Striving for Simplicity in Deep Neural Models Trained for Malware Detection

Malik AL-Essa[1,2][0000−0002−0892−975X], Giuseppina Andresini[1,2][0000−0002−5272−644X], Annalisa Appice[1,2][0000−0001−9840−844X], and Donato Malerba[1,2][0000−0001−8432−4608]

[1] Department of Computer Science, University of Bari "Aldo Moro", Italy
{malik.alessa,giuseppina.andresini,annalisa.appice,
donato.malerba}@uniba.it
[2] CINI - Consorzio Interuniversitario Nazionale per l'Informatica

Abstract. A multitude of recent studies have repeatedly shown the accuracy of deep neural models in several malware detection problems. Although deep learning has recently achieved amazing results in cybersecurity, deep neural models remain complex models, which often produce non-transparent decisions, and which are vulnerable to adversarial attacks. Hence, the evaluation of a deep neural model in cybersecurity should include the analysis of the simplicity and vulnerability of the model, in addition to its accuracy. In this study, we investigate how XAI can disclose useful information concerning the robustness of the input characteristics in deep neural models and how this knowledge can be used in malware detection problems to pursue simpler deep neural models that are still accurate, as well as to fool deep neural models. In particular, AI defenders are interested in identifying the minimum amount of input characteristics to train a simple deep neural model by preserving high accuracy. AI attackers are interested in identifying the minimum amount of input characteristics to perturb, in order to evade deep neural models. We explore how simplicity can be realized in malware detection problems by accounting for explanations of input characteristics, which are produced with either a global XAI technique or a Mutual Information analysis.

Keywords: XAI · Feature Ranking · Deep Learning · Adversarial Learning · Malware Detection · Model Simplicity

1 Introduction

In the digital age the use of deep learning is one of the most powerful AI paradigms for cybersecurity. Despite the amazing results recently achieved with deep learning techniques in securing the digital infrastructures of modern organizations, the security of deep neural models can easily be jeopardized by adversarial attacks. Adversarial learning [16] is the area of study that focuses on identifying the vulnerabilities in artificial systems (comprising deep neural models). An adversary injects a slight perturbation into the input sample to increase

the misclassification rate of the models. Such tainted samples are known as adversarial samples. Several techniques to generate adversarial samples have been described in the recent adversarial learning literature [7]. On the other hand, in the game process of adversarial attacks and defense technologies, researchers have also begun to pay attention to the research in the field of adversarial defense (e.g., adversarial training or defensive distillation) [7].

eXplainable Artificial Intelligence, or XAI, is the area of study that aims to enable humans to understand the decisions of artificial systems by producing more explainable models while maintaining a good level of predictive accuracy. Although deep learning techniques allow us to learn accurate classification models in several cybersecurity problems, these are commonly opaque models, that are difficult to explain. On the other hand, easier-to-explain models are becoming increasingly desirable in cybersecurity applications, to increase the stakeholders' confidence. In addition, simplicity is considered an important characteristic in cybersecurity problems since over-engineered deep neural models tend to increase the likelihood of overfitting, decrease detection efficiency, and lowering the explainability of the model's output.

In this study, we explore how an XAI technique can help in achieving simpler deep neural models by preserving, or even increasing, their accuracy. For this purpose, we consider DALEX [8] that is a post-hoc, global XAI technique. DALEX is used to provide measurable factors on which characteristics of the input space influence the prediction of a cyber-attack and to what extent on a deep neural model's decisions. We use these explanations to identify the subspace of input characteristics to train an accurate deep neural model for malware detection. In addition, we analyse the performance of Mutual Information [19] as a decision model-agnostic alternative to explain the relevance of input dimensions on an observed output. Finally, we analyse the performance of both DALEX and Mutual Information for achieving simplicity also in the attacker perspective, that is, to identify the subset of input dimensions to perturb, in order to fool a deep neural model. To perturb data, we use FGSM [12], that is one of the most popular adversarial sample generators. The experimental study is performed by considering two, multi-class, malware detection problems.

