

Logically Explainable Malware Detection

Peter Anthony¹, Francesco Giannini^{2,*}, Michelangelo Diligenti³, Marco Gori³,
Martin Homola¹, Stefan Balogh⁴ and Ján Mojžiš⁵

¹Comenius University Bratislava, Slovakia

²Scuola Normale Superiore, Pisa, Italy

³University of Siena, Siena, Italy

⁴Slovak Technical University Bratislava, Slovakia

⁵Slovak Academy of Sciences Bratislava, Slovakia

Abstract

Malware detection is a challenging application due to the rapid evolution of attack techniques, and traditional signature-based approaches struggle with the high volume of malware samples. Machine learning approaches face such limitation, but lack a clear interpretability, whereas interpretable models often underperform. This paper proposes to use Logic Explained Networks (LENs), a recently proposed class of interpretable neural networks that provide explanations using First-Order Logic rules, for malware detection. Applied to the EMBER dataset, LENs show robustness superior to traditional interpretable methods and performance comparable to black-box models. Additionally, we introduce a tailored LEN version improving the fidelity of logic-based explanations.

Keywords

Malware Detection, Explainable AI, First-Order Logic, Logic Explained Networks

1. Introduction

Malware detection is crucial in cybersecurity due to the rapid evolution of attack techniques, and traditional signature-based methods from companies like Comodo, Kaspersky, and Symantec struggle to keep up with the millions of new malware samples each year [1, 2, 3]. Machine learning, particularly Deep Neural Networks (DNN), offers robust solutions by recognizing complex patterns, handling large datasets, and detecting zero-day attacks. However, these methods often lack explainability, limiting their trustworthiness in safety-critical applications.


Recently, Logic Explained Networks (LENs) [4] have been proposed as an explainable-by-design class of neural networks. LENs use human-understandable predicates and provide explanations through First-Order Logic (FOL) formulas, balancing accuracy and interpretability. Although LENs have shown success in various domains [5] and to different kind of data, such as images [6], textual information [7] and graphs [8], their effectiveness on large datasets like the EMBER malware dataset [9] (800,000 samples with thousands of features) remains unexplored.

HI-AI@KDD, Human-Interpretable AI Workshop at the KDD 2024, 26th of August 2024, Barcelona, Spain

*Corresponding author.

✉ peter.anthony@fmph.uniba.sk (P. Anthony); francesco.giannini@sns.it (F. Giannini);
michelangelo.diligenti@unisi.it (M. Diligenti); marco.gori@unisi.it (M. Gori); homola@fmph.uniba.sk (M. Homola);
stefan.balogh@stuba.ska (S. Balogh); jan.mojzis@savba.sk (J. Mojžiš)

 0000-0001-8492-8110 (F. Giannini)

 © 2024 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

This paper demonstrates that LENSs can form a robust malware detection framework with competitive performance against black-box models and superior to other interpretable methods. Additionally, we introduce an innovative approach to enhance the fidelity of LENSs' explanations, making them more accurate and meaningful.

This paper makes three main contributions: (i) it shows that Logic Explained Networks are effective for malware detection, providing meaningful explanations and predictive performance comparable to state-of-the-art black box models while outperforming other interpretable models, (ii) it introduces an improved rule extraction process for LENSs, enhancing scalability, fidelity, complexity, and predictive accuracy, and (iii) it offers an in-depth analysis of the extracted rules, evaluating their fidelity, complexity, and accuracy as input feature size increases.

2. Related Work

Machine Learning Approaches. Machine learning techniques are commonly used to train malware detectors and uncover complex patterns in malicious software [10, 11, 12]. While deep learning shows promising results due to its ability to learn from large datasets and generalize to unknown samples [13], several limitations remain, including low generalization performance for unseen malware samples. Moreover, classical machine learning models act as black-boxes, hindering explainability. In security-critical domains, interpretability is crucial for trust and legal compliance [14, 15, 16, 17]. Indeed, cybersecurity experts need insights into model decisions to enhance system trustworthiness.

