principles of Explainable AI (XAI), highlighting its role in transparency, fairness, and user trust. While numerous studies explore XAI in domains such as healthcare or finance, few have systematically applied XAI to malware detection in conjunction with ontology-based representations.

To the best of our knowledge, no existing work has combined dynamic malware analysis, ontology-based semantic modeling grounded in MAEC/STIX, and explainable learning using GNNs and DeepProbLog. This research thus fills a key gap by proposing a novel end-to-end pipeline that incorporates all these elements for interpretable malware detection.

Acknowledgments

Funded by the EU NextGenerationEU through the Recovery and Resilience Plan for Slovakia under the project No. 09I05-03-V02-00064.

Declaration on Generative AI

During the preparation of this work, the author(s) used X-GPT-4 in order to: Grammar and spelling check. After using these tool/service, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

[1] B. Molina-Coronado, A. Ruggia, U. Mori, A. Merlo, A. Mendiburu, J. Miguel-Alonso, Light up that droid! on the effectiveness of static analysis features against app obfuscation for android malware detection, Journal of Network and Computer Applications 235 (2025) 104094. doi:https://doi.org/10.1016/j.jnca.2024.104094.

[2] J. Geng, J. Wang, Z. Fang, Y. Zhou, D. Wu, W. Ge, A survey of strategy-driven evasion methods for pe malware: Transformation, concealment, and attack, Computers & Security 137 (2024) 103595. doi:https://doi.org/10.1016/j.cose.2023.103595.

[3] B. Xu, Y. Li, X. Yu, Malware detection based on static and dynamic features analysis, in: X. Chen, H. Yan, Q. Yan, X. Zhang (Eds.), Machine Learning for Cyber Security, 2020.

[4] Z. Syed, A. Padia, T. W. Finin, M. L. Mathews, A. Joshi, Uco: A unified cybersecurity ontology, in: AAAI Workshop: Artificial Intelligence for Cyber Security, 2016. URL: https://api.semanticscholar.org/CorpusID:6896947.

[5] S. U. Zamida, M. J. M. Chowdhury, N. R. Chakraborty, K. Biswas, S. K. Sami, Cybersecurity vulnerability management: A conceptual ontology and cyber intelligence alert system, Information & Management 57 (2020) 103334. doi:https://doi.org/10.1016/j.im.2020.103334.

[6] P. Svec, S. Balogh, M. Homola, J. Kluka, T. Bisták, Semantic data representation for explainable windows malware detection models, CoRR abs/2403.11669 (2024). URL: https://doi.org/10.48550/arXiv.2403.11669. doi:10.48550/ARXIV.2403.11669. arXiv:2403.11669.

[7] B. Khemani, S. Patil, K. Kotecha, S. Tanwar, A review of graph neural networks: concepts, architectures, techniques, challenges, datasets, applications, and future directions, Journal of Big Data 11 (2024). URL: http://dx.doi.org/10.1186/s40537-023-00876-4. doi:10.1186/s40537-023-00876-4.

[8] R. Manhaeve, S. Dumančić, A. Kimmig, T. Demeester, L. De Raedt, Neural probabilistic logic programming in deepproblog, Artificial Intelligence 298 (2021) 103504. URL: http://dx.doi.org/10.1016/j.artint.2021.103504. doi:10.1016/j.artint.2021.103504.

[9] H. Yuan, H. Yu, S. Gui, S. Ji, Explainability in graph neural networks: A taxonomic survey, IEEE Transactions on Pattern Analysis and Machine Intelligence (2022) 1–19. URL: http://dx.doi.org/10.1109/tpami.2022.3204236. doi:10.1109/tpami.2022.3204236.

[10] J. Kakkad, J. Jannu, K. Sharma, C. Aggarwal, S. Medya, A survey on explainability of graph neural networks (2023). URL: https://arxiv.org/abs/2306.01958. doi:10.48550/ARXIV.2306.01958.

[11] F. H. Abanda, J. H. M. Tah, R. Keivani, Trends in built environment semantic web applications: Where are we today?, Expert Syst. Appl. 40 (2013) 5563–5577. URL: https://doi.org/10.1016/j.eswa.2013.04.027. doi:10.1016/J.ESWA.2013.04.027.

[12] J. Z. Pan, OWL for the Novice: A Logical Perspective, Springer US, 2008, pp. 159–182. doi:10.1007/978-0-387-48438-9_9.

[13] Z. Zuo, M. Zhou, Web ontology language owl and its description logic foundation, in: Proceedings of the 8th International Scientific and Practical Conference of Students, Post-graduates and Young Scientists. Modern Technique and Technologies. MTT'2002 (Cat. No.02EX550), IEEE, 2003, pp. 157–160. doi:10.1109/PDCAT.2003.1236278.

[14] B. Jordan, R. Piazza, T. Darley, STIX™ Version 2.1, Committee Specification 02, OASIS Cyber Threat Intelligence (CTI) Technical Committee, 2021. URL: https://docs.oasis-open.org/cti/stix/v2.1/cs02/stix-v2.1-cs02.html, approved 25 January 2021.

[15] M. A. Musen, The protégé project: a look back and a look forward, AI Matters 1 (2015) 4–12. URL: https://doi.org/10.1145/2757001.2757003. doi:10.1145/2757001.2757003.

[16] D. Trizna, P. Anthony, M. Homola, Z. Adams, Š. Balogh, Learning explainable malware characterization using knowledge base embedding, in: 2024 IEEE 5th International Conference on Electro-Computing Technologies for Humanity (NIGERCON), IEEE, 2024, pp. 20–24. doi:10.1109/NIGERCON62786.2024.10927262.

[17] P. Švec, Š. Balogh, M. Homola, J. Kľuka, T. Bisták, Semantic data representation for explainable windows malware detection models, arXiv preprint arXiv:2403.11669 (2024).

[18] A. G. R. de Geus, M. Jino, A. P. Lopes, Ontology for malware behavior: A core model proposal, in: IEEE 23rd International WETICE Conference, 2014, pp. 453–458. doi:10.1109/WETICE.2014.72.

[19] I. R. Chowdhury, D. Bhowmik, Capturing malware behaviour with ontology-based knowledge graphs, in: IEEE Conference on Dependable and Secure Computing (DSC), 2022, pp. 1–7. doi:10.1109/DSC54232.2022.9888860.

[20] P. Svec, S. Balogh, M. Homola, J. Kluka, T. Bisták, Semantic data representation for explainable windows malware detection models, arXiv preprint arXiv:2403.11669 (2024). URL: https://arxiv.org/abs/2403.11669. doi:10.48550/arXiv.2403.11669.

[21] S. Balogh, T. Galko, Integration of results from static and dynamic code analysis into an ontological model, in: 12th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems (IDAACS), 2023, pp. 680–685. doi:10.1109/IDAACS58523.2023.10348799.

[22] IBM, Explainable ai (xai), 2024. URL: https://www.ibm.com/watson/explainable-ai, accessed: 2025-07-16.