AKMIS 2025

The 4th Workshop on Application of Knowledge Methods in Information Security

on 2nd - 4th October, 2025
in Smolenice, Slovakia

BOOK OF ABSTRACTS

organized by

Institute of Computer Science and Mathematics,
Faculty of Electrical Engineering and Information Technology,
Slovak University of Technology in Bratislava
(FEI STU)

Department of Applied Informatics,
Faculty of Mathematics, Physics and Informatics,
Comenius University in Bratislava
(FMPH UK)

Institute of Informatics,
Slovak Academy of Sciences
(II SAS)




EDITORS:

Štefan BALOGH

affiliated with:
Institute of Computer Science and Mathematics
Faculty of Electrical Engineering and Information Technology
Slovak University of Technology in Bratislava

Martin HOMOLA

affiliated with:
Department of Applied Mathematics
Faculty of Mathematics, Physics, and Informatics
Comenius University in Bratislava

Martin KENYERES

affiliated with:
Institute of Informatics
Slovak Academy of Sciences

This research was funded by the Scientific Grant Agency of the Slovak Republic grant number APVV-23-0292

Organizators:

Funder:

PREFACE:

The ever-increasing availability of data continues to drive progress across nearly all domains of human endeavor. In the field of information security, such data sources constitute valuable repositories of knowledge that can be leveraged to enhance detection, protection, and timely response mechanisms against increasingly frequent and sophisticated security threats.

The AKMIS Workshop series serves as a platform for scientists, researchers, and practitioners to engage in open discussion and exchange of ideas concerning recent research developments and emerging trends in the utilization of data and knowledge for strengthening information security. AKMIS is committed to inclusivity and collaboration, and this Book of Abstracts has been published under the auspices of the Dymax Project Consortium. This volume presents the proceedings of the 4th AKMIS Workshop, held in Smolenice, Slovakia, from October 2 to 4, 2025. This year, AKMIS received 23 submissions in the form of extended abstracts. Each submission underwent a rigorous peer-review process, receiving three independent reviews from members of the program committee or additional external reviewers. Furthermore, all submissions were evaluated by domain experts, and given their relevance to the workshop's scope, all 23 were accepted for inclusion in this volume.

AKMIS 2025 was organized under the general patronage of the Dymax Project, supported by the Slovak Research and Development Agency (SRDA) under contract No. APVV-23-0292. The workshop co-chairs would like to express their sincere gratitude to all colleagues, authors, and participants for their contributions to the organization of the event, their engaging presentations, and the productive discussions that ensued. We look forward to continuing this tradition with future AKMIS workshops. Finally, we extend our appreciation to all reviewers for their insightful and thorough evaluations, which have greatly contributed to the quality and rigor of the published abstracts.

Bratislava, October 2025
Štefan Balogh
Ivana Budinská
Martin Homola
Martin Kenyeres

This research was funded by the Scientific Grant Agency of the Slovak Republic grant number APVV-23-0292

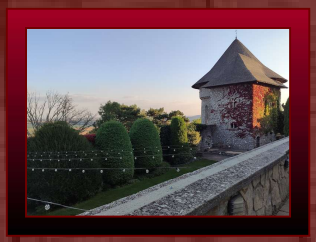
TOPICS:

The workshop is organized with the aim to bring together scientists. researchers and practitioners from any country and provide an open platform for discussion about recent research developments and future trends in creative and informal atmosphere. Application of various methods from knowledge extraction, management and artificial intelligence, such as:

- automated reasoning,
- information retrieval,
- data mining,
- machine learning,
- knowledge representation,
- ontologies reasoning, explanation,
- neural networks, argumentation, automated question answering,
- description logics learning algorithms,
and others with applications in

- system monitoring,
- information intelligence sharing,
- malware detection,
- malware features analysis,
- ontology based security,
- context reasoning,
- ontology models for security domain,
- incident detections, share schemes and responds strategies,
and others.

The fields of interests of the workshop include topics related to knowledge sharing in various domains. Therefore the topics are not limited to the information security domain.

This research was funded by the Scientific Grant Agency of the Slovak Republic grant number APVV-23-0292

Organizators:

STU FEI

Funder: SRDA

LIST OF REVIEWERS:

The editorial board would like to express its sincere gratitude to all reviewers for their professional work, valuable comments, and the time devoted to evaluating the submitted abstracts to AKMIS 2025. Their responsible approach has greatly contributed to the overall quality and scholarly level of this book of abstracts.

List in alphabetical order:

Zekeri ADAMS
Peter ANTHONY
Štefan BALOGH
Iveta BEČKOVÁ
Ivana BUDINSKÁ
Daniela CHUDÁ
Damas GRUSKA
Martin HOMOLA
Noufal ISSA
Matúš JÓKAY
Martin KENYERES
Jaroslav KOPČAN
Ján KĽUKA
Martin MOCKO
Ján MOJŽIŠ
Ítalo OLIVEIRA
Monday ONOJA
Roderik PLOSZEK
Štefan PÓCOŠ
Kefas RIMAMNUSKEB GALADIMA
Daniele Francesco SANTAMARIA
Elena ŠTEFANCOVÁ
Peter ŠVEC
Pavol ZAJAC

This research was funded by the Scientific Grant Agency of the Slovak Republic grant number APVV-23-0292

CONTENTS:

part 1

This research was funded by the Scientific Grant Agency of the Slovak Republic grant number APVV-23-0292

CONTENTS:

part 2

This research was funded by the Scientific Grant Agency of the Slovak Republic grant number APVV-23-0292

COMPLETE WORKSHOP PROGRAM:

	2 Oct 2025	3 Oct 2025	4 Oct 2025
08:00		Breakfast	Breakfast
09:15		AKMIS talks	
09:30			Networking + Coffee break
10:30			Excursion (castle)
10:45		Coffee break	
11:15		AKMIS talks	
11:30			
12:00			Lunch
12:45			
13:00		Lunch	
14:00			
15:00		Excursion (Včelovina)	
15:45	Opening + AKMIS talks		
16:30	Coffee Break		
17:00	AKMIS talks		
18:00		Dinner	
18:15			
19:00	Banquet		

This research was funded by the Scientific Grant Agency of the Slovak Republic grant number APVV-23-0292

Organizators:

Funder:

WORKSHOP PROGRAM:

part 1

SESSION #1

Time and Date :	15:45 - 16:30, 2nd October 2025
Session chair :	doc. RNDr. Martin HOMOLA, PhD.

Room: Lovecký salón

Presentation I

OPENING

PRESENTER: Martin Homola — Time: 15:45 - 16:00

Presentation II

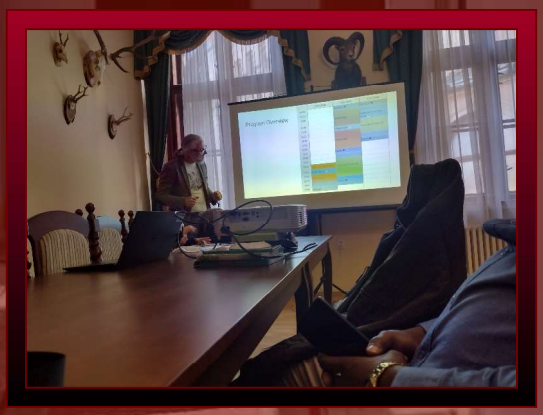
Improving Malware Clustering Through Self-Supervised Learning

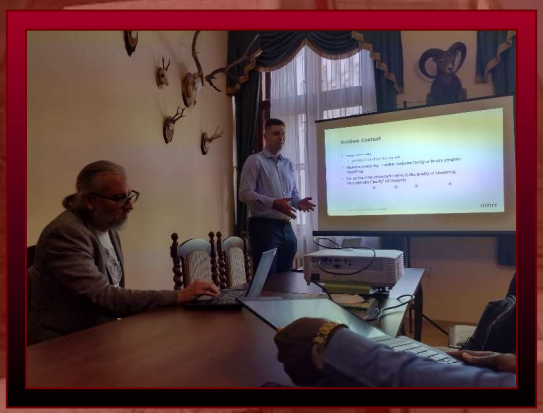
PRESENTER: Martin Mocko — Time: 16:00 - 16:15 (long talk)

Presentation III

Towards Explainable Malware Clustering

PRESENTER: Martin Mocko — Time: 16:15 - 16:30 (long talk)





This research was funded by the Scientific Grant Agency of the Slovak Republic grant number APVV-23-0292

Organizators: STU FEI

Funder: SRDA

WORKSHOP PROGRAM:

part 2

SESSION #2

Time and Date :	17:00 - 18:15, 2nd October 2025
Session chair :	doc. RNDr. Damas GRUSKA, PhD.

Room: Lovecký salón

Presentation I

Integrating Ontology and Graph Neural Network for Explainable Malware Detection

PRESENTER: Monday Onoja — Time: 17:00 - 17:15 (long talk)

Presentation II

Counterfactual Explanations to Detect Adversarial Vulnerability of Malware Classifiers

PRESENTER: Iveta Bečková — Time: 17:15 - 17:30 (long talk)

Presentation III

Qualitative Evaluation of Explainable Malware Detection

PRESENTER: Jaroslav Kopčan — Time: 17:30 - 17:45 (long talk)

Presentation IV

Preliminary User Study on Concept Expressions for Characterizing Malware

PRESENTER: Martin Homola — Time: 17:45 - 17:55 (short talk)

Presentation V

On the Prospects of EL and ELU Concept Learners for Explainable Malware Detection

PRESENTER: Martin Demovič — Time: 17:55 - 18:05 (short talk)

Presentation VI

On the Machine Learning Utilization for Concept Learning in Malware Domain

PRESENTER: Ján Mojžiš — Time: 18:05 - 18:15 (short talk)

This research was funded by the Scientific Grant Agency of the Slovak Republic grant number APVV-23-0292

Organizators: STU FEI

Funder: SRDA

WORKSHOP PROGRAM:

part 3

SESSION #3

Time and Date :	09:15 - 10:45, 3rd October 2025
Session chair :	Ing. Ivana BUDINSKÁ, PhD.

Room: Lovecký salón

Presentation I

LLM and interpretability in security domain

PRESENTER: Štefan Balogh — Time: 09:15 - 09:30 (long talk)

Presentation II

LEMNA vs. SHAP and LIME: Choosing the Right XAI Method for Malware Analysis

PRESENTER: Rimamnuskeb Galadima Kefas — Time: 09:30 - 09:45 (long talk)

Presentation III

X-MalNet: A Novel Multi-Level eXplainability Framework for Malware Detection Using Matrix Product...

PRESENTER: Peter Anthony — Time: 09:45 - 10:00 (long talk)

Presentation IV

MAECO: Malware Ontology Framework Towards Enhancing Explainable Malware Detection

PRESENTER: Zekeri Adams — Time: 10:00 - 10:15 (long talk)

Presentation V

Transforming Malware Ontology via Self-attention

PRESENTER: Štefan Pócoš — Time: 10:15 - 10:30 (long talk)

Presentation VI

Converting Malware Reports into Ontology: Progress Report

PRESENTER: Roderik Ploszek — Time: 10:30 - 10:40 (short talk)

This research was funded by the Scientific Grant Agency of the Slovak Republic grant number APVV-23-0292

Organizators: STU FEI

Funder: SRDA

WORKSHOP PROGRAM:

part 4

SESSION #4

Time and Date :	11:15 - 12:45, 3rd October 2025
Session chair :	Ing. Jaroslav KOPČAN, PhD.

Room: Lovecký salón

Presentation I

A Hybrid GAM-based Model for Predicting Vulnerability Exploitation

PRESENTER: Noufal Issa | Time: 09:15 - 09:30 (long talk)

Presentation II

Fair and Explainable Recommendations

PRESENTER: Elena Štefancová | Time: 09:30 - 09:45 (long talk)

Presentation III

Internet of Things - Cybersecurity Issues

PRESENTER: Ivana Budinská | Time: 09:45 - 10:00 (long talk)

Presentation IV

Anomaly Detection Framework For Fraud In E-Commerce Using Enhanced Isolation Forest: A GDPR...

PRESENTER: Aisha Suleiman | Time: 10:00 - 10:15 (long talk)

Presentation V

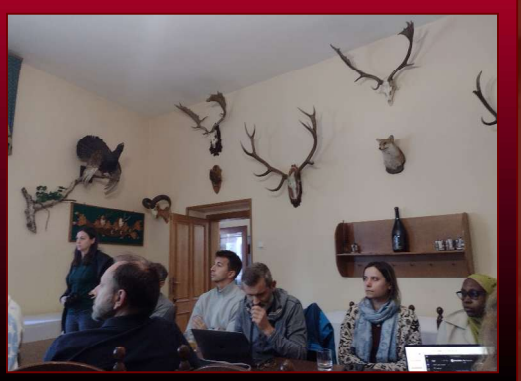
Attack Trees, Intruders and Defenders

PRESENTER: Damas Gruska | Time: 10:15 - 10:30 (short talk)

Presentation VI

Towards Semantic Security Policy Representation

PRESENTER: Pavol Zajac | Time: 10:30 - 10:40 (short talk)



This research was funded by the Scientific Grant Agency of the Slovak Republic grant number APVV-23-0292

Organizators: STU FEI

Funder: SRDA

Towards semantic security policy representation

Pavol Zajac *

Slovak University of Technology in Bratislava, Slovakia

Abstract

One of the aims of the project *Dynamic Malware Analysis by eXplainable AI* (DyMAX) is to improve malware analysis by with rich semantic representation. The research is primarily focused on extensive malware datasets, such as semantically-treated EMBER data [3]. The core classes of the proposed EMBER ontology focus on the malware PE file, and corresponding sections, file features and actions performed by the investigated file.

In this contribution we want to discuss some limitations of the current research, with additional proposal how to improve applications of the research in the security domain.

From the security perspective, the malware file itself is just one step in the attack, understood as a process by which some attacker wants to perform actions disallowed by the security policy. The same PE file can be understood as legitimate, and malware depending on the context. One example is the software that enables remote administration of the work station. It can be installed and maintained by the organization with respect to the organization's security policy. On the other hand, installation of remote administration software by the user tricked by some attacker is a very common technique in malware delivery [2].

We point out that semantic representation should not focus on determining whether some sample is malware or legitimate software. Instead, it should focus on providing security relevant information, such as "This software can enable remote access to the device on which it is run", or "The software contains encrypted sections".

The counterpart to this research is then a question of semantic security policy representation. This should be compatible with analysis results, so we can express rules such as "Software that has remote access capabilities is legitimate only if it is signed by organization X", or "This workstation can only execute files that do not contain encrypted sections".

The combination of semantic security description of the application, with the semantic security policy can then become a basis for a more efficient protection against attacks that involve user installation of the software, similar to technique presented in [1] for Android applications.

References

[1] Muska, P., and Varga, J. Presenting risks introduced by android application permissions in a user-friendly way. *Tatra Mt. Math. Publ 60* (2014), 85–100.