Interpretable AI Models. Interpretable models (e.g., linear regression, decision trees) offer explanations, but struggle with complex features [18]. These methods prioritize interpretability over performance, unlike deep neural networks [19]. To address this trade-off, various techniques have been proposed. One such method is permutation feature importance, which interprets a wide range of machine learning models but comes with high computational costs [20]. Additionally, surrogate model methods like LIME [21] and SHAP [22] approximate the target model using interpretable models. However, their expressive ability may not match that of the complex target model, leading to inaccurate interpretations [23]. For the malware detection task, Švec et al. [24] explored interpretable concept learning algorithms all implemented in DL-Learner: OCEL (OWL Class Expression Learner), CELOE (Class Expression Learning for Ontology Engineering), PARCEL (Parallel Class Expression Learner), and SPARCEL (Symmetric Parallel Class Expression Learner). Their approach provided clear explanations but faced computational challenges and low performance.

3. Background on Logic Explained Networks

Logic Explained Networks [4] combine the advantages of black-boxes and transparent models by providing promptly interpretable neural networks in First-Order Logic (FOL). LENSs take human-understandable predicates as inputs, such as tabular data or concepts extracted from raw data, and express explanations in FOL rules involving these predicates. Thanks to their complex neural processing, LENSs achieve high-level performance while being easily interpretable.

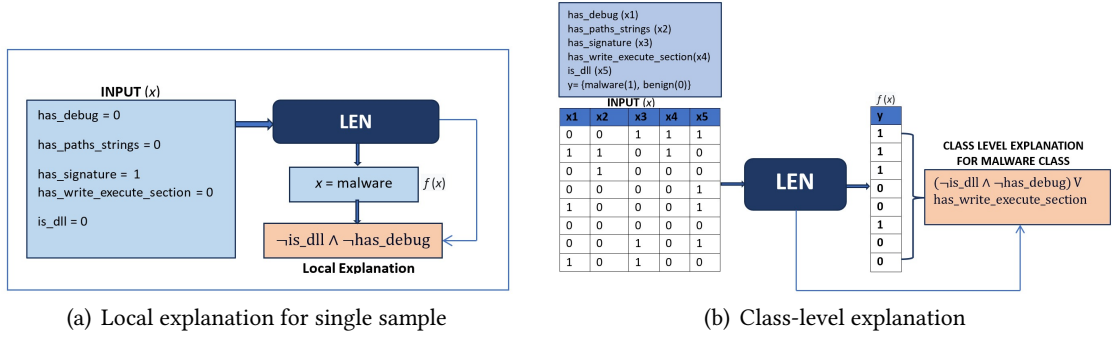


Figure 1: Illustration of LENS' local and class-level explanations for malware samples.

Formally, a LENS f can be defined as a map from $\mathcal{C} = [0, 1]^d$ -valued input concepts to $\mathcal{Y} = [0, 1]^r$ output classes, which can be used to directly classify samples and provide meaningful *local* and/or *global* explanations (cf. Figure 1).

As a special case, in malware detection we have $r = 2$ classes, i.e. $\{\text{malware}, \text{benign}\}$. In general, for each sample x , with $i \in \{1, \dots, r\}$, a prediction $f_i(x) = 1$ is *locally explained* by the conjunction of the most relevant input features $\phi_i^L(x) = \bigwedge_{j \in A(i)} (\neg)x_j$, where x_j is a logic predicate associated with the j -th input feature, and $A(i)$ is the set of relevant input features for the i -th task. Notice that each x_j can occur as a positive x_j or negative $\neg x_j$ literal, according to a given threshold (e.g. 0.5). The most representative local explanations can be aggregated to get a *global explanation* $\phi_i^G = \bigvee_{\phi_i^L(x) \in B(i)} \phi_i^L(x)$, where $B(i)$ collects the k -most frequent local explanations for the class i in the training set. We will refer later to such global explanations as **raw LENS explanations**. To prevent too complex global explanations, Gabriele et al. [4] suggests a top-(k) strategy, which focuses on aggregating only the most accurate local explanations. This method, we will refer to as **standard LENS explanations**, only includes local explanations that contribute to an improvement in validation accuracy, thus ensuring that the generalization of the rules is effective when applied to new data sets. However, even with this strategy, the complexity can remain high for large datasets with many samples and features.

4. Tailored-LENS' Explanations

Standard LENS explanations are more accessible than raw ones but still have some drawbacks, such as (i) determining the optimal k -value can be computationally intensive, (ii) selecting top- k local explanations based on their individual accuracy tends to select explanations that increase false positives rate, due to favoring high recall over precision, which is not preferable to construct a robust discrimination against malware. To enhance global LENS explanations for malware detection, this paper introduces what we call the **Tailored-LENS explanation** method, which uses a line search optimization to find the best threshold for choosing the right combination of local explanations, and removing terms from outlier samples to improve explanation quality. More specifically, we used a precision threshold to aggregate the local explanations, aiming to reduce false positives and avoid misrepresenting the model. Then,

Table 1

Comparison of LENs against black-box models. $LEN_{n,f}$ indicates that the model uses only n features.