The paper is organized as follows. The related work is presented in Section 2. The adopted methods are described in Section 3, while the experimental setup and the results are discussed in Section 4. Finally conclusions are drawn in Section 5

2 Related work

With the boom of deep learning in cybersecurity, most of the recent research in cybersecurity has been developed with the priority of providing accurate classifications of cyber data [18]. On the other hand, adversarial learning has recently gained growing attention also in the cybersecurity field by identifying potential vulnerabilities of deep learning algorithms during learning and classification, devising the corresponding attacks and evaluating their impact on the artificial

systems, as well as proposing countermeasures to improve the security of deep neural models against the considered attacks [7].

On the other hand, significant interest in the cybersecurity research community has recently been observed in the development of both post-hoc explanations, in which an XAI technique can be applied to already trained deep neural models [20, 5] and intrinsic explanations, in which an XAI technique can be used to allow the deep neural model to see the most important information during the learning stage [3]. An in-depth examination of XAI studies in cyber-security has been conducted in [9]. A taxonomy of XAI techniques used in cybersecurity problems, as well as a novel black-box attack, have been proposed in [14], to explore consistency, correctness and confidence security properties of gradient-based XAI techniques.

Finally, a few recent cybersecurity studies have also started the investigation of XAI in adversarial learning for both offensive [15] and defensive purposes [1]. Finally, in [2], XAI is used for selecting the dimensions of the input space to gain accuracy with adversarial training. On the other hand, this idea of feature selection is well-known in cybersecurity. For example, feature selection is used as a crucial step to learn deep neural models for network intrusion detection in [4, 13].

3 Methods

Let us consider a dataset $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$ of N samples, where $\mathbf{x} \in \mathbf{X} \subseteq \mathbb{R}^d$ is a d-dimensional space of input characteristics that describe both normal and malicious samples (e.g., malware apps), whereas $y \in Y$ is the value of the target variable Y. The target variable may assume K distinct classes: either the class *normal* or one of the $K-1$ distinct classes of a *malware* behaviour, depending on those historically detected and labeled. The machine learning steps performed in this study include: (1) Training a deep neural model – DNN – of the multi-class classification function $\mathbf{X} \mapsto Y$. (2) The use of DALEX and Mutual Information to measure the effect of input characteristics on target values. (3) The use of FGSM to generate adversarial samples. We divide $\mathcal{D}$ into training set and testing set. We train a DNN from the training set. We use either DALEX or Mutual Information, to measure the importance of the input characteristics in both the training set and the testing set. We use the information collected on the importance of the input characteristics in the training set, to select the subset of input characteristics for training a simpler, still accurate DNN. The accuracy performance of the new DNN is evaluated on the testing set. In this case, our intuition is that the less relevant characteristics may cause overfitting and lack of generality. So, the removal of the less relevant characteristics from the input space can foster the training of a simpler DNN that may also achieve higher accuracy on unseen data. On the other hand, we use the information on the importance of input characteristics in the testing set to select the subset of input characteristics that may become the target of an attacker for the adversarial sample generation. In this case, we would understand how measurements of

input characteristic relevance can reveal model vulnerabilities that can become the target of attackers.

3.1 DNN

We learn a multi-class deep neural model through a DNN architecture that consists of three fully connected layers, one dropout layer and one batch normalization layer, to mitigate the overfitting risk. The output probabilities are obtained using the softmax activation function in the last layer. The Rectified Linear Unit (ReLU) activation function is used in all the other hidden layers. This DNN architecture has been recently used in cybersecurity problems [1, 2].