[2] Proofpoint, Inc. The human factor 2025 vol. 2: Url phishing. Threat report, Proofpoint, Inc., Aug. 2025. Volume 2 of the "Human Factor" report series, focusing on phishing and malicious URL-based threats; accessible via Proofpoint's website.

[3] Svec, P., Balogh, S., Homola, M., and Kluka, J. Knowledge-based dataset for training PE malware detection models. *CoRR abs/2301.00153* (2023).

*This work was supported by the Slovak Research and Development Agency under the Contract no. APVV-23-0292.

Attack Trees, Intruders and Defenders *

Damas Gruska
Department of Applied Informatics, Comenius University in Bratislava,
Mlynská dolina, 842 48 Bratislava, Slovakia
gruska@fmph.uniba.sk

Abstract

In this paper, we investigate a dynamic security model based on Attack Trees (ATs) enriched with time and cost parameters, where both an attacker and a supervisor (defender) operate under partial observability and constrained budgets. The attacker aims to compromise the system by reaching the root node of the AT, while the supervisor seeks to prevent this by strategically allocating defensive resources. We explore several modeling options to represent defense costs and observability, including dual-cost parameters and time-windowed visibility. Our central questions address the conditions under which an AT can be defended, whether the root node is always vulnerable, and the minimal budget required for guaranteed defense. We propose a formal framework for analyzing these interactions and provide insights into optimal defense strategies under uncertainty.

Attack Trees Attack-Defense Security Threat Modeling Cyber-Physical Systems

A traditional attack tree ([2, 5]) is a rooted tree whose leaves are atomic attack steps (e.g., "obtain VPN credentials", "exploit firmware vulnerability"), and whose internal nodes are gates (OR, AND, SAND) that dictate how subgoals combine to achieve a higher-level goal (e.g., "gain root on the historian"). In some extended models, each node carries cost and time attributes , while others introduce fixed time intervals within which each step must be completed [1, 3, 4, 8, 9, 10]. The analysis then reduces to a shortest-path search: identifying the cheapest set of leaf nodes whose compromise ultimately triggers the root This abstraction is elegant, but it hides three crucial realities:

- **Observability gaps.** The analyst sees the tree on paper, but the attacker in the field does not. A phishing campaign may or may not succeed; the attacker will not know until much later, if ever. Conversely, a security-monitoring dashboard may light up with alerts that reveal only a subset of compromised nodes.

*Work funded by the EU NextGenerationEU through the Recovery and Resilience Plan for Slovakia under the project No. 09I03-03-V04-00095

- **Resource scarcity.** Unlimited budgets do not exist. An attacker can spend at most budget B_A ; a defender can spend at most budget B_D. Once either budget is exhausted, the game ends.
- **Dynamic interaction.** traditional ATs are *static*: the defender's countermeasures (if any) are pre-computed and baked into the tree. In reality the defender *reacts* patching a node, shortening an exploit window, or reconfiguring a firewall after observing some, but not all, of the attacker's moves.

SATs address these realities by layering two additional ingredients onto the traditional model:

- **Partial observation sets.** The attacker is allowed to observe only a subset $\mathcal{N}_A$ of nodes; the supervisor observes a possibly different subset $\mathcal{N}_D$. Neither party ever sees the full state.
- **Defence budget and actions.** Each non-leaf node n carries a defence cost $c_D(n)$. By paying this cost, the supervisor can postpone the interval during which n can be compromised, effectively "buying time" or "raising the bar" for the attacker. Importantly, the supervisor chooses *which* nodes to defend *after* seeing a partial snapshot of the attack, not *a priori*.

Research Questions:

With these ingredients in place, we revisit the fundamental questions posed in threat analysis (see also [6, 7]):

- **Defendability.** Given SATs, a defender budget B_D, and an attacker budget B_A, is there a strategy for the supervisor based solely on observations $\mathcal{N}_D$ that guarantees the root will *never* be compromised, no matter how the attacker spends B_A?
- **Attackability.** Dually, is there a strategy for the attacker based solely on observations $\mathcal{N}_A$ that guarantees the root will *eventually* be compromised, no matter how the supervisor spends B_D?
- **Minimum defence budget.** What is the smallest B_D such that the tree becomes *always defendable*? This question is of acute practical interest to Chief Information Security Officers (CISOs) who must justify security investments.
- **Minimum attack budget.** Symmetrically, what is the smallest B_A that renders the tree *always attackable*?

References

[1] Aliyu Tanko Ali. Simplified timed attack trees. In *International Conference on Research Challenges in Information Science*, pages 653–660. Springer, 2021.

[2] Florian Arnold, Dennis Guck, Rajesh Kumar, and Mariële Stoelinga. Sequential and parallel attack tree modelling. In *Computer Safety, Reliability, and Security: SAFECOMP 2015 Workshops, ASSURE, DECSoS. ISSE, ReSA4CI, and SASSUR, Delft, The Netherlands, September 22, 2015, Proceedings 34*, pages 291–299. Springer, 2015.

[3] Aliyu Tanko Ali and Damas P Gruska. Attack trees with time constraints. In *CS&P*, pages 93–105, 2021.

[4] Aliyu Tanko Ali and Damas P Gruska. Attack protection tree. In *CS&P*, 2019.

[5] Bruce Schneier. Attack trees. *Dr. Dobb's journal*, 24(12):21–29, 1999.

[6] Stefano Bistarelli, Marco Dall'Aglio, and Pamela Peretti. Strategic games on defense trees. In *Formal Aspects in Security and Trust: Fourth International Workshop, FAST 2006, Hamilton, Ontario, Canada, August 26-27, 2006, Revised Selected Papers 4*, pages 1–15. Springer, 2007.

[7] Barbara Kordy, Sjouke Mauw, Saša Radomirović, and Patrick Schweitzer. Attack–defense trees. *Journal of Logic and Computation*, 24(1):55–87, 2014.

[8] Aliyu Tanko Ali and Damas Gruska. States of attack under incomplete information. In *2022 IEEE 12th Annual Computing and Communication Workshop and Conference (CCWC)*, pages 0801–0807. IEEE, 2022.

[9] Ali Aliyu Tanko, Gruska Damas, Kharraz Karam, and Leucker Martin. Analysis of attack time and costs in attack trees via smt resolution. In *Proceedings of the 8th International Conference on Future Networks and Distributed Systems*, 2024.

[10] Damas Gruska, Aliyu Tanko Ali, and Martin Leucker. Using attack trees for security education and training: Simplifying threat analysis. In *IFIP World Conference on Information Security Education*, pages 64–79. Springer, 2025.

Explainable Malware Detection via Relational Graph Neural Networks with Bidirectional Relations

Monday Onoja[1], Zekeri Adams[1], Peter Anthony[1], and Martin Homola[1]

Comenius University in Bratislava, Faculty of Mathematics, Physics, and Informatics, Mlynská dolina, 842 48 Bratislava, Slovakia
{monday.onoja,zekeri.adams,peter.anthony,martin.homola}@fmph.uniba.sk

Abstract. Graph Neural Networks (GNNs) are increasingly applied to cybersecurity tasks such as malware detection, intrusion detection, and program analysis, as they can model structured program representations and capture relational dependencies beyond flat feature vectors. However, their black-box nature poses challenges in security-critical domains, where analysts and stakeholders require explanations for trust and forensic analysis. This has motivated growing interest in explainable GNNs (XGNNs), which aim to provide interpretable insights into model decisions. In this work, we investigate Relational Graph Convolutional Networks (R-GCNs) for ontology-based malware detection. We introduce a proof-of-concept framework that incorporates bidirectional relations through edge reversal to strengthen semantic representation. Experimental results on the numeric subset of the Ontology–Knowledge Graph EMBER dataset (1,000 binaries) show that bidirectional relations substantially improve performance: R-GCN with edge reversal (RGCN2) achieved 98% accuracy and true positive rate (TPR), compared to 67% in baseline models, and delivered 87% fidelity with the Captum explainer. These findings demonstrate the effectiveness of relational GNNs in leveraging semantic structures for robust and interpretable malware detection.

Keywords: Explainability · Malware detection· Ontology · GNN.

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

A Hybrid GAM-based Model for Predicting Vulnerability Exploitation

Noufal Issa[1][0000−0003−0606−5182], Damas Gruska[2][0000−0002−8517−4688], and Loubna Ali[3][0000−0002−6706−1890]

Department of Applied Informatics, Faculty of Math, Physics, and Informatics, Comenius University, Bratislava, Slovakia[1] Faculty of Computer Science and Informatics, Berlin School of Business and Innovation, Berlin, Germany[2]https://fmph.uniba.sk/en/departments/department-of-applied-informatics/

Abstract. Vulnerability management requires prioritizing which vulnerabilities to patch, since only a small fraction are ever exploited, and writing, testing, and installing patches can involve considerable resources, requiring companies to prioritize based on some notion of risk. Traditional severity scores, such as the Common Vulnerability Scoring System are often poor predictors of exploitation risk. Data-driven scores, such as the Exploit Prediction Scoring System, provide probabilities of exploitation, but still leave room for improvements. We propose a lightweight hybrid model using a Generalized Additive Model (GAM) that combines numeric features (CVSS base score, EPSS probability, age, reference count) with semantic text features (derived from the vulnerability description via Term Frequency–Inverse Document Frequency and Singular Value Decomposition). The GAM framework yields an interpretable, additive risk score without black-box explanations. On a 2023 training set (with labels from CISA's KEV and public exploits), our model achieves significantly better precision-recall tradeoff than CVSS or EPSS alone. Tested on 2024 disclosures, our presented model consistently outperforms the baselines at nearly all recall levels.

Keywords: Vulnerability, Prioritization, CVSS, EPSS, Hybrid GAM, Interpretability

1 Literature Review

Every year, thousands of new software vulnerabilities ranging from memory-corruption bugs (e.g. buffer overflows and use-after-free), injection flaws (e.g. SQL or command injection), authentication and authorization errors, to misconfigurations in operating systems, network services, and applications—are disclosed in public databases such as the National Vulnerability Database (NVD). However, only a small fraction of these vulnerabilities are ever exploited. Nevertheless, the sheer volume of disclosures continues to outpace the capacity of organizations to develop, test, and deploy patches. For example, Bilge and Dumitras [4] observed that after disclosure, "the volume of attacks exploiting [a

vulnerability] increases by 5 orders of magnitude", yet only 15% of disclosed flaws are ultimately exploited. This disparity underscores the need to predict which vulnerabilities are likely to be exploited so defenders can prioritize patching. Frei et al.[6] also find that over 70% of vulnerabilities that do get exploited had available exploits at disclosure time, suggesting that known attributes (impact metrics, references, exploit code) can inform risk estimates. Currently, most organizations rely on the Common Vulnerability Scoring System (CVSS) to rank CVEs by severity (with scores 0–10). However, CVSS is a severity-oriented, ordinal scale and was not designed to measure exploitability or risk [1]. In practice, it often fails to correlate with actual exploitation. For instance, Howland et al.[3] show that CVSS has "no correlation to exploited vulnerabilities in the wild" and "is unable to provide a meaningful metric for describing a vulnerability's severity, let alone risk"[1]. To address this gap, FIRST's EPSS initiative produced a data-driven score representing each vulnerability's likelihood of being exploited, achieving a reported ROC AUC of 0.838 on held-out data[2]. EPSS has become widely used for prioritization. However, the trade-offs between recall and precision indicate that there is still potential for improvement. In parallel, the research community has proposed machine-learning models to predict exploitability using various data sources (patch dates, social media, exploit databases). Most of these are complex or proprietary, and often have inflated performance when tested on static splits. In contrast, we seek a transparent, efficient model trained only on public data available at disclosure. Specifically, we combine CVSS base metrics, EPSS scores, and features extracted from the free-form vulnerability description. The description text is transformed via TF–IDF and reduced with singular value decomposition to capture its semantic content. Importantly, we do *not* rely on manually-derived flags (e.g. `AV:N`/remote, `AC:L`/low) to remain data-driven and to avoid depending on expert opinions, which some researchers use. Our model is a Generalized Additive Model (GAM) under a logistic link. GAMs (sums of feature-specific functions) are well-known interpretable models that balance accuracy with human-understandability[5]. They are particularly useful for financial and health service organizations that require interpretable models without using any additional tools [7].

Acknowledgments. This study was funded by the Slovak Research and Development Agency (SRDA) under grant APVV-23-0292 (DyMAX). The authors gratefully acknowledge this support.

References

1. B. L. Bullough, A. K. Yanchenko, C. L. Smith, and J. R. Zipkin. Predicting Exploitation of Disclosed Software Vulnerabilities Using Open-source Data. *IWSPA*, 2017.
2. J. Jacobs, S. Romanosky, I. Adjerid, and W. Baker. Improving vulnerability remediation through better exploit prediction. *Journal of Cybersecurity*, 6(1):1–15, 2020.

3. H. Howland. CVSS: Ubiquitous and Broken. *Digital Threats: Research and Practice*, 4(1), 2021.
4. L. Bilge and T. Dumitras. Before we knew it: an empirical study of zero-day attacks in the real world. In *CCS*, 2012.
5. E. Caruana, Y. Lou, J. Gehrke, P. Koch, M. Sturm, and N. Elhadad. Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission. In *Proc. 21st ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1721–1730, 2015.
6. S. Frei, D. Schatzmann, B. Plattner, and B. Trammell. Modeling the security ecosystem—The dynamics of (in)security. In T. Moore, D. J. Pym, and C. Ioannidis (eds.), *Economics of Information Security and Privacy*, pages 79–106. Springer, 2010.
7. C.-H. Chang, S. Tan, B. Lengerich, A. Goldenberg, and R. Caruana. How interpretable and trustworthy are GAMs? In *Proc. KDD*, 2021.

MAECO: Malware Ontology For Hybrid Explainable Malware Detection

Zekeri Adams[1], Monday Onoja[1], Ján Kľuka[1], Martin Homola[1], Štefan Balogh[2], and Roderik Ploszek[2]

[1] Comenius University in Bratislava, Faculty of Mathematics, Physics, and Informatics, Mlynská dolina, 842 48 Bratislava, Slovakia
{monday.onoja,zekeri.adams,martin.homola,jan.kluka}@fmph.uniba.sk
[2] Faculty of Electrical Engineering and Information Technology, Slovak University of Technology, Ilkovičova 3, 841 04 Bratislava, Slovakia
{stefan.balogh,roderik.ploszek}@stuba.sk

Abstract. Malware, short for malicious software, is continuously posing an increasing number of threats in today's interconnected world. In order to sufficiently represent malware behaviour for effective malware detection and to derive interpretations for decisions of machine learning models, malware researchers and cybersecurity experts are now delving into the application of ontology-based techniques in the malware domain. Although leveraging ontologies also has the potential of enhancing explainability, most efforts in the literature are focused on static malware ontologies covering limited features. In this work, we propose a malware ontology framework, called MAECO, based on static and dynamic attributes (hybrid), which will capture more actions, artifacts, and threat patterns sufficient for effective malware detection. Additionally, we propose a vocabulary formalization that is based on established standard languages for malware attribute representation, specifically combining the MAEC and STIX standards.

Keywords: Dynamic analysis · Malware · Ontology.