Model	XAI	Accuracy	Precision	Recall	FP-Rate	F1-Score
LGBM[27]	No	0.9363	0.9244	0.9504	0.0605	0.9372
ANN/DNN[27]	No	0.95	0.96	0.94	0.0478	0.95
Improved DNN[28]	No	0.9404	0.9014	0.8885	0.1571	0.8866
FFN[29]	No	-	0.97	0.97	-	0.97
CNN[29]	No	-	0.95	0.95	-	0.95
MalConv w/ GCG[30]	No	0.9329	-	-	-	-
LEN_{10f}	Yes	0.8014	0.7766	0.8475	0.2449	0.8105
LEN_{100f}	Yes	0.92074	0.9124	0.9313	0.0898	0.9217
LEN_{1000f}	Yes	0.9232	0.9335	0.9117	0.0652	0.9224
LEN_{2000f}	Yes	0.9227	0.9257	0.9196	0.0742	0.9227
LEN_{all_f}	Yes	0.8695	0.8783	0.8568	0.1179	0.8674

an optimization process iteratively adjusts this threshold to find an optimal balance between precision and recall. The best solution is a simplified formula that improves validation accuracy, and only beneficial local explanations are included in the final Tailored-LEN global explanation. The details of this method can be found in Algorithm 1 in the Appendix.

5. Experiments

All the experiments carried out in this section are based on the EMBER dataset [9], a well-known dataset for malware detection with 800,000 labelled (400,000 benign and 400,000 malicious) and 300,000 unlabelled samples, respectively. For our experimental analysis we utilized the version with derived features, as defined by Mojžiš and Kenyeres [25], which represent a variation of the ontology realized by Švec et al. [26]. More details about the experimental settings can be found in Appendix A.1. The experiments aim to demonstrate that: (i) LENs perform similarly to black-box models while offering explanations; (ii) LENs surpass previously used interpretable machine learning models; (iii) Tailored-LENs explanations provide a better trade-off in terms of complexity vs accuracy vs fidelity wrt standard and raw LEN explanations.

Comparison against black-box models. We compare LENs with black-box models using the full EMBER dataset, with 600k samples allocated for training and 200k samples for testing. We also evaluated the performances of LENs varying different subsets' size of the most informative features, ranging from 10 to 2000 features, to observe the impact on the model's results.

Results. Table 1 shows that LENs have high performances in malware detection, achieving an accuracy of at least 92.07% and an F1-score of 92.17% with a minimum of 100 features. The performances are slightly lower with the full feature set, possibly due to the feature binarization process, which increases the feature dimensionality and potentially introduces noise. Despite this, LENs closely match the best deep-learning black-box models, with less than a 5% difference in most metrics. Notably, LENs are competitive even with a small feature set and maintain high generalization capabilities with larger feature sizes. In addition, the key advantage of LENs over

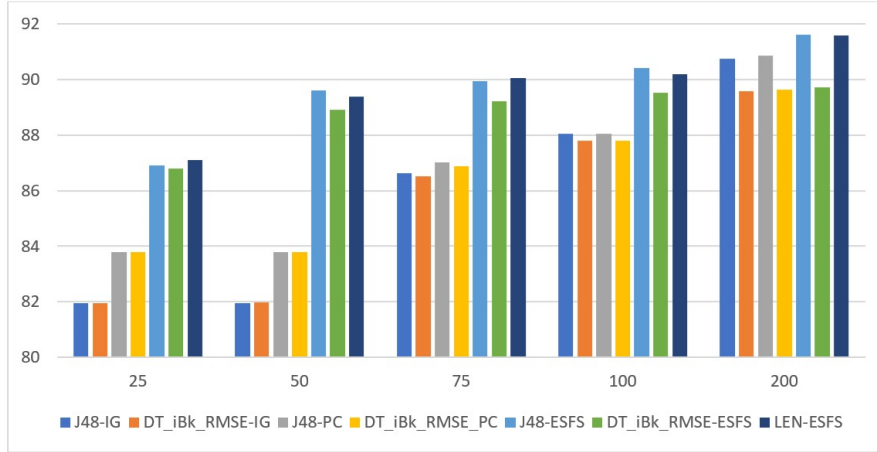


Figure 2: Accuracy, highest is the best. LEN vs 3 decision tree models, five different feature sizes counts.

black-box models is their interpretability, which provides insights into the decision-making process.