3.2 DALEX and Mutual Information

DALEX [8] is a post-hoc, XAI framework that implements techniques for understanding both the global and local structure of predictive black-box models. In this study, we integrate the global, post-hoc explanation methodology that allows us to explain the behavior of the DNN by measuring the global relevance of different input characteristics on DNN decisions. DALEX uses a permutation-based variable-importance measurement to quantify the relevance of each characteristic on the decisions of a model [11]. For each characteristic of the input space, its effect is removed by permuting the values of the characteristic and the loss function compares the performance before and after. Intuitively, if a characteristic is important, randomly permuting its values will cause the loss to increase. The Mutual Information [19] is a data-driven, decision model-agnostic measurement that quantifies the amount of information obtained about the target Y through observing the input characteristic X, that is, $MI(X,Y) = \sum_{y}\sum_{x} \frac{p(x,y)}{p_1(x)p_2(y)}$, where $p_1(x)$ and $p_2(y)$ are the marginal distribution probabilities of x and y, respectively. The Mutual Information is commonly used for feature scoring in classification problems without using any classification model. The higher the Mutual Information, the more important the input characteristic.

3.3 FGSM

FGSM (Fast Gradient Sign Method) [12] is one of the most popular adversarial sample generators that is prone to catastrophic overfitting [6]. This is a white-box gradient-based method that finds the loss (e.g., the cross-entropy) to apply to an input sample, to make the decisions of the DNN less robust for a specific class. FGSM is based on the gradient formula $g(\mathbf{x}) = \nabla_{\mathbf{x}} J(\theta, \mathbf{x}, y)$, where $\nabla_{\mathbf{x}}$ represents the gradient computed with respect to $\mathbf{x}$, and $J(\theta, \mathbf{x}, y)$ is the loss function of the DNN. Specifically, FGSM identifies the minimum perturbation ϵ to add to a training sample $\mathbf{x}$ to create an adversarial sample, in order to maximize $J()$. Therefore, given ϵ, for each $(\mathbf{x}, y) \in \mathcal{D}$, a new sample $(\mathbf{x^{adv}}, y) \in \mathcal{A}$ can be generated so that $\mathbf{x^{adv}} = \mathbf{x} + \epsilon sign(g(\mathbf{x}))$.

Table 1: Data set description

Dataset	#Training set	#Testing set	#Input characteristics	#Classes
CICMalDroid20	8118	3480	40	5
CICMalMem22	41017	17579	55	4

4 Empirical Evaluation

We performed an experimental study with two malware detection datasets. The datasets are described in Section 4.1. The implementation details of the proposed method are reported in Section 4.2. The results are illustrated in Section 4.3.

4.1 Datasets

Two datasets were considered in the evaluation study: an android malware dataset, namely CICMalDroid20 and a malware memory analysis dataset, namely CICMalMem22. A description of characteristics of both datasets is reported in Table 1. In particular, CICMaldroid20 dataset [17] includes samples of Android apps collected from December 2017 to December 2018 from different sources including VirusTotal service, Contagio security blog, AMD, MalDozer. It comprises apps labeled in five distinct classes: Adware, Banking malware, SMS malware, Riskware, and Normal. Each app is described by forty input characteristics that represent the top-40 static and dynamic attributes extracted using CopperDroid. Static characteristics mainly describe intents, permissions and services, frequency counts for different file types, incidents of obfuscation and sensitive API invocations. Dynamic characteristics describe behaviours broken down into three categories of system calls, binder calls and composite behaviors. CICMalMem22 dataset [10] is an obfuscated malware dataset that was created to evaluate obfuscated malware detection methods. VolMemLyzer was used to extract fifty-five malware characteristics from the memory dump. These characteristics are classified into five categories: Malfind characteristics that allow the identification of potential malicious executables; Ldrmodule that allow the identification of injected code into the system; Handle characteristics that allow the analysis of the type of information stored in memory; Process view characteristics that provide information of the list of processes; API-hook characteristics that count the number of the most important API-hooks performed.This dataset is made up by 50% malicious memory dumps and 50% benign memory dumps. Malicious dumps belong to three malware classes: Trojan Horse, Spyware and Ransomware. For each dataset, we adopted a stratified division of the datasets into a training set (70%) and a testing set (30%).