1 Ontology Overview

The proposed MAECO ontology integrates concepts from the Malware Attribute Enumeration and Characterization (MAEC) 5.0 language [2, 3] and the Structured Threat Information eXpression (STIX) 2.1 language [1].

The MAEC 5.0 language serves as the core foundation of our ontology. It defines two major structural components:

MAEC Top-Level Objects representing the fundamental malware entities, such as Malware Instances, Malware Families, Behaviors, and Actions;

MAEC Types providing fine-grained categorization and semantic enrichment for the MAEC top-level objects. These include capabilities, static features, dynamic features, API calls, and so on.

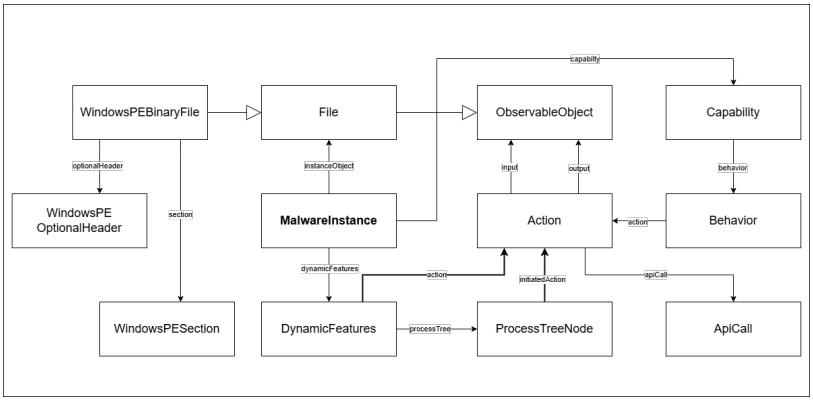


Fig. 1. UML Diagram of MAECO Core Classes and Relationships

To ensure interoperability with broader cyber threat intelligence standards, MAECO establishes explicit semantic links between MAEC objects and STIX Cyber-Observable Objects (SCOs). This linkage enables MAECO to incorporate contextual information from STIX such as files, network traffic, artifacts, certificates, IP addresses, and so on. These interconnections form a significant portion of the object properties and semantic relationships within the ontology. An overview of the core MAECO classes and their relationships is depicted in Fig. 1.

Another major strength of the MAECO ontology lies in its ability to describe malware techniques and tactics through the alignment of the MAEC 5.0 language with the MITRE ATT&CK framework [4]. By mapping ATT&CK's high-level tactics and low-level techniques into MAECO, our ontology supports formal reasoning about adversary behaviors, malware capabilities, and potential attack paths. Tactics and techniques are represented in the ontology by the ExternalReference class modeled after the respective MAEC 5.0 type, but also by more specific classes that enable expressing further properties and relationships.

For ontology validation and reasoning, we evaluated MAECO using the HermiT reasoner within the Protégé ontology editor. The reasoning results demonstrate that the ontology is logically consistent, semantically rich, and well-structured, making it suitable for representing complex malware knowledge and supporting automated inference.

2 Ongoing and Future Work

We are currently in the process of mapping the outputs of a hybrid malware analysis service[3] to the ontology with the aim of creating an ontological dataset of hybrid analysis data with both malware and benign samples. The dataset will then be used for experiments with training explainable AI models for malware detection, clustering, and other tasks.

[3] `https://hybrid-analysis.com`

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

References

1. Jordan, B., Piazza, R., Darley, T.: STIX™ version 2.1. Committee specification 02, OASIS Cyber Threat Intelligence (CTI) Technical Committee (Jan 2021), https://docs.oasis-open.org/cti/stix/v2.1/cs02/stix-v2.1-cs02.html, approved 25 January 2021, accessed: 2025-05-08
2. MAEC Project: MAEC™ 5.0 specification. Core concepts. Tech. rep., MITRE Corporation (2017), https://maecproject.github.io/releases/5.0/MAEC_Core_Specification.pdf, accessed: 2025-05-08
3. MAEC Project: MAEC™ 5.0 specification. Vocabularies. Tech. rep., MITRE Corporation (2017), https://maecproject.github.io/releases/5.0/MAEC_Vocabularies_Specification.pdf, accessed: 2025-05-08
4. Strom, B.E., Applebaum, A., Miller, D.P., Nickels, K.C., Pennington, A.G., Thomas, C.B.: MITRE ATT&CK®: Design and philosophy. Tech. rep., MITRE Corporation (2020), https://attack.mitre.org/docs/ATTACK_Design_and_Philosophy_March_2020.pdf, accessed: 2025-05-08

Anomaly Detection Framework For Credit Card Fraud In E-Commerce Using Enhanced Isolation Forest: A GDPR And DPIA-Compliant Approach

Hafsat Ashafa, Aisha Umar Suleiman

Faculty of Computing, Northwest University, Kano

ashafahafsat@gmail.com,ausuleiman@yumsuk.edu.ng

Abstract. Financial fraud and frequent data breaches are caused by cybercriminals taking advantage of flaws in systems that handle payments, store data, and authenticate users.

This study develops and evaluates a cybersecurity framework designed to ensure General Data Protection regulation (GDPR) compliance for online shopping platforms. By integrating the Data Protection Impact Assessment (DPIA) approaches, the framework identifies and mitigates data protection risks through features like privacy by design, multi-factor authentication, data encryption, role-based access controls and continuous security monitoring.

The research also aims to show how an optimised Isolation Forest algorithm may be modified for cybersecurity use cases while following data privacy standards. The integration of technical precision with regulatory adherence is a significant advancement in this domain.

Keywords. Data protection, compliance, cybersecurity framework, e-commerce

1 Introduction:

Significant financial and security threats have arisen as a result of the quick expansion of e-commerce which has led to a surge in credit card transactions and fraud. A probabilistic anomaly detection framework utilising Isolation Forest and CBLOF was presented by [1]. However, it had problems with scalability, failed to detect minority fraud situations, and lacked privacy-preserving features. By improving the Isolation Forest model, this study overcomes these drawbacks and incorporates GDPR and DPIA compliance to guarantee open, responsible and privacy-preserving fraud detection in e-commerce systems.

2 Related Works:

ML applications for fraud detection reviewed by [3]. [4] looked at concept drift and class imbalance in particular as challenges in identifying credit card fraud. [5] used Multiple Classifier Systems (MCS) in conjunction with sequential decision techniques

on two datasets, but despite ignoring imbalance handling [6] found that Logistic Regression (LR) was superior to Naïve Bayes and KNN when they analysed skewed data. To improve the efficacy of fraud detection [7] created a hybrid model by fusing supervised and unsupervised methodologies. They used annotated real-world datasets to evaluate their model. This work's drawback is that the imbalance problem was not addressed.

3 Approach

The design's essential elements include: Dataset Utilization: The Credit Card Fraud Detection dataset serves as the foundation for model development. Model Enhancement: Incorporates Pearson correlation-based feature selection, hybrid normalization, and RandomizedSearchCV for hyperparameter optimization. Comparative Assesment: Standard and hybrid anomaly detection models are used to compare the Enhanced Isolation Forest. Compliance assessment makes ensuring that DPIA risk control procedures.

4 Results

Fig. 1.

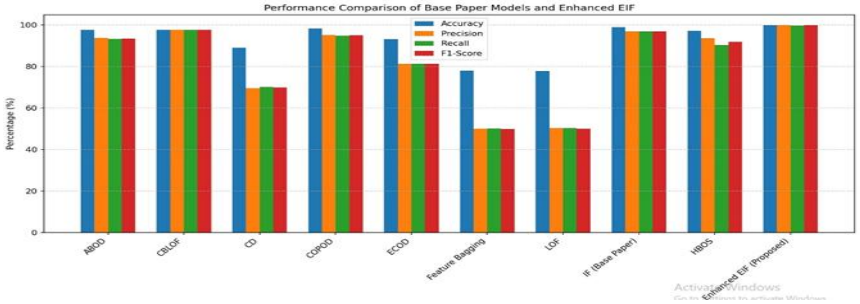
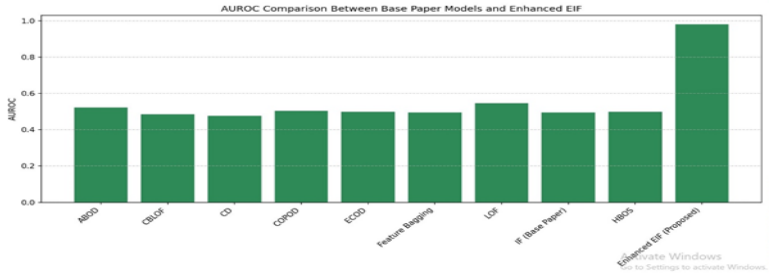


Fig. 2.



5 Conclusion

According to the experimental findings, the suggested Enhanced EIF framework offers significant advancements in both the technological and regulatory domains: Technical Performance: Even in extremely unbalanced datasets, the model's strong ROC-AUC, precision, and recall show that it can reliably detect infrequent fraud cases. The datasets are the focus of the enhancement. It improves upon the poor performance of traditional models that just use density estimation or statistical distance measures. Computational Efficiency: Although execution time and memory use increased slightly, inference time per transaction dropped, demonstrating the Enhanced EIF's applicability for real-time fraud detection in extensive e-commerce settings.

Acknowledgments

This study was funded by the Slovak Research and Development Agency (SRDA) under grant APVV-23-0292 (DyMAX).

References

[1] Chugh, B., Malik, N., Gupta, D., & Alkahtani, B. S. (2025). A probabilistic approach driven credit card anomaly detection with CBLOF and isolation forest models. *Alexandria Engineering Journal*, *114*, 231–242. https://doi.org/10.1016/j.aej.2024.11.054.

[2] Alfaiz, N. S., & Fati, S. M. (2022). Enhanced Credit Card Fraud Detection Model Using Machine Learning. *Electronics*, *11*(4), 662. https://doi.org/10.3390/electronics11040662.

[3] Modi, K., & Dayma, R. (2017). Review on fraud detection methods in credit card transactions. *2017 International Conference on Intelligent Computing and Control (I2C2)*, 1–5. https://doi.org/10.1109/I2C2.2017.8321781.

[4] Lucas, Y., & Jurgovsky, J. (2020). *Credit card fraud detection using machine learning: A survey* (Version 1). arXiv. https://doi.org/10.48550/ARXIV.2010.06479.

[5] Kalid, S. N., Ng, K.-H., Tong, G.-K., & Khor, K.-C. (2020). A Multiple Classifiers System for Anomaly Detection in Credit Card Data With Unbalanced and Overlapped Classes. *IEEE Access*, *8*, 28210–28221. https://doi.org/10.1109/ACCESS.2020.2972009.

[6] Itoo, F., Meenakshi, & Singh, S. (2021). Comparison and analysis of logistic regression, Naïve Bayes and KNN machine learning algorithms for credit card fraud detection. *International Journal of Information Technology*, *13*(4), 1503–1511. https://doi.org/10.1007/s41870-020-00430-y.

[7] Carcillo, F., Le Borgne, Y.-A., Caelen, O., Kessaci, Y., Oblé, F., & Bontempi, G. (2021). Combining unsupervised and supervised learning in credit card fraud detection. *Information Sciences*, *557*, 317–331. https://doi.org/10.1016/j.ins.2019.05.042.

LEMNA vs. SHAP and LIME: Choosing the Right XAI Method for Explainable Malware Detection

Kefas Rimamnuskeb Galadima, Roderik Ploszek, Štefan Balogh, and Pavol Zajac

Institute of Computer Science and Mathematics, Faculty of Electrical Engineering and Information Technology, Slovak University of Technology, Bratislava, Slovakia
{rimamnuskeb.kefas, roderik.ploszek, stefan.balogh, pavol.zajac}@stuba.sk

High accuracy has long been the primary objective of data-driven artificial intelligence (AI) research, with deep learning (DL) models being the preferred approach owing to their superior predictive capabilities. However, when applied in cybersecurity operations, their opaque decision-making raises concerns among analysts, as the high precision of DL models may result from biases embedded in training data rather than the genuine understanding of threats. In this work, we examine the potential of LEMNA, SHAP, and LIME in providing reliable and actionable justifications for a DL-based malware classifier trained using the EMBER dataset. Key metrics (fidelity, robustness, and inter-explainer agreement) were used to assess the faithfulness of the explanation. The MLP classifier had a false positive rate (FPR) of 0.0675 with an accuracy of 92%. On average across evaluation samples, LEMNA achieved the best fidelity (RMSE of 0.1258), whereas LIME exhibited the lowest fidelity (RMSE of 0.350), while demonstrating unusually high robustness, with a cosine similarity of 0.997. SHAP provided a balanced trade-off between robustness and fidelity. The Jaccard similarity score shows that both LIME and SHAP produced more consistent feature attributions, while LEMNA mostly identified distinct feature sets. The low Jaccard similarities between the explainers underscore the importance of employing multiple explanation techniques to obtain comprehensive and reliable insights in high-stakes malware detection. This work addressed the critical question: ***'When and why should LEMNA be chosen over SHAP or LIME for Explainable Malware Detection?'***. The findings offer researchers and practitioners practical guidance on the selection of a suitable explainable AI method for malware classifiers.

Keywords: XAI· Malware Detection· EMBER Dataset· Deep Learning· SHAP· LIME· LEMNA

Acknowledgements: This research was funded by the EU NextGenerationEU through the Recovery and Resilience Plan for Slovakia under project No. 09I05-03-V02-00064 and by the Slovak Research and Development Agency under the contract No. APVV-23-0292.

Improving Malware Clustering Through Self-Supervised Learning

Martin Mocko[1][0000−0001−8982−0141] and Daniela Chudá[3][0000−0002−3873−9308]

[1] Faculty of Information Technology, Brno University of Technology, Brno, Czechia
[2] Kempelen Institute of Intelligent Technologies, Bratislava, Slovakia
[3] Faculty of Electrical Engineering and Information Technology, Slovak University of Technology, Bratislava, Slovakia
martin.mocko@kinit.sk daniela.chuda@stuba.sk

Abstract. Clustering malware is a key task that supports domain experts in threat analysis, facilitates the discovery of novel attacks, and enables compact dataset representations. However, most existing approaches to malware clustering have not yet leveraged advanced deep learning methods, which hold promise for improving clustering quality. In this work, we review the current state-of-the-art in malware clustering and highlight potential directions for enhancement. We further propose an extension to BYOL and SimSiam models, BYOL-Tabular and SimSiam-Tabular, which is our adaptation of the BYOL and SimSiam models for tabular data. The models achieve competitive results on the malware (binary program) clustering task when training is done in a guided way. Our results demonstrate that self-supervised learning (SSL) methods can yield measurable improvements in malware clustering.

Keywords: malware clustering · self-supervised learning · BYOL · SimSiam.