Table 2

Performance comparison of LEN against explainable approaches.

Model	Comparison with concept learning approaches				
	Accuracy	Precision	Recall	FP rate	F1
PARCEL(0/√/X/20)[24]	0.68 ± 0.01	0.80 ± 0.02	0.49 ± 0.03	0.12 ± 0.01	0.60 ± 0.03
PARCEL(0/X/X/5) [24]	0.62 ± 0.04	0.90 ± 0.02	0.29 ± 0.09	0.03 ± 0.00	0.43 ± 0.12
PARCEL(1/√/X/10)[24]	0.72 ± 0.01	0.71 ± 0.01	0.72 ± 0.02	0.28 ± 0.01	0.72 ± 0.01
PARCEL(1/X/X/5)[24]	0.70 ± 0.02	0.81 ± 0.01	0.52 ± 0.04	0.12 ± 0.00	0.63 ± 0.04
SPARCEL(1/√/X/20)[24]	0.72 ± 0.01	0.72 ± 0.00	0.73 ± 0.02	0.27 ± 0.00	0.72 ± 0.01
SPARCEL(1/X/X/5)[24]	0.64 ± 0.03	0.88 ± 0.04	0.33 ± 0.06	0.04 ± 0.01	0.48 ± 0.08
OCEL(25/√/√/5)[24]	0.69 ± 0.01	0.68 ± 0.05	0.74 ± 0.10	0.35 ± 0.12	0.70 ± 0.02
CELOE(25/√/√/5)[24]	0.68 ± 0.01	0.65 ± 0.03	0.77 ± 0.05	0.40 ± 0.07	0.70 ± 0.01
LEN	0.87 ± 0.01	0.88 ± 0.02	0.87 ± 0.02	0.13 ± 0.02	0.88 ± 0.01

Comparison against interpretable approaches. We identified two significant contributions for interpretable malware detection using the EMBER dataset: (i) the concept learning method by Švec et al. [24], and (ii) the decision-tree-based techniques by Mojžiš and Kenyeres [25]. The former approach, due to its complexity, was tested on 5,000 random samples with 5-fold cross-validation. The latter utilized a dataset of 600,000 samples with an 80%/20% training/testing for 25, 50, 75, 100 and 200 feature sets. To ensure consistency, the same sample size and cross-validation method were used for the concept learning approach, and the same feature set and sample distribution were applied for the decision-tree-based approach.

Results. LENs significantly outperform concept learning approaches in all metrics (cf. Table 2). This supports the claim that LENs can represent a promising solution for real-world deployment, meeting the increasing need for both clarity and effectiveness in malware detection systems. Additionally, when compared with standard decision tree models, LENs demonstrate good performance wrt different feature sizes, as evidenced in Figure 2. While LENs outperform most