1 Self-Supervised Learning for Malware Clustering

Recent research in the field of malware clustering has defined the state-of-the-art the performance of the malware clustering task on two public benchmark malware datasets [9]. The comparison included four clustering algorithms and three different standard feature representations (dimensionality reduction techniques) on malware benchmark datasets of Bodmas [10] and Ember [1]. Although the results are promising (i.e. Homogeneity in ranges of 70%-90%), we are still far away from perfect clustering results. For a malware clustering to be useful in practice, Homogeneity upwards of 90% is expected [9]. For clustering samples in the malware domain, it is also important what data [2] we use for the task of clustering, classification, or malware detection.

One promising way of improving malware clustering results is through self-supervised learning (SSL). SSL algorithms aim to learn discriminative features from vast quantities of unlabeled instances without relying on human annotations [7]. Often, a pretext task is defined based on unlabeled data, creating pseudo-labels for samples. Afterwards, the models (as far as this paper is

concerned) are trained on positive sample pairs which originated based on the pseudo-labels.

Achieving state-of-the-art results on the ImageNet dataset, self-supervised learning models, such as BYOL [6], SimSiam [4], and SwAV [3] have all proven to be competitive and interesting models for learning suitable image representations for downstream classification tasks. Therefore, these models could offer potential for improvements in other domains and tasks as well - like the task of malware clustering.

However, a huge obstacle arises when switching to the very different malware domain. We specifically aim for a model able to learn on tabular data. For tabular data, input augmentations such as rotations, cropping, or grayscale (which are often used in the computer vision domain) are not meaningful. We need to be able to create positive pairs for all (malware/benign) samples even though no universal tabular data transformations for SSL models exist.

To the best of our knowledge, the malware domain lacks a comprehensive set of transformations applicable to either raw binaries or their derived features. The prevailing approaches are simple, adversarial-style modifications developed primarily to attack machine-learning classifiers [5]; these do not adequately represent the sophisticated alterations employed by malware authors to evade detection. Domain-agnostic transformations (e.g., targeted value replacement or additive noise) constitute an alternative, but they likewise fail to reproduce the realism of in-the-wild malware evolution and neglect inter-feature dependencies that often play a critical role.

In the end, for our experiments, we decided to take a more powerful approach where we utilize label information to create positive pairs (for both benign and malware samples). We call it the Guided BYOL-Tabular / SimSiam-Tabular model. For each malware sample m_1 in the data, we check its family and sample a pair malware sample, m_2, which must belong to the same malware family. For benign samples, we do not have any family information (as benign software families are not recognized in the literature). Therefore, for any benign sample b_1 we sample a pair sample b_2 from the whole subset of all benign samples in the respective dataset.

Our preliminary experiments show competitive performance of the Guided BYOL-Tabular and SimSiam-Tabular models. We include an AutoEncoder baseline from [9]. Both SSL models were able to beat the baseline in terms of Homogeneity on class label. For Homogeneity on family label, the SSL models were yet unsuccessful in beating the baseline. However, this could be due to some specifics in the experiment setup regarding the malware families presented as pairs to the model.

2 Future work

In the future we would like to experiment with simple transformations like replacement by value and modification by noise (on the feature level). In this way, we could analyze the landscape of simple transformations and compare whether

Table 1. Results of experiments utilizing self-supervised learning models

Model	Dataset	Homogeneity on class label		Homogeneity on family label	
		Train set	Test set	Train set	Test set
AE Baseline	Bodmas	89.73%	91.39%	85.71%	88.48%
SimSiam-Tabular	Bodmas	97.23%	97.36%	80.22%	82.98%
BYOL-Tabular	Bodmas	97.98%	97.88%	74.78%	76.74%

these transformations could be effective in training SSL models in the malware domain. Furthermore, during the last few years, Tabular Representation Learning (TRL), the process of transforming tabular data into feature embeddings which aims to improve downstream tasks [8], has gained a lot of traction. Methods like these could prove to be useful in learning better tabular representations and could be another way towards achieving better clustering quality. Additionally, to have fuller variety of models covered, we also want to modify BarlowTwins for tabular data and include it in our experiments.

Acknowledgments. Funded by the EU NextGenerationEU through the Recovery and Resilience Plan for Slovakia under the project No. 09I05-03-V02-00064 and by the Slovak Research and Development Agency under the contract No. APVV-23-0292.

References

1. Anderson, H.S., Roth, P.: Ember: an open dataset for training static pe malware machine learning models. arXiv preprint arXiv:1804.04637 (2018)
2. Balogh, , Mojžiš, J.: New direction for malware detection using system features **1**, 176–183 (2019). https://doi.org/10.1109/IDAACS.2019.8924358
3. Caron, M., Misra, I., Mairal, J., Goyal, P., Bojanowski, P., Joulin, A.: Unsupervised learning of visual features by contrasting cluster assignments. Advances in neural information processing systems **33**, 9912–9924 (2020)
4. Chen, X., He, K.: Exploring simple siamese representation learning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 15750–15758 (2021)
5. Demetrio, L., Coull, S.E., Biggio, B., Lagorio, G., Armando, A., Roli, F.: Adversarial exemples: A survey and experimental evaluation of practical attacks on machine learning for windows malware detection. ACM Transactions on Privacy and Security (TOPS) **24**(4), 1–31 (2021)
6. Grill, J.B., Strub, F., Altché, F., Tallec, C., Richemond, P., Buchatskaya, E., Doersch, C., Avila Pires, B., Guo, Z., Gheshlaghi Azar, M., et al.: Bootstrap your own latent-a new approach to self-supervised learning. Advances in neural information processing systems **33**, 21271–21284 (2020)
7. Gui, J., Chen, T., Zhang, J., Cao, Q., Sun, Z., Luo, H., Tao, D.: A survey on self-supervised learning: Algorithms, applications, and future trends. IEEE Transactions on Pattern Analysis and Machine Intelligence **46**(12), 9052–9071 (2024)
8. Jiang, J.P., Liu, S.Y., Cai, H.R., Zhou, Q., Ye, H.J.: Representation learning for tabular data: A comprehensive survey. arXiv preprint arXiv:2504.16109 (2025)

9. Mocko, M., Ševcech, J., Chudá, D.: Clustering malware at scale: A first full-benchmark study. In: International Conference on Availability, Reliability and Security. pp. 231–251. Springer (2025)
10. Yang, L., Ciptadi, A., Laziuk, I., Ahmadzadeh, A., Wang, G.: Bodmas: An open dataset for learning based temporal analysis of pe malware. In: 2021 IEEE Security and Privacy Workshops (SPW). pp. 78–84. IEEE (2021)

Towards Explainable Malware Clustering

Martin Mocko[1,2][0000−0001−8982−0141], Jaroslav Kopčan[2][0000−0002−7467−6615], and Daniela Chudá[3,2][0000−0002−3873−9308]

[1] Faculty of Information Technology, Brno University of Technology, Brno, Czechia
[2] Kempelen Institute of Intelligent Technologies, Bratislava, Slovakia
[3] Faculty of Electrical Engineering and Information Technology, Slovak University of Technology, Bratislava, Slovakia
martin.mocko@kinit.sk jaroslav.kopcan@kinit.sk daniela.chuda@stuba.sk

Abstract. Malware clustering is a valuable technique for organizing large collections of malicious samples, yet its usefulness often depends on whether analysts can understand the reasons behind cluster assignments. Explainable AI (XAI) can help experts make faster and more reliable decisions, but explainability has received little attention in the context of malware clustering. In this extended abstract, we primarily conduct a short review of the state-of-the-art approaches for the task of explainable clustering. The field of explainable clustering has been neglected in the past and has been slowly getting attention in the last few years. This review will help guide us in deciding about the next steps for the task of generating explanations for malware clustering.

Keywords: explainable clustering · malware clustering · interpretability.

1 Research Context

In the malware domain, research focusing on the explainability of clustering is, to the best of our knowledge, as well as according to a recent study [13], unexplored. This limits the understanding of how these models discern patterns and relationships in malware samples. So far, most of the attention regarding explainability has been focused towards malware detection/classification - or in general the typical supervised-learning-type tasks [5], whereas little attention has been paid to the explainability of unsupervised learning. An interesting task is to explain malware detection, which can be implemented through Logic Explained Networks (LENs) [1].

In general, clustering solutions that provide some kind of explainability could be divided into a) interpretable by design (or "In-Clustering") or b) post-hoc approaches [4]. Most clustering methods are not interpretable by design, therefore post-hoc approaches must often be used to make sense about why certain samples form separate clusters. Such post-hoc approaches are often model-agnostic, which increases their general usefulness.

During our research of the related work for clustering explainability, we have observed that most of the papers use decision trees (or some kind of binary

trees) to explain the existing clustering [10, 3, 8, 7, 11]. This also means most of the works are post-hoc explainers. Among the most prominent examples of explanation methods based on trees are IMM [10], Ex-Greedy [8], ExKMC [3], ExShallow [7], and others. The debate in these works is often about how many leaves should the decision tree have to describe the clusters - where often k leaves seems to be the answer.

There are also some explanation methods for clustering that work on a different principle than decision trees. Here, it also makes sense to emphasize the difference between explainability and interpretability of clustering methods as noted by [12]. Explainability often refers to post-hoc explanations by various approaches to enhance the understandability of the model [12]. Different from explainability, interpretability is rooted in the design of the model itself, which is highly expected but also quite challenging. Regarding the different explanation approaches, we identify two approaches which are *interpretable* - a multi-polytope approach [9] and an approach based on sorting [2]. Other types of approaches which are post-hoc (explainable) include two feature attribution approaches [14, 5] and one approach based on counterfactuals [15].

So far, we have not mentioned ontologies, or rather - knowledge graphs, which we also expect to be working with in the near future. Up until very recently, we have not seen algorithms which would be capable of clustering knowledge graphs. This no longer seems to be the case as there is a very new recent publication by Koopmann [6] which aims to do conceptual clustering in which each cluster is described by an $\mathscr{EL}$ concept. We are not aware of any other works or methods which would be able to do semantic clustering based on ontologies and knowledge graphs. The only idea that comes to mind is cluster in a vector space and then use some concept learning approach to explain the learned clusters.

To conclude, the field of explainable unsupervised learning has been relatively neglected in the past and much more attention has been paid to supervised XAI. For malware clustering, we are not aware of any study which tries to make explainable malware clustering. Our analysis of the state-of-the-art has revealed that most of the works on explainable clustering (in general) utilize trees to explain cluster assignments. However, other types of approaches do exist as well. The way towards formulating our solution for explainable malware clustering has to take into account the data and features we would work with, as well as the desired presentation of explanations.

Acknowledgments. Funded by the EU NextGenerationEU through the Recovery and Resilience Plan for Slovakia under the project No. 09I05-03-V02-00064 and by the Slovak Research and Development Agency under the contract No. APVV-23-0292.

References

1. Anthony, P., Giannini, F., Diligenti, M., Homola, M., Gori, M., Balogh, S., Mojzis, J.: Explainable malware detection with tailored logic explained networks (2024), https://arxiv.org/abs/2405.03009

2. Chen, X., Güttel, S.: Fast and explainable clustering based on sorting. Pattern Recognition **150**, 110298 (2024)
3. Frost, N., Moshkovitz, M., Rashtchian, C.: Exkmc: Expanding explainable k-means clustering. arXiv preprint arXiv:2006.02399 (2020)
4. Hu, L., Jiang, M., Dong, J., Liu, X., He, Z.: Interpretable clustering: A survey. arXiv preprint arXiv:2409.00743 (2024)
5. Kauffmann, J., Esders, M., Ruff, L., Montavon, G., Samek, W., Müller, K.R.: From clustering to cluster explanations via neural networks. IEEE transactions on neural networks and learning systems **35**(2), 1926–1940 (2022)
6. Koopmann, P., Bakel, R., Cochez, M.: Towards conceptual clustering in el with simulation graphs - experimental data and extended version (Aug 2025). https://doi.org/10.5281/zenodo.16794054, https://doi.org/10.5281/zenodo.16794054
7. Laber, E., Murtinho, L., Oliveira, F.: Shallow decision trees for explainable k-means clustering. Pattern Recognition **137**, 109239 (2023)
8. Laber, E.S., Murtinho, L.: On the price of explainability for some clustering problems. In: International Conference on Machine Learning. pp. 5915–5925. PMLR (2021)
9. Lawless, C., Kalagnanam, J., Nguyen, L.M., Phan, D., Reddy, C.: Interpretable clustering via multi-polytope machines. In: Proceedings of the aaai conference on artificial intelligence. vol. 36, pp. 7309–7316 (2022)
10. Moshkovitz, M., Dasgupta, S., Rashtchian, C., Frost, N.: Explainable k-means and k-medians clustering. In: International conference on machine learning. pp. 7055–7065. PMLR (2020)
11. de Oliveira, G.S., Silva, F.A., Ferreira, R.V.: Explainable clustering: A solution to interpret and describe clusters. Journal of Information and Data Management **16**(1), 170–180 (2025)
12. Peng, X., Li, Y., Tsang, I.W., Zhu, H., Lv, J., Zhou, J.T.: Xai beyond classification: Interpretable neural clustering. Journal of Machine Learning Research **23**(6), 1–28 (2022)
13. Rui, L., Gadyatskaya, O.: Position: the explainability paradox-challenges for xai in malware detection and analysis. In: 2024 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW). pp. 554–561. IEEE (2024)
14. Scholbeck, C.A., Funk, H., Casalicchio, G.: Algorithm-agnostic feature attributions for clustering. In: World Conference on Explainable Artificial Intelligence. pp. 217–240. Springer (2023)
15. Spagnol, A., Sokol, K., Barbiero, P., Langheinrich, M., Gjoreski, M.: Counterfactual explanations for clustering models. arXiv preprint arXiv:2409.12632 (2024)

Qualitative Evaluation of Explanations for Malware Analysis

Jaroslav Kopčan[1][0000−0002−7467−6615], Martin Mocko[1,2][0000−0001−8982−0141], and Daniela Chudá[1,3][0000−0002−3873−9308]

[1] Kempelen Institute of Intelligent Technologies, Bratislava, Slovakia
[2] Faculty of Information Technology, Brno University of Technology, Brno, Czechia
[3] Faculty of Electrical Engineering and Information Technology, Slovak University of Technology, Bratislava, Slovakia
jaroslav.kopcan@kinit.sk martin.mocko@kinit.sk daniela.chuda@stuba.sk

Abstract. Malware analysis systems can provide valuable insights to security practitioners, but their explanations must be both accurate and understandable to support effective human decision-making. Yet, systematic evaluation of explanation quality in this domain has received limited attention. Therefore, a comprehensive human user evaluation framework that combines qualitative and quantitative assessment methods to measure explanation effectiveness in malware analysis is needed. Through controlled experiments incorporating both performance metrics and user perceptions, the evaluation aims to establish how explanation quality influences analyst decision-making, which explainable methods are most suitable, and to provide guidelines for developing more effective explainable malware analysis systems.

Keywords: evaluation · explainability · malware analysis · user study.

1 Research Context

Human evaluation of explanations in malware analysis represents a specialized area that combines explainable AI, cybersecurity, and human-computer interaction research. While the field of explainable artificial intelligence (XAI) for malware analysis shows quite significant technical advances in recent years, human evaluation assessments of proposed methods remain heavily understudied because of the laborious and rigid work when introducing humans in the loop, often leading to ambiguous results with levels of uncertainty.