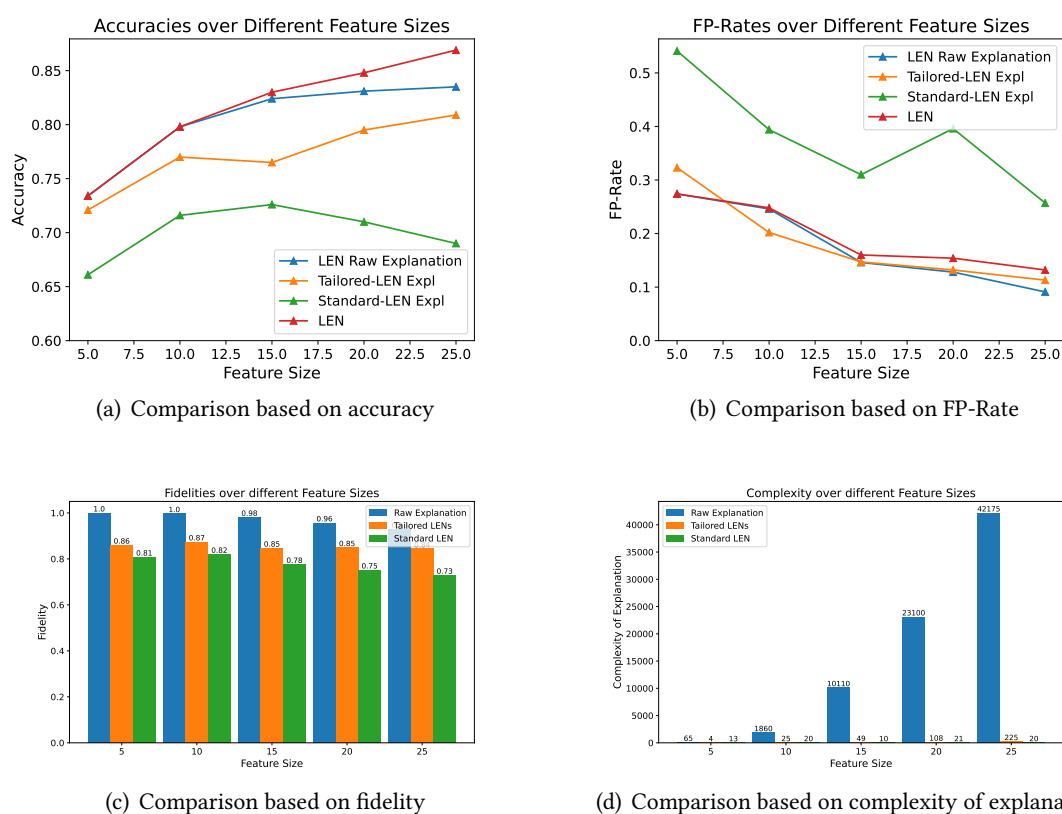


Figure 3: Plots comparing the performance of the explanations of the different LEXs in terms of (a) accuracy, (b) FP-Rate, (c) Fidelity and (d) Complexity, over different feature size on the EMBER dataset.

decision tree models, their performance is on par with the J48-ESFS model.

Analysis of provided Explanations. The Tailored-LEN explanation method (Section 4) was evaluated against the Raw-LEN and Standard-LEN methods proposed in the original paper [4]. This evaluation used 25,000 samples and tested feature sets of 5, 10, 15, 20, and 25 features. with the data divided into a 75% and a 25% split for training and testing, respectively.

Results. Figure 3 illustrates that Tailored LEN explanations outperform Standard LEXs across all feature sizes, offering better fidelity and lower complexity. Raw LEXs provide a better predictive performance but suffer from high complexity, making them less interpretable. Standard-LEXs, while similar in complexity to Tailored-LEXs, fall short in both fidelity and accuracy compared to Tailored-LEXs. Additionally, the practicality of the extracted rules was analyzed in the context of malware detection applications by a cybersecurity expert (cf. Table 3 in Appendix).

6. Conclusions

This paper studies the application of Logic Explained Networks to malware detection. The conducted experiments demonstrate that LENSs can attain performance comparable to complex black-box neural models, while maintaining explainability and outperforming other interpretable machine learning alternatives in terms of efficacy.

Furthermore, this study introduces a novel algorithm designed to extract global explanations from LENSs. This algorithm improves the predictive precision of LENSs, while yielding explanations characterized by both elevated fidelity and reduced complexity. The findings support the claim that LENSs are a promising candidate for the integration of explainable methodologies into malware detectors.

Acknowledgments

This work was supported by the TAILOR Collaboration Exchange Fund (CEF) under the European Union's Horizon 2020 research and innovation program GA No 952215. This work has been also supported by the Partnership Extended PE00000013 - "FAIR - Future Artificial Intelligence Research" - Spoke 1 "Human-centered AI". The work was also sponsored by the Slovak Republic under the grant no. APVV-19-0220 (ORBIS).

References

- [1] Y. Ye, T. Li, D. Adjeroh, S. Iyengar, A survey on malware detection using data mining techniques, *ACM Computing Surveys* 50 (2017) 1–40. doi:10.1145/3073559.
- [2] H. R. Borojerdi, M. Abadi, Malhunter: Automatic generation of multiple behavioral signatures for polymorphic malware detection, in: *ICCKE 2013, IEEE*, 2013, pp. 430–436.
- [3] D. Venugopal, G. Hu, Efficient signature based malware detection on mobile devices, *Mobile Information Systems* 4 (2008) 33–49.
- [4] G. Ciravegna, P. Barbiero, F. Giannini, M. Gori, P. Liò, M. Maggini, S. Melacci, Logic explained networks, *Artificial Intelligence* 314 (2023) 103822. URL: <https://www.sciencedirect.com/science/article/pii/S000437022200162X>. doi:<https://doi.org/10.1016/j.artint.2022.103822>.
- [5] G. Ciravegna, F. Giannini, P. Barbiero, M. Gori, P. Lio, M. Maggini, S. Melacci, Learning logic explanations by neural networks, in: *Compendium of Neurosymbolic Artificial Intelligence*, IOS Press, 2023, pp. 547–558.
- [6] P. Barbiero, G. Ciravegna, F. Giannini, P. Lió, M. Gori, S. Melacci, Entropy-based logic explanations of neural networks, in: *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, 2022, pp. 6046–6054.
- [7] R. Jain, G. Ciravegna, P. Barbiero, F. Giannini, D. Buffelli, P. Lio, Extending logic explained networks to text classification, in: *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, 2022, pp. 8838–8857.
- [8] S. Azzolin, A. Longa, P. Barbiero, P. Lio, A. Passerini, et al., Global explainability of gnn's via logic combination of learned concepts, in: *ICLR 2023*, online, 2023, pp. 1–19.

- [9] H. S. Anderson, P. Roth, Ember: an open dataset for training static pe malware machine learning models, arXiv preprint arXiv:1804.04637 (2018).
- [10] S. Millar, N. McLaughlin, J. Martinez del Rincon, P. Miller, Z. Zhao, Dandroid: A multi-view discriminative adversarial network for obfuscated android malware detection, in: Proceedings of the tenth ACM conference on data and application security and privacy, 2020, pp. 353–364.
- [11] S. Millar, N. McLaughlin, J. M. del Rincon, P. Miller, Multi-view deep learning for zero-day android malware detection, Journal of Information Security and Applications 58 (2021) 102718. URL: <https://www.sciencedirect.com/science/article/pii/S2214212620308577>. doi:<https://doi.org/10.1016/j.jisa.2020.102718>.
- [12] Y. Ye, L. Chen, S. Hou, W. Hardy, X. Li, Deepam: a heterogeneous deep learning framework for intelligent malware detection, Knowledge and Information Systems 54 (2018) 265–285.
- [13] P. S. H. K. J. Hirpara, Exploring the diverse applications of deep learning across multiple domains, Recent Research Reviews Journal 4 (2023) 100692. URL: <https://irojournals.com/rrrj/articles/view/2/1/16>. doi:<http://dx.doi.org/10.36548/rrrj.2023.1.16>.
- [14] G. Iadarola, F. Martinelli, F. Mercaldo, A. Santone, Towards an interpretable deep learning model for mobile malware detection and family identification, Computers & Security 105 (2021) 102198. URL: <https://www.sciencedirect.com/science/article/pii/S0167404821000225>. doi:<https://doi.org/10.1016/j.cose.2021.102198>.
- [15] A. Mills, T. Spyridopoulos, P. Legg, Efficient and interpretable real-time malware detection using random-forest, in: 2019 International conference on cyber situational awareness, data analytics and assessment (Cyber SA), IEEE, 2019, pp. 1–8.
- [16] B. Marais, T. Quertier, C. Chesneau, Malware analysis with artificial intelligence and a particular attention on results interpretability, in: Distributed Computing and Artificial Intelligence, Volume 1: 18th International Conference 18, Springer, 2022, pp. 43–55.
- [17] J. Dolejš, M. Jureček, Interpretability of machine learning-based results of malware detection using a set of rules, in: Artificial Intelligence for Cybersecurity, Springer, 2022, pp. 107–136.
- [18] A. Orlenko, J. H. Moore, A comparison of methods for interpreting random forest models of genetic association in the presence of non-additive interactions, BioData Mining 14 (2021) 9. URL: <https://doi.org/10.1186/s13040-021-00243-0>. doi:10.1186/s13040-021-00243-0.
- [19] H. Xu, X. Zhang, H. Li, G. Xiang, An ensemble of adaptive surrogate models based on local error expectations, Mathematical Problems in Engineering 2021 (2021) 8857417. URL: <https://doi.org/10.1155/2021/8857417>. doi:10.1155/2021/8857417.
- [20] F. Fumagalli, M. Muschalik, E. Hüllermeier, B. Hammer, Incremental permutation feature importance (ipfi): towards online explanations on data streams, Machine Learning 112 (2023) 4863–4903. URL: <http://dx.doi.org/10.1007/s10994-023-06385-y>. doi:10.1007/s10994-023-06385-y.
- [21] M. T. Ribeiro, S. Singh, C. Guestrin, " 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, pp. 1135–1144.
- [22] S. M. Lundberg, S.-I. Lee, A unified approach to interpreting model predictions, in: I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, R. Garnett (Eds.), Advances in Neural Information Processing Systems