Therefore, current research predominantly focuses on algorithmic methods and metrics for explanation quality rather than human-centered evaluation. XAI methods in malware detection falls into multiple categories such as: model-agnostic techniques like SHAP [6] and LIME [10] (both methods are nowadays considered solid baseline in malware analysis domain), model-specific techniques like gradient-based explanations, and inherently interpretable models (decision trees, rule-based systems) [7]. Current solutions for explainable malware analysis also include explanations like input-feature importance, visual explanations, or

rule extraction methods which provide readable logic-rules [1]. Despite successful use of existing XAI method, the coverage of evaluation is insufficient, even for evaluation of vulnerabilities to adversarial attacks which could exploit the explanations itself, underscoring the critical need for thorough evaluation frameworks tailored to domain experts' requirements.[2]

Moreover the absence of systematic human evaluation represents a critical gap in the field. Technical advances demonstrating improved stability and computational efficiency of explanation methods are feasible, but we lack empirical evidence of their practical usefulness for malware analysts and applications. The few studies that incorporate human assessment either target general security contexts [3], or inspect related domains like reverse engineering processes [13].

Key missing elements include large-scale user studies oriented to cybersecurity professionals, task-based evaluations measuring explanation impact on analyst decision-making accuracy and efficiency, and comparative assessments of analyst preferences across different explanation approaches. Furthermore, no standardized metrics exist for evaluating explanation understandability specifically in malware analysis contexts [4].

Other technical domains (e.g. fact-checking), have established robust methodologies for user studies that could work as direct inspiration for malware analysis [11]. For example, user studies in the natural language processing domain usually use comprehensive approach involving both experts and novice participants, multiple explanation types sourced from different methods, and proxy measures. Integral part is evaluation itself, where frameworks measures the fidelity, usefulness, and trust[12] by employing local and global metrics and 'blind trust' scenarios where system intentionally provide user with incorrect predictions to test human over-reliance [8][9]. Similar methodologies have been used in other domains like medical imaging, finance, or autonomous vehicle systems, showing that usefulness of explanations could vary greatly. [5]

In case of malware analysis, the evaluation could be conducted with security analysts with varying expertise levels, testing explanation modalities specific to malware, and even incorporating into evaluation realistic proxy tasks - cybersecurity workflows to put explanations into use. Explainable malware analysis domain introduces also challenges, like high technical nature of explanations, trust-calibration because of high-stakes consequences in case of misclassification and fast evolving landscape of threats[14].

Acknowledgments. Funded by the EU NextGenerationEU through the Recovery and Resilience Plan for Slovakia under the project No. 09I05-03-V02-00064 and by the Slovak Research and Development Agency under the contract No. APVV-23-0292.

References

1. Anthony, P., Giannini, F., Diligenti, M., Homola, M., Gori, M., Balogh, S., Mojžiš, J.: Explainable malware detection with tailored logic explained networks (09 2024)
2. Aryal, K., Gupta, M., Abdelsalam, M.: A survey on adversarial attacks for malware analysis (2022), https://arxiv.org/abs/2111.08223

3. Das, D., Kim, B., Chernova, S.: Subgoal-based explanations for unreliable intelligent decision support systems (2023), https://arxiv.org/abs/2201.04204
4. Kim, J., Maathuis, H., Sent, D.: Human-centered evaluation of explainable ai applications: a systematic review. Frontiers in Artificial Intelligence **Volume 7 - 2024** (2024). https://doi.org/10.3389/frai.2024.1456486, https://www.frontiersin.org/journals/artificial-intelligence/articles/10.3389/frai.2024.1456486
5. Longo, L., Brcic, M., Cabitza, F., Choi, J., Confalonieri, R., Ser, J.D., Guidotti, R., Hayashi, Y., Herrera, F., Holzinger, A., Jiang, R., Khosravi, H., Lecue, F., Malgieri, G., Páez, A., Samek, W., Schneider, J., Speith, T., Stumpf, S.: Explainable artificial intelligence (xai) 2.0: A manifesto of open challenges and interdisciplinary research directions. Information Fusion **106**, 102301 (Jun 2024). https://doi.org/10.1016/j.inffus.2024.102301, http://dx.doi.org/10.1016/j.inffus.2024.102301
6. Lundberg, S., Lee, S.I.: A unified approach to interpreting model predictions (2017), https://arxiv.org/abs/1705.07874
7. Manthena, H., Shajarian, S., Kimmell, J., Abdelsalam, M., Khorsandroo, S., Gupta, M.: Explainable artificial intelligence (xai) for malware analysis: A survey of techniques, applications, and open challenges (2025). https://doi.org/https://doi.org/10.1109/ACCESS.2025.3555926, https://arxiv.org/abs/2409.13723
8. Mohseni, S., Yang, F., Pentyala, S., Du, M., Liu, Y., Lupfer, N., Hu, X., Ji, S., Ragan, E.: Machine learning explanations to prevent overtrust in fake news detection (07 2020). https://doi.org/10.48550/arXiv.2007.12358
9. Nguyen, A.T., Kharosekar, A., Krishnan, S., Krishnan, S., Tate, E., Wallace, B.C., Lease, M.: Believe it or not: Designing a human-ai partnership for mixed-initiative fact-checking. Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology (2018), https://api.semanticscholar.org/CorpusID:52239205
10. Ribeiro, M.T., Singh, S., Guestrin, C.: "why should i trust you?": Explaining the predictions of any classifier (2016), https://arxiv.org/abs/1602.04938
11. Schmitt, V., Csomor, B.P., Meyer, J., Villa-Areas, L.F., Jakob, C., Polzehl, T., Möller, S.: Evaluating human-centered ai explanations: Introduction of an xai evaluation framework for fact-checking. p. 91–100. MAD '24, Association for Computing Machinery, New York, NY, USA (2024). https://doi.org/10.1145/3643491.3660283, https://doi.org/10.1145/3643491.3660283
12. Schmitt, V., Villa-Arenas, L.F., Feldhus, N., Meyer, J., Spang, R., Möller, S.: The role of explainability in collaborative human-ai disinformation detection. pp. 2157–2174 (06 2024). https://doi.org/10.1145/3630106.3659031
13. Votipka, D., Rabin, S.M., Micinski, K., Foster, J.S., Mazurek, M.M.: An observational investigation of reverse engineers' processes. In: Proceedings of the 29th USENIX Conference on Security Symposium. SEC'20, USENIX Association, USA (2020)
14. Warnecke, A., Arp, D., Wressnegger, C., Rieck, K.: Evaluating explanation methods for deep learning in security (2020), https://arxiv.org/abs/1906.02108

Counterfactual Explanations to Detect Adversarial Vulnerability of Malware Classifiers

Iveta Bečková[0000−0002−6396−9770]

Faculty of Mathematics, Physics and Informatics
Comenius University in Bratislava
Mlynská dolina, 84248 Bratislava, Slovakia
iveta.beckova@fmph.uniba.sk

Abstract. Explainability has gained great attention of machine learning-focused research, to achieve multiple goals such as increasing trustworthiness and fairness of machine learning models or mitigating their issues like adversarial vulnerability.

Most explainability methods produce explanations in the form of feature importance scores or logical rules. This is not the case for *counterfactual explanations* [3], which are local explanations taking the form of statements "If the value of feature ... were changed to ..., then the output would change to ...".

The process of creating counterfactual explanations is similar to the optimization performed while searching for the so-called *adversarial examples* [4], which are inputs causing misclassification despite being similar to correctly classified inputs from the dataset.

In the domain of malware detection, adversarial examples need to follow some constraints. For example, when modifying a Windows Portable Executable file, the modification must preserve its functionality. The modification also needs to be subtle enough to preserve the character of the file (i.e., cannot cause malware to become benign or vice versa – an undesirable effect commonly referred to in the literature as *true label change*).

Some of the admissible modifications include adding a file section, adding an imported function, appending bytes (thus modifying entropy of a section or an entire file), modifying a section name etc. [1]. In a high-level feature representation like a vectorization of an ontology [2], such change corresponds to changing a single feature vector element from 0 to 1.

If a classifier can be locally explained by counterfactual explanations corresponding to changes of this kind, for example "This PE file is malware but if it had one more section with high entropy, then it would be benign." it hints that it is possible to craft adversarial inputs fooling this classifier.

Though a potential attacker may not have access to the high-level feature representation, which makes finding adversarial examples more challenging, existence of such vulnerabilities is still concerning. Therefore, using counterfactual explanations to detect such flaws can provide a valuable information to an expert user.

Keywords: Malware detection · Adversarial examples · Counterfactual explanations.

Acknowledgments. This research was funded by the EU NextGenerationEU through the Recovery and Resilience Plan for Slovakia under the project No. 09I05-03-V02-00064.

References

1. Anderson, H.S., Kharkar, A., Filar, B., Roth, P.: Evading machine learning malware detection. black Hat **2017**, 1–6 (2017)
2. Homola, M., Anthony, P., Bečková, I., Kľuka, J., Mojžiš, J., Švec, P., Balogh, Š., Cardillo, F.A., Debole, F., Straccia, U., Kenyeres, M., Giannini, F., Diligenti, M., Gori, M., Bisták, T., Trizna, D., Adams, Z.: A note on methods for explainable malware analysis. In: Accepted to Semantic Shields 2 (to appear)
3. Stepin, I., Alonso, J.M., Catalá, A., Pereira-Fariña, M.: A survey of contrastive and counterfactual explanation generation methods for explainable artificial intelligence. IEEE Access **9**, 11974–12001 (2021)
4. Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., Fergus, R.: Intriguing properties of neural networks. In: International Conference on Learning Representations (2014), http://arxiv.org/abs/1312.6199

Transforming Malware Ontology via Self-Attention

Štefan Pócoš[0000−0003−3799−7038] and Iveta Bečková[0000−0002−6396−9770]

Faculty of Mathematics, Physics and Informatics
Comenius University Bratislava
Mlynská dolina F1, 842 48 Bratislava, Slovak Republic
{stefan.pocos,iveta.beckova}@fmph.uniba.sk

Abstract. Malware detection remains a critical challenge in cybersecurity, demanding continued research and innovation. Despite years of dedicated effort, novel approaches continue to emerge, reflecting the evolving complexity of malicious software. While some methods can work with semantic-rich data representations such as ontologies, most machine learning algorithms operate on vectorized data. However, for a fair comparison between various approaches, different data representations also need to be comparable in the sense that there should exist a meaningful transformation between them. Considering some representations are necessarily more expressive than others, the goal becomes finding such a transformation that preserves as much information as possible.

The EMBER dataset [3] has become a valuable resource in this domain, as it provides a robust foundation for distinguishing between benign and malicious executables. In this work, we focus on transforming the existing EMBER ontology [1] into a vectorized representation specifically tailored for use with self-attention architectures.

Our proposed representation consists of two main streams: (a) vector $\mathbf{v}$ representing the file actions, file features (we omit the derived features), and data properties; (b) vectors $\mathbf{s}_1, ..., \mathbf{s}_n$ each representing a single section along with its type, features and flags. Since the number of sections of Portable Executable (PE) files is variable, standard machine learning architectures are not applicable to these data. One could create a simplified representation for example by aggregating information about all present sections and creating boolean features corresponding to the existence of a section with a certain combination of type, features, and flags. This approach would yield a constant input size for all PE files for the cost of losing information about individual sections, which can be undesirable.

To leverage our representation, we propose to process the data via a self-attention structure presented in [2]. Our embedding design consists of a three-step process: (1) each of the input values is transformed using a learnable *embedding projection*; (2) we add a *type embedding* to the projected inputs, which serves as a distinction between various data types included in the ontology; and (3) to further promote the diversity of the input source carrying different semantic meaning, we also add *positional embedding*. Using these three steps, we ensure that the varying number of

sections (and input vectors) does not influence the inference, since more input vectors do not require any additional trained parameters.
The created embeddings maintain a clear correspondence to their original semantic meaning. This means that, during classification (and any subsequent analysis), we can inspect the self-attention mechanism and extract the cross-relations of the input vectors, making the process highly explainable.

Keywords: Ontology · Attention · Explainability

Acknowledgments. This research was funded by the EU NextGenerationEU through the Recovery and Resilience Plan for Slovakia under the project No. 09I05-03-V02-00064.

References

1. Švec, P., Balogh, Š., Homola, M., Kľuka, J., Bisták, T.: Semantic data representation for explainable windows malware detection models. arXiv:2403.11669 (2024)
2. Vaswani A., Shazeer N., Parmar N., Uszkoreit J., Jones L., Gomez N. A., Kaiser Ł, and Polosukhin I.: Attention is all you need. Advances in Neural Information Processing Systems 30 (2017)
3. H. S. Anderson, P. Roth, EMBER: an open dataset for training static PE malware machine learning models. arXiv:1804.04637 (2018)

On the Prospects of $\mathcal{EL}$ and $\mathcal{ELU}$ Concept Learners for Explainable Malware Detection

Martin Demovič[1][0009−0004−2012−4526], Peter Švec[2][0000−0002−8315−5301], Martin Homola[1][0000−0001−6384−9771], and Maurice Funk[3][0000−0003−1823−9370]

[1] Comenius University in Bratislava, Mlynská dolina, 84248 Bratislava, Slovakia
demovic17@uniba.sk, homola@fmph.uniba.sk
[2] Slovak Technical University, Ilkovičova 3, 84104 Bratislava, Slovakia
peter.svec1@stuba.sk
[3] Leipzig University, Augustusplatz 10, Leipzig, 04109, Germany

Keywords: Explainable AI · Malware analysis · Concept learning.

Interpretable classification is crucial in security-critical domains such as malware detection. Structured machine learning (SML) [11] enables the learning of *symbolic classifiers* [4, 5, 11], logical formulas that evaluate to true or false for a given input sample, depending on whether it satisfies the learned concept. This binary, rule-based nature makes symbolic classifiers inherently interpretable. As a result, they offer intrinsic explainability that is often preferable to post-hoc methods like LIME [10] or SHAP [8], which attempt to approximate the decision boundaries of black-box models.

We present a comparative evaluation of concept learning systems for description logic ($\mathcal{EL}$ and $\mathcal{ELU}$) over real-world malware datasets derived from EMBER [1] and the PE Malware Ontology [12]. We benchmark DL-Learner [2], SPELL [3], and ALC-SAT [7] across increasing dataset sizes (up to 20,000 samples) using standard classification metrics.

While expressive learners such as OCEL and CELOE, available within the DL-Learner framework, achieve better approximation performance, they often produce concepts that are harder to interpret. In contrast, $\mathcal{EL}$-based learners like SPELL and ELTL rely on simpler, less expressive constructors, resulting in hypotheses that are more easily understandable to humans, albeit with varied trade-offs in F1 score and false positive rate.

Our results confirm the scalability and practical feasibility of recent SAT-based approaches (SPELL, ALC-SAT), and show that interpretable concept learning can be effectively applied to large-scale malware datasets, offering a compelling alternative to black-box classifiers in security contexts [6, 9].

Acknowledgments. The coauthors wish to thank Jean Christoph Jung. Funded by the EU NextGenerationEU through the Recovery and Resilience Plan for Slovakia under the project No. 09I05-03-V02-00064.

References

1. Anderson, H.S., Roth, P.: EMBER: an open dataset for training static PE malware machine learning models. CoRR **abs/1804.04637** (2018), http://arxiv.org/abs/1804.04637
2. Bühmann, L., Lehmann, J., Westphal, P.: DL-learner - A framework for inductive learning on the semantic web. J. Web Semant. **39**, 15–24 (2016). https://doi.org/10.1016/J.WEBSEM.2016.06.001
3. ten Cate, B., Funk, M., Jung, J.C., Lutz, C.: SAT-based PAC learning of description logic concepts. In: Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI 2023, 19th-25th August 2023, Macao, SAR, China. pp. 3347–3355. ijcai.org (2023). https://doi.org/10.24963/IJCAI.2023/373
4. Darwiche, A.: Three modern roles for logic in AI. In: Proceedings of the 39th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems, PODS 2020, Portland, OR, USA, June 14-19, 2020. pp. 229–243. ACM (2020). https://doi.org/10.1145/3375395.3389131
5. Darwiche, A.: Logic for explainable AI. In: 38th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2023, Boston, MA, USA, June 26-29, 2023. pp. 1–11. IEEE (2023). https://doi.org/10.1109/LICS56636.2023.10175757
6. Dolejš, J., Jureček, M.: Interpretability of machine learning-based results of malware detection using a set of rules. In: Artificial Intelligence for Cybersecurity, pp. 107–136. Springer International Publishing (2022). https://doi.org/10.1007/978-3-030-97087-1_5
7. Funk, M., Jung, J.C., Voellmer, T.: SAT-based bounded fitting for the description logic $\mathcal{ALC}$ (extended abstract). In: 38th International Workshop On Description Logics, Opole, Poland (2025)
8. Lundberg, S.M., Lee, S.: A unified approach to interpreting model predictions. In: Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA. pp. 4765–4774 (2017), https://proceedings.neurips.cc/paper/2017/hash/8a20a8621978632d76c43dfd28b67767-Abstract.html
9. Mills, A., Spyridopoulos, T., Legg, P.: Efficient and interpretable real-time malware detection using random-forest. In: 2019 International conference on cyber situational awareness, data analytics and assessment (Cyber SA). pp. 1–8 (2019)
10. Ribeiro, M.T., Singh, S., Guestrin, C.: "why should I trust you?": Explaining the predictions of any classifier. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016. pp. 1135–1144. ACM (2016). https://doi.org/10.1145/2939672.2939778
11. Westphal, P., Bühmann, L., Bin, S., Jabeen, H., Lehmann, J.: Sml-bench - A benchmarking framework for structured machine learning. Semantic Web **10**(2), 231–245 (2019). https://doi.org/10.3233/SW-180308
12. Švec, P., Balogh, S., Homola, M., Kluka, J., Bisták, T.: Semantic data representation for explainable windows malware detection models. CoRR **abs/2403.11669** (2024). https://doi.org/10.48550/ARXIV.2403.11669

Preliminary User Study on Concept Expressions for Characterizing Malware

Martin Homola[1][0000−0001−6384−9771], Peter Švec[2][0000−0002−3406−3574], Ján Kľuka[1][0000−0002−8315−5301], and Štefan Balogh[2][0000−0003−0634−9476]

[1] Comenius University in Bratislava, Mlynská dolina, 84248 Bratislava, Slovakia
homola|kluka@fmph.uniba.sk
[2] Slovak Technical University, Ilkovičova 3, 84104 Bratislava, Slovakia
peter.svec1|stefan.balogh@stuba.sk

Keywords: Explainable AI · Malware analysis · User study.

A number of works showed how symbolic [4, 6, 8, 10] and neuro-symbolic [2, 9] machine learning methods enable to output explanations in form of expressions in a formal language that characterize the malware samples. These works build on the assumption that such expressions offer good understandability for malware analysts – not only to understand the features by which data instances are classified as malware or benign and also *the exact logic of this decision.*

While we side with this assumption, it also needs to be verified by user studies. We report on a preliminary user survey on five independent security analysts. The participants were presented five expressions learned over the EMBER dataset [1] and PE Malware Ontology [11] using the algorithms OCEL, CEOE, PARCEL and SPACEL[3] obtained by DL-Learner [3]. The expressions were presented in the DL syntax and transliterated to English. With each expression, the participants answered the following questions:

- *Q1. Is the given justification indicative of a sample possibly being malware?*
- *Q2. Does the reading of the formula or its transliteration help you to understand why the system classifies the samples as malware/benign?*
- *Q3. Is an explanation/justification of this form useful compared to black-box malware detection methods or compared to post-hoc explainers such as LIME or SHAP?*

The answers were selected from the scale: *Yes – Yes, to some extent – No.* The average answers over the five expressions and over the five participants are shown in the chart in Figure 1. We have computed also the average weighted score for each question (given in brackets), where the three answers were assigned the weight of 100%, 50%, and 0%, respectively. We observe that the first two answers largely dominate against the clearly dismissive third answer. Particularly

[3] The expressions obtained from the parallel algorithms were partial, corresponding the selected disjuncts from the overall learned expression. The participants were informed about this fact.

encouraging for us are the results for Q2, the affirmative answer "Yes" prevails and the overall weighted score is 80%.

To a lesser extend the results (Q3) confirmed that the expressions offer better interpretability compared to what the participants would expect from post-hoc methods like LIME [7] and SHAP [5] (although such explanations were not provided).

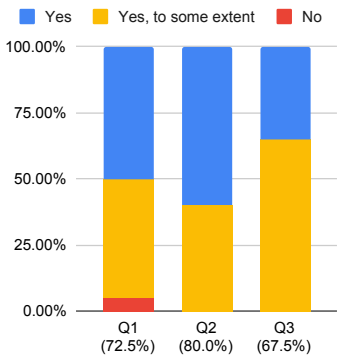


Fig. 1: User study results

The survey is preliminary with respect to the range of expressions being evaluated by the participants. Different applicable methods possibly result in expressions of varying length and expressivity of the language and it is vital to understand the properties of a well suited and useful characteristic expressions that is both informative and well understandable by the users. It is also interesting to compare different parameters for human language transliteration and to pitch the expressions against explanations obtained by different baseline methods. We plan to investigate these issues by more refined user studies in the future.

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

References

1. Anderson, H.S., Roth, P.: EMBER: an open dataset for training static PE malware machine learning models. CoRR **abs/1804.04637** (2018)
2. Anthony, P., Giannini, F., Diligenti, M., Homola, M., Gori, M., Balogh, Š., Mojžis, J.: Explainable malware detection with tailored logic explained networks. In: XKDD (2024), to appear
3. Bühmann, L., Lehmann, J., Westphal, P.: DL-learner - A framework for inductive learning on the semantic web. J. Web Semant. **39**, 15–24 (2016)
4. Cardillo, F.A., Debole, F., Straccia, U.: PN-OWL: A two-stage algorithm to learn fuzzy concept inclusions from OWL2 ontologies. Fuzzy Sets Syst. **490** (2024)
5. Lundberg, S.M., Lee, S.: A unified approach to interpreting model predictions. In: Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA. pp. 4765–4774 (2017)
6. Mojžiš, J., Kenyeres, M.: Interpretable rules with a simplified data representation - a case study with the ember dataset. In: CoMeSySo (2023)
7. Ribeiro, M.T., Singh, S., Guestrin, C.: "Why should I trust you?": Explaining the predictions of any classifier. In: KDD. ACM (2016)
8. Švec, P., Balogh, Š., Homola, M.: Experimental evaluation of description logic concept learning algorithms for static malware detection. In: ICISSP (2021)
9. Trizna, D., Anthony, P., Homola, M., Adams, Z., Balogh, Š.: Learning explainable malware characterization using knowledge base embedding. In: NIGERCON (2024)

10. Švec, P.: Ontologická reprezentácia pre bezpečnosť informačných systémov. Phd thesis, STU, Bratislava, Slovakia (2024)
11. Švec, P., Balogh, S., Homola, M., Kľuka, J., Bisták, T.: Semantic data representation for explainable windows malware detection models. CoRR **abs/2403.11669** (2024)

On the Machine Learning Utilization for Concept Learning in Malware Domain

Ján Mojžiš[1[0000-0002-2196-2271]] and Martin Kenyeres[1[0000-0003-2430-1126]]

[1] Institute of Informatics, Slovak Academy of Sciences, Dúbravská cesta 9,
845 07 Bratislava, Slovak Republic
{jan.mojzis,martin.kenyeres}@savba.sk

Concept learning, in contrast to classical machine learning, draws its resources from ontological concepts and mathematical modeling of conceptual space. In comparison to classical machine learning, concept learning can use advanced approaches such as reasoning. However, due to its extensive advanced mathematical space modeling and reasoning, the performance of common concept learning implementations lags behind those of classical machine learning. To address this issue, we utilize the performance advantage of classical machine learning with mathematical modeling of concept learning algorithms.

From classical machine learning, we propose algorithms based on two different families.

1. Tree-based algorithms, such as Random tree [1] and C4.5 [2].
2. Black-box algorithm families of neural networks.

With the black-box algorithm, we propose three different hierarchical tree reconstruction algorithms.

1. TREPAN [3]
2. Random tree [1]
3. C4.5 [2]

We already evaluated the performance of the C4.5 algorithm in several of our former works [4, 5]. The C4.5 tree model was constructed upon data from malware domain and then the rules were extracted from the tree structure of C4.5 model. In Figure 1, there is an example of one part of the C4.5 tree model with some rules extracted in Table 1. In this example, the leaf is covering the space of benign samples with 39,548 true positive and 336 false positive cases. We have extensively evaluated multiple hierarchical or rule-based models in our former work [4] with the objective of lowest false positive rate. Based on that, the best model is C4.5.

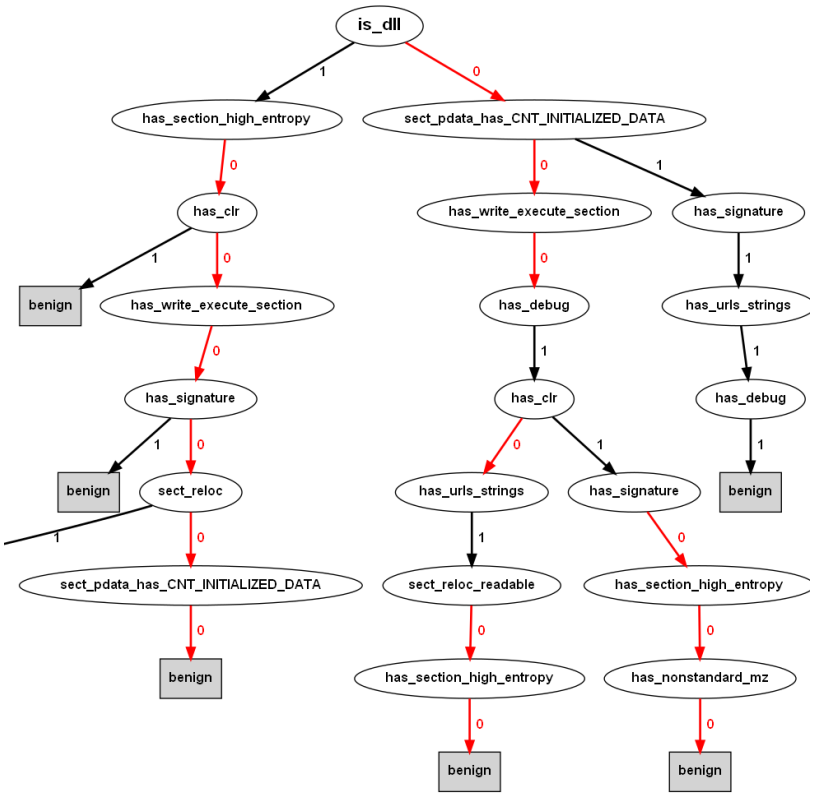


Fig. 1. Resulting, small part of C4.5 tree as stated in [5].

Table 1. Example of rules extracted from tree (Figure 1).

PROPERTY	*VALUE*
is_dll	1
has_section_high_entropy	0
has_clr	1

Based on our results, the classical machine learning models, such as C4.5 can model the feature space and the results are extracted rules that may further serve for concept learning algorithms to alleviate issues of weak performance. In the future work we would like to evaluate the TREPAN algorithm, on results based on black-box models.

Acknowledgments. This work was supported by the Slovak Scientific Grant Agency VEGA under the contract 2/0135/23 “Intelligent sensor systems and data processing” and by the Slovak Research and Development Agency under the contract No. SK-SRB-23-0038 and No. APVV-23-0292.

References

1. Le Gall, Jean-François. "Random trees and applications." (2005): 245-311.
2. Quinlan, J. Ross. C4. 5: programs for machine learning. Elsevier, 2014.
3. Confalonieri, Roberto, et al. "Trepan reloaded: A knowledge-driven approach to explaining black-box models." ECAI 2020. IOS Press, 2020. 2457-2464.
4. Mojžiš, Ján, and Martin Kenyeres. "Interpretable rules with a simplified data representation-a case study with the ember dataset." Proceedings of the Computational Methods in Systems and Software. Cham: Springer International Publishing, 2023. 1-10.
5. Mojžiš J. "On the Possibility of Interpretable Rules Generation for the Classification of Malware Samples. Industry 4.0." 2022;7(6):248-250.

Explainable Malware Detection: Integration of LIME and SHAP into a Dynamic Analysis Pipeline

Matej Skulský[1][0009-0007-6891-2443]

Institute of Informatics and Mathematics, Faculty of Electrical Engineering and Information Technology, Slovak University of Technology, Bratislava, Slovakia
matej.skulsky@stuba.sk

Keywords: Explainable Artificial Intelligence (XAI), malware detection, LIME, SHAP, Karton, Drakvuf.

For today's cybersecurity, AI has become essential to advance the detection of malware. However, Machine learning (ML) provide high accuracy, but operate as "black boxes", making their decision-making processes opaque to human analysts. In general, AI models are not suitable for precise workloads and need human oversight to achieve high accuracy. The lack of transparency is raised as a problem to adopt ML in critical security work, where understanding the rationale behind a detection is crucial [1] for pushing the precision. Explainable Artificial Intelligence (XAI) aims to eliminate this problem by making model decisions interpretable. This paper proposes an end-to-end pipeline for explainable malware detection focusing on dynamic analysis. The core of our solution is the integration of the Karton analysis orchestration system with the Drakvuf dynamic analysis tool, enabling safe and sandboxed collection of behavioral data. We then integrate model-agnostic XAI methods, specifically LIME [2] and SHAP [3], to generate human-understandable explanations for each classification, identifying the key malicious behaviors.

The expected outcome is a functional pipeline that allows security analysts to not only accurately detect malware but also provides important, interpretable insights from toolbox that integrates XAI methods with multiple tools for static and dynamic analysis.