- 30, Curran Associates, Inc., 2017, pp. 4765–4774. URL: <http://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions.pdf>.
- [23] S. Nembrini, I. R. König, M. N. Wright, The revival of the gini importance?, *Bioinformatics* 34 (2018) 3711–3718.
- [24] P. Švec, Š. Balogh, M. Homola, J. Kl'uka, T. Bisták, Semantic data representation for explainable windows malware detection models, *arXiv preprint arXiv:2403.11669* (2024).
- [25] J. Mojžiš, M. Kenyeres, Interpretable rules with a simplified data representation - a case study with the ember dataset, in: R. Silhavy, P. Silhavy (Eds.), *Data Analytics in System Engineering*, Springer International Publishing, Cham, 2024, pp. 1–10.
- [26] P. Svec, S. Balogh, M. Homola, J. Kluka, Knowledge-based dataset for training PE malware detection models, *CoRR abs/2301.00153* (2023). URL: <https://doi.org/10.48550/arXiv.2301.00153>. doi:10.48550/ARXIV.2301.00153.
- [27] C. Connors, D. Sarkar, Machine learning for detecting malware in pe files, in: *2023 International Conference on Machine Learning and Applications (ICMLA)*, IEEE, 2023, pp. 2194–2199.
- [28] S. Lad, A. Adamuthe, Improved deep learning model for static pe files malware detection and classification, *International Journal of Computer Network and Information Security* 14 (2022) 14–26. doi:10.5815/ijcnis.2022.02.02.
- [29] S. Pramanik, H. Teja, Ember - analysis of malware dataset using convolutional neural networks, in: *2019 Third International Conference on Inventive Systems and Control (ICISC)*, 2019, pp. 286–291. doi:10.1109/ICISC44355.2019.9036424.
- [30] E. Raff, W. Fleshman, R. Zak, H. S. Anderson, B. Filar, M. McLean, Classifying sequences of extreme length with constant memory applied to malware detection, 2020. *arXiv:2012.09390*.
- [31] H. Liu, M. Cocea, W. Ding, Decision tree learning based feature evaluation and selection for image classification, in: *2017 International Conference on Machine Learning and Cybernetics (ICMLC)*, volume 2, 2017, pp. 569–574. doi:10.1109/ICMLC.2017.8108975.
- [32] J. An, P. Kilmartin, B. Young, R. Deed, W. Yu, Decision trees as feature selection methods to characterize the novice panel's perception of pinot noir wines, 2023. URL: <https://doi.org/10.21203/rs.3.rs-2650497/v1>. doi:10.21203/rs.3.rs-2650497/v1.
- [33] A. Papenmeier, G. Englebienne, C. Seifert, How model accuracy and explanation fidelity influence user trust, 2019. *arXiv:1907.12652*.

A. Appendix

A.1. Experimental Setting and Details

Dataset and Pre-processing. The Elastic Malware Benchmark for Empowering Researchers (EMBER), released in 2018 [9], is a well-known dataset of malware samples. The EMBER dataset is the main dataset used throughout this study and provides a comprehensive collection of features extracted from Windows Portable Executable (PE) files. The dataset comprises both benign and malicious samples. It contains features from 1.1 million PE files with diverse attack types, of which 800,000 are labelled samples (400,000 benign and 400,000 malicious), and

300,000 are unlabelled samples. We harnessed only the labelled samples for our study.

While the EMBER dataset is in JSON format, for our experimental analysis we utilized the version with derived features, as defined by Mojžiš and Kenyeres [25]. This dataset consists of binary features (each feature can be either true or false – i.e. it is boolean, denoting the presence/positiveness or the absence/negativeness of each feature) to create a simplified representation. This representation is actually a variation of the ontology realized over the EMBER dataset by Švec et al. [26].