Acknowledgement: This research was funded by the EU NextGenerationEU through the Recovery and Resilience Plan for Slovakia under the project No. 09I05-03-V02-00064.

References

1. Arrieta, A.B., Díaz-Rodríguez, N., Del Ser, J., et al.: Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. Information Fusion (2020)

2. Ribeiro, M.T., Singh, S., Guestrin, C.: "Why Should I Trust You?": Explaining the Predictions of Any Classifier. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, (2016)
3. Lundberg, S.M., Lee, S.I.: A Unified Approach to Interpreting Model Predictions. In: Advances in Neural Information Processing Systems 30, (2017)
4. CERT Polska: Karton — microservice-based malware analysis framework. GitHub. https://github.com/CERT-Polska/karton (accessed September 10, 2025)
5. CERT Polska: Drakvuf — virtual machine-based dynamic analysis framework. GitHub. https://github.com/CERT-Polska/drakvuf (accessed September 10, 2025)

X-MalNet: A Novel Multi-Level e**X**plainability Framework for **Mal**ware Detection Using Matrix Product States (MPS) Tensor **Net**works

Peter Anthony*[a], Philip Wilson[b], Zekeri Adam[a]

[a] *Comenius University, Mlynská dolina, Bratislava, Slovakia*
[b] *Technische Universität Berlin, Marchstraße 23, Berlin, 10587, Germany*

Abstract

Effective explainable malware detection framework requires holistic explanations that captures different levels of abstractions, including feature importance, contextual or logical dependencies (interactions), and distilling them into precise rules. However, existing interpretability approaches offer only partial perspectives. Linear models reveal feature importance but miss dependencies; post-hoc explainers such as LIME and SHAP provide local attributions without contextual logic; and rule-based methods like Anchor and LORE yield decision rules without clarifying feature significance or interactions. This fragmentation prevents security analysts from gaining the complete understanding needed for trust, actionable insights, and system improvement. We propose X-MalNet, a novel framework that bridges this gap through an inherently interpretable model based on Matrix Product States (MPS) tensor networks, achieving high detection accuracy while natively generating a unified suite of explanations from a single, coherent architecture. From its core tensor decomposition, we derive: (1) exact first-order feature importance scores; (2) second-order feature interactions quantified via the entanglement spectrum from Schmidt decomposition, revealing non-linear logical dependencies; and (3) minimal, precise rule-based explanations extracted analytically by fixing feature values and marginalizing the network. Evaluated on binarized malware datasets, X-MalNet can deliver a holistic suite of faithful explanations by design without sacrificing performance.

Keywords: Malware Detection, Explainable AI (XAI), Tensor Networks

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

An Analysis of the EMBER Datasets's Evolution

Stanislava Pecková[1][0009−0007−7689−4126]

Institute of Computer Science and Mathematics, Faculty of Electrical Engineering and Information Technology Slovak University of Technology, Ilkovicova 3, 81219 Bratislava, Slovakia
stanislava.peckova@stuba.sk

Machine learning models require datasets to train on. The quality and quantity of data directly correlate with a model's results. The EMBER project provides high-quality classification data and representative samples of malware. The recent EMBER2024 release significantly improved upon its predecessor, EMBER2018. The dataset has expanded to over 3.2 million samples across six file formats (Win32, Win64, .NET, APK, ELF, and PDF) and supports seven distinct classification tasks, including malware family identification, whereas EMBER2018 contained only 1 million samples in the PE format. The new version also provides a curated *challenge set* of malware that successfully went undetected by more than 70 conventional antivirus engines [1].

Evaluation of the challenge set provides an interesting insight: state-of-the-art models achieve near-perfect accuracy on the standard test set (PR-AUC ≈ 0.997), but their performance drops considerably on the challenge set (PR-AUC ≈ 0.572) [2].

The following Table 1 provides a comparison of the properties of the EMBER datasets.

Table 1. Comparison of EMBER Datasets

Property	**EMBER 2017**	**EMBER 2018**	**EMBER2024**
Total Files	1.1 million	1.0 million	>3.2 million
File Types	PE files only	PE files only	Win32, Win64, .NET, APK, ELF, PDF
Feature Version	Version 1	Version 2	Version 3
Feature Extraction Library	LIEF (v0.8.3)	LIEF (v0.9.0)	pefile ("thrember")
Number of Labels/Tags	1 (malicious/benign)	1 (malicious/benign)	7 (malicious/benign, family, behavior, etc.)
Inclusion of Challenge Set	No	No	Yes
Dataset Split	Temporal (collected pre-2017)	Temporal (collected pre-2018)	Temporal (52 weeks train, 12 weeks test)

Table 1 shows how EMBER2024 goes further than the earlier releases. The earlier datasets only covered PE files with a basic malware–benign split, whereas EMBER2024 introduces diverse file types, broader label categories, and even a curated *challenge set*. All of these updates make EMBER2024 a tougher and more realistic test for modern malware classifiers.

Keywords: Malware Detection, Machine Learning, EMBER Dataset

Acknowledgments. This research was funded by the EU NextGenerationEU through the Recovery and Resilience Plan for Slovakia under the project No. 09I05-03-V02-00064.

References

1. C. Zhang - EMBER2024: A New Benchmark for Holistic Malware Classification, *Medium*, 2025. [Online]. https://medium.com/@zhanghaolin66/ember2024-a-new-benchmark-for-holistic-malware-classification-62dcb260b47a, last accessed 2025/9/10
2. R. J. Joyce, G. Miller, P. Roth, R. Zak, E. Zaresky-Williams, H. Anderson, E. Raff, and J. Holt - EMBER2024 – A Benchmark Dataset for Holistic Evaluation of Malware Classifiers, https://arxiv.org/pdf/2506.05074, 2025.

LLM and interpretability in security domain

Štefan Balogh, Juraj Paška, and Peter Švec

Faculty of Electrical Engineering and Information Technology, Slovak University of Technology, Ilkovičova 3, 841 04 Bratislava, Slovakia
{stefan.balogh,juraj.paska}@stuba.sk

Abstract. Recent advancements in malware detection emphasize explainable methods to improve transparency and trust in machine learning models. Using RAG in LLMs enhances accuracy, reduces hallucinations, and adds interpretability. A notable approach integrates RAG with knowledge graphs – Graph RAG. This study explores these techniques and presents experimental results demonstrating RAG's potential.

Keywords: ontology · explainable methods · malware detection.

1 Introduction

Cybersecurity increasingly demands AI systems that are both accurate and interpretable. Large Language Models (LLMs) excel at natural language processing but operate as black boxes, often hallucinating information. Knowledge Graphs (KGs), by contrast, provide structured, semantically rich data that is fully interpretable [1]. Their integration addresses LLM weaknesses by combining flexible reasoning with factual grounding.

Two main interaction modes exist: LLMs can help build KGs, and KGs can serve as external knowledge sources for LLMs. The latter approach often uses Natural Language Querying (NLQ), implemented through Retrieval-Augmented Generation (RAG), prompt-to-query, and fine-tuning [2]. Among these, RAG stands out for improving accuracy without retraining. Ontologies further enhance interpretability by enabling traceability and reducing inconsistencies. Research such as Microsoft's OG-RAG [3] and neuro-symbolic approaches [4] confirms that combining LLMs, KGs, and ontologies is key to building transparent and trustworthy AI systems.

2 Retrieval-Augmented Generation and GraphRAG

RAG supplements prompts with external information, reducing hallucinations and improving relevance. Two retrieval strategies dominate: vector-based retrieval, which embeds queries and data into a shared vector space, and prompt-to-query retrieval, which generates structured queries for KGs [2]. GraphRAG extends these principles by leveraging KG semantics for more accurate retrieval. Hybrid approaches combine both methods, using vector similarity for initial filtering and KG queries for refinement. This design improves precision and supports explainability, as users can inspect retrieved data and query logic.

3 Application to Malware Analysis

Malware evolves rapidly, making static detection models insufficient. RAG enables LLMs to access technical details – assembly code, binary metadata, YARA rules – without retraining, ensuring timely and accurate analysis. The Malware Analysis RAG project illustrates this approach using Meta's LLaMA 3 model. The system retrieves relevant information from a vector database containing embeddings of assembly code segments and augments the LLM's prompt, allowing it to answer malware-related queries with greater precision.

Implementation Details. The effectiveness of RAG depends on several factors. Knowledge base freshness is critical, as outdated data compromises detection. Retrieval algorithm accuracy determines whether relevant information is selected efficiently. Another key aspect is document segmentation (chunking). Malware reports and assembly code files are often lengthy; splitting them into smaller chunks (e.g., 20 lines of code) ensures granular retrieval preserving semantic coherence. Poor chunking can lead to irrelevant or incomplete context, reducing classification accuracy.

Unlike fine-tuning, RAG does not modify model parameters, allowing flexible updates to the knowledge base without additional training – a major advantage in fast-changing domains. The system uses embeddings stored in a vector database for initial retrieval, followed by augmentation of the LLM prompt with the most relevant code fragments and metadata. This design ensures that the model operates with contextually rich information while maintaining efficiency.

Experimental Results. Evaluation compared LLaMA 3.2 3B with and without RAG on 57 samples (31 malicious, 26 benign). Each input file was processed as a batch of smaller fragments, and final classification was derived from aggregated predictions. Results showed substantial improvements with RAG:

- Accuracy rose from 52% to 67%, indicating more correct predictions overall.
- Precision increased from 70% to 92%, reducing false positives significantly.
- Recall nearly doubled, from 22% to 41%, capturing more true malicious files.
- F1 score improved from 34% to 57%, confirming better balance between precision and recall.

These gains demonstrate that adding contextual information from a vector database enables smaller models to analyze and classify assembly code more effectively. Figures in the original study illustrate interpretable answers based on hashes and YARA rules, reinforcing the value of RAG for transparency.

4 Conclusion

Integrating RAG – especially GraphRAG – with LLMs represents a major step toward interpretable and effective AI systems in cybersecurity. By leveraging

external knowledge sources, these systems deliver accurate, context-aware responses without costly retraining. Experimental evidence from malware analysis confirms the practical value of this approach, particularly in improving precision and recall. Future research should refine hybrid retrieval strategies and deepen ontology integration to further enhance transparency and trustworthiness.

Acknowledgments. This research was sponsored by the Slovak Republic under grants APVV-23-0292 (Dynamic Malware Analysis with Explainable AI).

References

1. Steve Hedden, *How to Implement Knowledge Graphs and Large Language Models (LLMs) Together at the Enterprise Level*, Towards Data Science, April 19, 2024. Available at: https://towardsdatascience.com/how-to-implement-knowledge-graphs-and-large-language-models-llms-together-at-the-enterprise-level-cf2835475c47 (Accessed: September 10, 2025).
2. Steve Hedden, *How to implement Graph RAG using knowledge graphs and vector databases*, Towards Data Science, 2024. Available at: https://towardsdatascience.com/how-to-implement-graph-rag-using-knowledge-graphs-and-vector-databases-60bb69a22759/ (Accessed: September 10, 2025).
3. Microsoft Research, "OG-RAG: Ontology-Grounded Retrieval-Augmented Generation for Large Language Models," *arXiv preprint arXiv:2412.15235*, 2025. Available: https://arxiv.org/abs/2412.15235.
4. E. Magana and A. Monti, "Enhancing Large Language Models through Neuro-Symbolic Integration and Ontological Reasoning," *arXiv preprint arXiv:2504.07640*, 2025. Available: https://arxiv.org/abs/2504.07640.
5. CYBERSCIENCELAB. *Malware Analysis RAG*. 2024. Available at: https://github.com/CyberScienceLab/Malware_Analysis_Rag. Accessed: 2025-05-12.

Converting malware reports into ontology: progress report

Roderik Ploszek[0000−0002−3192−0630] and Matúš Jókay[0000−0002−8058−9294]

Slovak University of Technology in Bratislava, Slovakia
Institute of Computer Science and Mathematics
{roderik.ploszek, matus.jokay}@stuba.sk

1 Hybrid Analysis and ontology

This abstract describes the development of a conversion script for converting data from the malware analysis tool *Hybrid Analysis* [2] into ontology form. Ontology representation is ideal for machine processing, where inference algorithms can be used to discover new knowledge about the dataset that was not previously not explicitly available in the dataset. This is well suited for the field of malware analysis, where it is possible to identify typical characteristics of malware, and create better models for detection mechanisms.

The input to the conversion script consists of reports from Hybrid Analysis. This tool performs a combination of static and dynamic analysis of samples, thus the name, *Hybrid* Analysis. The reports are generated by *CrowdStrike Falcon Sandbox*. The report includes general information about the executable file, such as its size, specific type, sections, and strings found within it. During analysis, the file is executed in a sandbox environment and all its activities are recorded, including system calls, created files, and accessed registry entries. From this low-level data, the sandbox then extracts indicators that indicate the maliciousness of the sample. This may include, for example, writing data to a remote process, the presence of malicious strings, or bundling of additional executable files.

Almost all information that Hybrid Analysis extracts is available in a JSON structure that can be downloaded via API. However, more detailed information such as file accesses, system calls, and registry accesses are currently not available in the JSON. This data has to be parsed from the HTML report, which is downloaded separately.

The ontology into which the sample analyses are converted was based on the MAEC standard [3], which is a standard for malware representation. After analyzing the data structure of Hybrid Analysis output, it was found that certain data points cannot be represented using MAEC. Therefore, classes specific to Hybrid Analysis were added to the ontology. The result was the MAECO ontology [1], into which samples are converted.

2 The conversion script

The script processes the JSON structure generated from Hybrid Analysis directly. A data-oriented programming approach is used for processing, where the JSON

structure is explicitly visible in the code and the processing of individual items is delegated to helper functions. The main control object is a dictionary that follows the structure of the input JSON, as shown in the example below:

```
{...
'size': (submission_file, set_value, 'size'),
'image_base': (header, set_hex, 'imageBase'),
'classification_tags': (instance, append_list, 'label',
                        get_or_create_individual, maeco,
                        maeco.MalwareLabel)
...}
```

Parsing is executed according to data stored in the dictionary. The dictionary keys specify which key from the JSON report will be processed. The value of each key is a tuple of variable length. The first element denotes the ontology object, the second specifies the method used to parse the value from the report (which varies depending on the complexity of the specific data), and the third element is the property name of the ontology object where the value will be stored. Additional elements, if present, serve as supplementary arguments for the parsing method. This approach enables efficient processing of the JSON structure while maintaining code readability similar to the original structure. It also facilitates reuse of parsing methods and seamless extensions when needed.

3 Challenges

The conversion script is currently work in progress. Some examples of challenges that have been solved in the script follow.

The field values from Falcon Sandbox analysis results are practically undocumented. In the case of enumeration types (e.g., analysis environments), it is not possible to determine what values the analysis can return without documentation. We partially solved this problem by going through all the analyses we have downloaded and identifying the values that were present in the analyses. Based on this, we were then able to map, for example, the analysis environment to a specific `OperatingSystem` individual in the MAECO ontology.