Feature Selections Methods. Since the majority of interpretable methods have severe performance limitations and also fail at providing human-readable explanations when a large feature set is available, we compared our approach against other interpretable models using the same feature selection techniques used in the original papers. In particular, following what was done in [31, 32], a decision tree-based feature selection technique was used in the experiments to identify and retain the most informative features. In the experiments we considered a varying amount of the most informative features. On the other hand, for the comparison of LENs with state-of-the-art black-box models, like Deep Neural Networks (DNN), we used the full of set of features.

Evaluation Metrics. We evaluated both the LEN model and explanations performance using standard metrics, i.e. accuracy, precision, recall, False Positive Rate and F1-score. For evaluating the explanations performance, two additional metrics were employed: Fidelity [4, 33] and Complexity [4].

The *Fidelity* metric measures the extent to which explanations faithfully represent the inner workings of predictive models. Formally, given a data collection, a predictive model (Model_{PM}), and a model explanation (Model_{Ex}), the $\text{Fidelity}(\text{Model}_{\text{Ex}})$ is defined as the accuracy obtained when comparing the predictions made by Model_{PM} and the predictions derived from the explanations Model_{Ex} :

$$\text{Fidelity} = \frac{1}{N} \sum_{i=1}^N \text{Acc}(\text{Model}_{\text{PM}}(x_i), \text{Model}_{\text{Ex}}(x_i)) \quad (1)$$

where N is the number of samples in the data collection, Acc denotes the accuracy metric, and $\text{Model}_{\text{PM}}(x_i)$ and $\text{Model}_{\text{Ex}}(x_i)$ represent the predictions made by Model_{PM} and Model_{Ex} , respectively, for the i -th sample x_i . This fidelity metric will serve as a crucial indicator of the trustworthiness of the explanations extracted.

The *Complexity* metric counts the number of terms in the explanation as a proxy for the human understandability of the explanation.

A.2. Global Explanations in Tailored LENs

Algorithm 1 reports the full procedure to get the global explanations for Tailored LENs.

Algorithm 1: Compilation of the global explanations in Tailored LENSs.

Input: $local_expl, LEN_Precision, \alpha$
 $Threshold_{current} \leftarrow LEN_Precision$
 $best_expl \leftarrow FilterExpl(local_expl, Threshold_{current})$
 $Accuracy_{best} \leftarrow EvaluateAcc(best_expl)$

while not reached optimal accuracy **do**
 $Threshold_{current} \leftarrow Threshold_{current} - \alpha$
 $remaining_expl \leftarrow FilterExpl(local_expl, Threshold_{current})$
 $Accuracy_{current} \leftarrow EvaluateAcc(remaining_expl)$
 if $Accuracy_{current} > Accuracy_{best}$ **then**
 $best_expl \leftarrow remaining_expl$
 $Accuracy_{best} \leftarrow Accuracy_{current}$
 else
 break

$Threshold_{current} \leftarrow LEN_Precision$
while not reached optimal accuracy **do**
 $Threshold_{current} \leftarrow Threshold_{current} + \alpha$
 $remaining_expl \leftarrow FilterExpl(local_expl, Threshold_{current})$
 $Accuracy_{current} \leftarrow EvaluateAcc(remaining_expl)$
 if $Accuracy_{current} > Accuracy_{best}$ **then**
 $best_expl \leftarrow remaining_expl$
 $Accuracy_{best} \leftarrow Accuracy_{current}$
 else
 break // Optimal accuracy reached

A.3. Human-expert remarks on logic explanations provided by LENSs.

Table 3 shows some local explanations provided by LENSs together with some remarks provided by a cybersecurity expert. The expert highlighted that all the explanations indicated meaningful reasons for the sample being a malware, and that it is impressive to be able to have this level of insight into the workings of an ML-based model that was able to process the full EMBER dataset. At the same time, all explanations were more general and abstract compared to those derived by concept learning on a fractional dataset [24].

Table 3

On the left we have some examples of local explanations for a detected malware. On the right side it is reported a justification of the plausibility of the rule according to domain expertise in the field of malware detection.

Explanation	Cybersecurity expert remarks
<code>has_section_high_entropy ^ ¬is_dll ^ ¬has_debug</code>	It points to packed malware that is not .dll and has not debug symbols enabled (which is a typical malware behaviour).
<code>has_write_execute_section ^ ¬has_debug</code>	Malware can typically use a section with write and execute permission for self injection.
<code>has_section_high_entropy ^ ¬has_signature</code>	This also points to packed malware and malware usually has no signature
<code>has_section_high_entropy ^ sect_text_write_execute_section</code>	It points to packed (encrypted) code