Hashes are represented in the ontology as the `Hashes` class, which contains the values of individual hash functions as properties. In this case, the script must track all occurrences of hash values and assign the correct individual. If the script did not track this, duplicate individuals would be created.

More challenges need to be solved, such as extension of supported file types (MAECO currently lists only five subclasses of file types), mapping MITRE ATT&CK tactics and techniques into MAECO capabilities and behaviours, and some fine tuning of ranges of properties where other type might be more appropriate.

Acknowledgments. This research was funded by the EU NextGenerationEU through the Recovery and Resilience Plan for Slovakia under the project No. 09I05-03-V02-00064 and by the Slovak Research and Development Agency under the contract No. APVV-23-0292.

References

1. Adams, Z., Onoja, M., Kľuka, J., Homola, M., Štefan Balogh, Ploszek, R.: Maeco: Malware ontology framework towards enhancing explainable malware detection (2025), submitted to the 4th workshop on Application of Knowledge Methods in Information Security
2. Hybrid Analysis: Free Automated Malware Analysis Service - powered by Falcon Sandbox (2025), https://hybrid-analysis.com/
3. The MITRE Corporation: Malware Attribute Enumeration and Characterization (2022), https://maecproject.github.io/

Fair and Explainable Recommendations

Elena Stefancova[1 [0000−0001−8683−939X]] and Martin Homola[1 [0000−0001−6384−9771]]

Comenius University Bratislava, Bratislava, Slovakia
stefancova27@uniba.sk homola@fmph.uniba.sk

Abstract. Ensuring transparency and trustworthiness in artificial intelligence remains a critical challenge, particularly in complex, multi-stakeholder environments such as recommender systems. This work investigates how methods from knowledge representation can be applied to enhance fairness and explainability in recommendation pipelines. We investigate a hybrid approach that combines dynamic fairness-aware reranking mechanisms with latent factor–based synthetic data generation to systematically evaluate trade-offs between fairness and accuracy. Furthermore, we explore argumentation frameworks as a means of providing structured, context-aware explanations of recommendations, enabling users to understand how competing fairness objectives and stakeholder preferences are balanced. By integrating knowledge representation, explainability, and fairness into a unified framework, our research contributes to the development of transparent, accountable, and socially responsible intelligent systems, with broader applicability to domains where trust, reasoning, and knowledge sharing are essential.

Keywords: recommender systems · fairness · explainability · knowledge representation

1 Explainability in Recommender Systems

Explainability is a key requirement for trustworthy recommender systems, complementing accuracy with interpretability and transparency [6]. It helps stakeholders understand why items are recommended, detect potential biases, and build trust in the system [4].

Methods for explainability include inherently interpretable models and post-hoc techniques [3]. The former, such as knowledge-based approaches, make reasoning explicit, while the latter—e.g., attention mechanisms or LIME—approximate the logic of black-box models. Although useful, post-hoc methods may lack fidelity to the model's internal processes [1].

Explainability also concerns communication. Comparative justifications (e.g., why one item outranks another) align system outputs with human reasoning and improve user comprehension [7]. Argumentation frameworks extend this by structuring supporting and opposing reasons for recommendations, enabling interactive, context-aware, and multi-stakeholder explanations [5].

2 Towards Explainable Recommendations through Knowledge Representation

This project advances explainability by combining knowledge representation with comparative and argumentation-based explanations. Comparative methods clarify deviations introduced by fairness-aware re-ranking, while argumentation frameworks articulate how competing objectives and stakeholder preferences are balanced [2].

Planned user studies will assess the impact of these approaches on trust, transparency, and satisfaction. The long-term goal is a unified recommendation pipeline that integrates fairness and argumentation-based explanations, delivering outputs that are accurate, accountable, and socially responsible.

Acknowledgments. This research was funded by the EU NextGenerationEU through the Recovery and Resilience Plan for Slovakia under the project No. 09I05-03-V02-00064.

References

1. Musto, C., de Gemmis, M., Lops, P., Semeraro, G.: Generating post hoc review-based natural language justifications for recommender systems. User Modeling and User-Adapted Interaction **31**(3), 629–673 (jul 2021). https://doi.org/10.1007/s11257-020-09270-8, https://doi.org/10.1007/s11257-020-09270-8
2. Naveed, S., Donkers, T., Ziegler, J.: Argumentation-based explanations in recommender systems: Conceptual framework and empirical results. In: Adjunct Publication of the 26th Conference on User Modeling, Adaptation and Personalization. p. 293–298. UMAP '18, Association for Computing Machinery, New York, NY, USA (2018). https://doi.org/10.1145/3213586.3225240, https://doi.org/10.1145/3213586.3225240
3. Rudin, C.: Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. Nature machine intelligence **1**(5), 206–215 (2019)
4. Sinha, R., Swearingen, K.: The role of transparency in recommender systems. In: CHI '02 Extended Abstracts on Human Factors in Computing Systems. p. 830–831. CHI EA '02, Association for Computing Machinery, New York, NY, USA (2002). https://doi.org/10.1145/506443.506619, https://doi.org/10.1145/506443.506619
5. Vassiliades, A., Bassiliades, N., Patkos, T.: Argumentation and explainable artificial intelligence: a survey. The Knowledge Engineering Review **36**, e5 (2021). https://doi.org/10.1017/S0269888921000011
6. Wang, S., Zhang, X., Wang, Y., Ricci, F.: Trustworthy recommender systems. ACM Trans. Intell. Syst. Technol. **15**(4) (jul 2024). https://doi.org/10.1145/3627826, https://doi.org/10.1145/3627826
7. Yang, A., Wang, N., Cai, R., Deng, H., Wang, H.: Comparative explanations of recommendations. In: Proceedings of the ACM Web Conference 2022. pp. 3113–3123 (04 2022). https://doi.org/10.1145/3485447.3512031

Internet of Things – Cybersecurity Issues: Mini Review

Ivana Budinská[1] and Michaela Leľová[1,2]

[1] Institute of Informatics Slovak Academy of Sciences, Dúbravská cesta 9, Bratislava, 845 07, Slovakia
[2] Faculty of Electrotechnics and Informatics Technical University in Košice, Letná 9, Košice, 10587, Slovakia
ivana.budinska@savba.sk
michaela.lelova@savba.sk

Abstract. Smart devices have become an everyday part of people, companies and institutions. In addition to traditional devices such as smartphones, laptops, printers, there are devices that serve for entertainment and relaxation, and also devices that are connected to the operation of the household, cars, and last but not least, medical devices on which patients often depend for their lives. The cybersecurity of these devices is often underestimated. In this article, we will focus on an overview of known attacks, including specific real-life examples. In conclusion, we will outline some best practices and possible methods of protection.

Keywords: smart devices, cybersecurity, cyber attacks

1 Introduction

Real-world attacks on smart devices include large-scale DDoS attacks using compromised devices [1-2]. One of the most notorious attacks is the Mirai botnet, which took down major websites in 2016 [1]. Other attacks include gaining access to devices such as webcams to spy on users, manipulating devices for phishing campaigns, or exploiting vulnerabilities to disable smart home appliances, shut down systems, or even cause physical harm [1-2]. These attacks often exploit weak passwords, lack of security updates, and inadequate device security, demonstrating the significant privacy and security risks associated with the growing number of connected devices [1].

2 Examples of Attacks

The Mirai Bottnet scans the Internet for IoT devices running on the ARC processor [3-4]. This processor runs a stripped-down version of the Linux operating system. If the default username-and-password combo is not changed, Mirai is able to log into the device and infect it. Mirai has the potential to harness the collective power of millions of IoT devices into botnets, and launch attacks. In 2016 it launched DDoS taking down popular websites like Twitter, Netflix, and Reddit across the U.S. and Europe [1]. The

ThingBot - Phishing Campaigns discovered in 2020 manipulated over 100,000 smart home devices to send out a vast number of spam emails [5]. Hijacked Cameras and Baby Monitors: connected cameras and baby monitors can be used by hackers to spy on individuals and families. Footage from these devices, sometimes stored in poorly secured cloud environments, has been viewed online. A smart washing machine based in the Spinozacampus housing complex in Amsterdam was hacked to let students their clothes cleaned for free. Hackable cardiac devices were affected in an incident in 2017 when the FDA announced that they had discovered a serious vulnerability in implantable pacemakers made by St. Jude Medical.

3 Common Vulnerabilities Exploited

• **Weak Default Passwords**:
Many devices ship with weak or default passwords that are not changed by users, making them easy targets for hackers [6].

• **Lack of Software Updates:**
Some devices lack the ability to receive security updates, leaving them with unpatched vulnerabilities that can be exploited [6].

• **Insufficient Security by Design:**
Manufacturers often prioritize bringing products to market quickly, overlooking security as an added expense, leading to devices with limited built-in protection [6].

4 Consequences for users

• **Privacy Violations:**
Hackers can gain unauthorized access to personal data and sensitive information [7].

• **Financial Loss:**
In some cases, attacks could lead to financial repercussions for businesses and individuals [7].

• **Disruption of Services:**
Smart devices could be disabled or manipulated, disrupting daily routines [7].

• **Involvement in Larger Attacks:**
Hijacked smart devices can be used as part of a larger botnet to launch attacks against other targets on the internet [7].

5 Conclusion

The IoT promises to change our future, but at the same time, it poses severe security risks. Therefore, we should be aware and learn to protect our devices against cyber-attacks.

Acknowledgements. This work was supported by the Slovak Research and Development Agency under the Contract no. APVV-23-0292 and by the Slovak Scientific Grant Agency VEGA under the contract 2/0135/23 “Intelligent sensor systems and data processing”.

References

1. ANTONAKAKIS, Manos, et al. Understanding the mirai botnet. In: 26th USENIX security symposium (USENIX Security 17). 2017. p. 1093-1110.
2. VARDAKIS, George, et al. Review of smart-home security using the internet of things. Electronics, 2024, 13.16: 3343.
3. DE DONNO, Michele, et al. DDoS-capable IoT malwares: comparative analysis and Mirai investigation. Security and Communication Networks, 2018, 2018.1: 7178164.
4. KNOW, What Should You; BARANCHUK, Adrian; KRISHNAN, Kousik. Cybersecurity for Cardiac Implantable Electronic Devices. Journal of the American College of Cardiology, 2018, 71.11.
5. ABBAS, Syed Ghazanfar, et al. Identifying and mitigating phishing attack threats in IoT use cases using a threat modelling approach. Sensors, 2021, 21.14: 4816.
6. DEEP, Samundra, et al. A survey of security and privacy issues in the Internet of Things from the layered context. Transactions on Emerging Telecommunications Technologies, 2022, 33.6: e3935.
7. ZHOU, Wei, et al. The effect of IoT new features on security and privacy: New threats, existing solutions, and challenges yet to be solved. IEEE Internet of things Journal, 2018, 6.2: 1606-1616.

Secure Authentication for Mobile Applications using KeyCloak

Emil Gatial[1] and Zoltan Balogh[1]

[1]Institute of Informatics, Slovak Academy of Sciences, Bratislava, Slovakia
emil.gatial@savba.sk

Abstract. Secure authentication is essential for mobile applications handling sensitive data. This paper proposes a framework combining standards-based protocols (OAuth 2.0, OpenID Connect), a proxy service shielding administrative credentials, and mobile-side hardening. The approach enforces proof-of-control onboarding, short-lived tokens, and role-based access control. By integrating identity federation, credential minimization, and resilience measures, the model enhances security, scalability, and user trust in mobile environments.

Keywords: Authentication, KeyCloak, citizen science.

1 Introduction

The wide-spread use of mobile applications in domains such as healthcare, e-commerce, and citizen engagement has made secure user authentication a critical requirement. Mobile devices store sensitive personal information, enable financial transactions, and facilitate access to institutional resources, making them attractive targets for attackers. Designing authentication systems for mobile environments therefore requires balancing security, usability, and scalability, while ensuring compliance with privacy and data protection regulations.

2 Using KeyCloak for the User Authentication

This paper presents a security and authentication framework tailored for mobile applications, combining open-source identity management platforms, lightweight proxy services, and client-side hardening practices. At its foundation, the architecture employs standards-based identity protocols implemented in KeyCloak, such as OAuth 2.0 and OpenID Connect, which have become the de facto methods for secure token-based access and federated identity management in mobile ecosystems [2]. To minimize the exposure of administrative credentials, the system introduces an Admin Proxy service that mediates account creation, enforces email verification, and ensures that sensitive operations remain on the server side. This proxy pattern reduces the attack surface by preventing direct client access to high-privilege keys, while simultaneously supporting centralized monitoring and anomaly detection. Prior studies have highlighted the

weaknesses of password-only authentication and the growing role of multi-factor and biometric methods in improving resilience while maintaining usability [1]. Work on federated identity underscores the importance of interoperability across platforms [2], while comparative analyses of identity providers such as Keycloak, Auth0, and Okta reveal trade-offs between control, integration ease, and long-term dependency [3].

On the mobile device, authentication workflows are designed to enforce the principle of least knowledge and to increase user transparency. Registration processes collect minimal information (e.g., username, email, consent) and rely on server-side verification before account activation. The mobile client supports explicit login, logout, and password reset operations, while session metadata such as token expiration and scope can be visualized to provide users with a sense of control. By separating responsibilities, the model ensures that no single component holds excessive trust, thereby strengthening the system's resilience against compromise.

From a security perspective, robustness is ensured through: (i) credential minimization, removing long-lived client secrets; (ii) proof-of-control onboarding via email or equivalent verification; (iii) short-lived tokens with refresh rotation; (iv) role-based access control (RBAC) separating general and privileged users; and (v) event-driven monitoring of registration, verification, and token use.

3 Conclusion

In summary, this paper contributes a generalizable authentication architecture for mobile applications that integrates identity federation, proxy-based credential shielding, and mobile-side hardening into a coherent model. The approach addresses both technical threats such as credential theft and token replay, as well as governance challenges including user trust, privacy, and data integrity.

Acknowledgement

The publication is the result of these projects implementation: SILVANUS-SK (Grant No. 09I01-03-V04-00107), funded through the Recovery and Resilience Plan Mechanism and INFOTICK (Grant No. APVV-22-0372) Getting the right info on ticks.

References

1. Florêncio, D., & Herley, C. (2007). A Large-Scale Study of Web Password Habits. Proceedings of the 16th International Conference on World Wide Web, 657–666. DOI: 10.1145/1242572.1242661.
2. Podapati, V. H., Nigam, D., & Das, S. (2025). SoK: A Systematic Review of Context- and Behavior-Aware Adaptive Authentication in Mobile Environments. arXiv preprint arXiv:2507.21101.
3. Kuzminykh, L., Ghita, B., & Shiaeles, S. (2020) Comparative Analysis of Cryptographic Key Management Systems, Internet of Things, Smart Spaces, and Next Generation Networks and Systems, pp. 80-94.

photos by Júlia Pukancová, Emília Jókayová, and Martin Kenyeres
some images included in this book are for illustrative purposes only

ISBN 978-80-974468-2-6
EAN 9788097446826

designer of this book of abstracts:
Martin KENYERES