4.2 Implementation Details

The code used in this experimental study was implemented in Python 3.9 with Keras 2.7. [3] For each dataset, the hyper-parameters of the deep neural models

[3] The source code is available at https://github.com/malikalessa/NFMCP.

Table 2: Hyper-parameter search space

Hyper-parameter	Search space
Mini-batch size	$\{2^5, 2^6, 2^7, 2^8, 2^9\}$
Learning rate	[0.0001, 0.001]
Dropout	[0,1]
# of neurons per hidden layer	$\{2^5, 2^6, 2^7, 2^8, 2^9, 2^{10}\}$

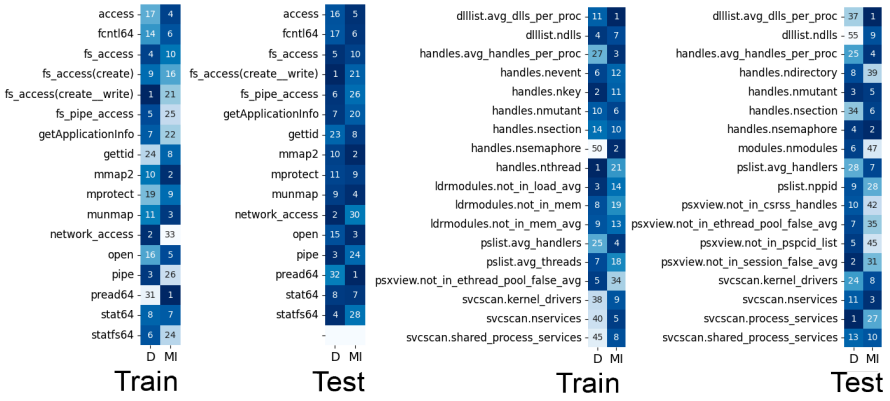
were optimized with the tree-structured Parzen estimator algorithm, using 20% of the entire training set as a validation set. The search spaces of the hyper-parameter optimization are reported in Table 2. The maximum number of epochs was set equal to 150 and an early stopping approach was adopted. The early stopping was based on the lowest loss on the same validation set considered in the hyper-parameter optimization, in order to retain the best models. The FGSM algorithm (as implemented in the Adversarial Robustness Toolbox library[4]) was used to generate the adversarial samples. The adopted implementation of FGSM allowed us to use a mask to decide which input characteristics must be perturbed to generate adversarial samples. The DALEX Python package 1.2.0[5] was integrated to measure the global relevance of input characteristics.

4.3 Results

We evaluate the accuracy performance of deep neural models by measuring the overall accuracy (OA), the average F1-score (MacroF1) and the weighted F1-score (WeightedF1) of decisions produced on the testing set of both CICMalDroid20 and CICMalMem22. We start the analysis, by exploring the ranking of input characteristics determined with both DALEX and Mutual Information on both the training set and the testing set of each dataset. We proceed evaluating the accuracy performance of the deep neural models trained with subsets of input characteristics selected according to the ranking produced by both DALEX and Mutual Information. Finally, we analyse the performance of FGSM used to generate adversarial samples of testing sets by using both DALEX and Mutual Information to identify subsets of input characteristics to perturb.

Input space analysis Figure 1 shows the relevance of the input characteristics as it was measured with DALEX and Mutual Information on both the training set and the testing set of CICMalDroid20 (Fig. 1a) and CICMalMem22 (Fig. 1b), respectively. For each dataset, we consider the union of the top-10 input characteristics, selected according to the ranking provided by both DALEX and Mutual Information. We note that DALEX and Mutual Information return measurements of the relevance of input characteristics that produce a different ranking of these characteristics. This is an expected outcome as DALEX accounts for models' decisions on data, while Mutual Information is model decision-agnostic and fully

[4] https://adversarial-robustness-toolbox.readthedocs.io/
[5] https://github.com/ModelOriented/DALEX

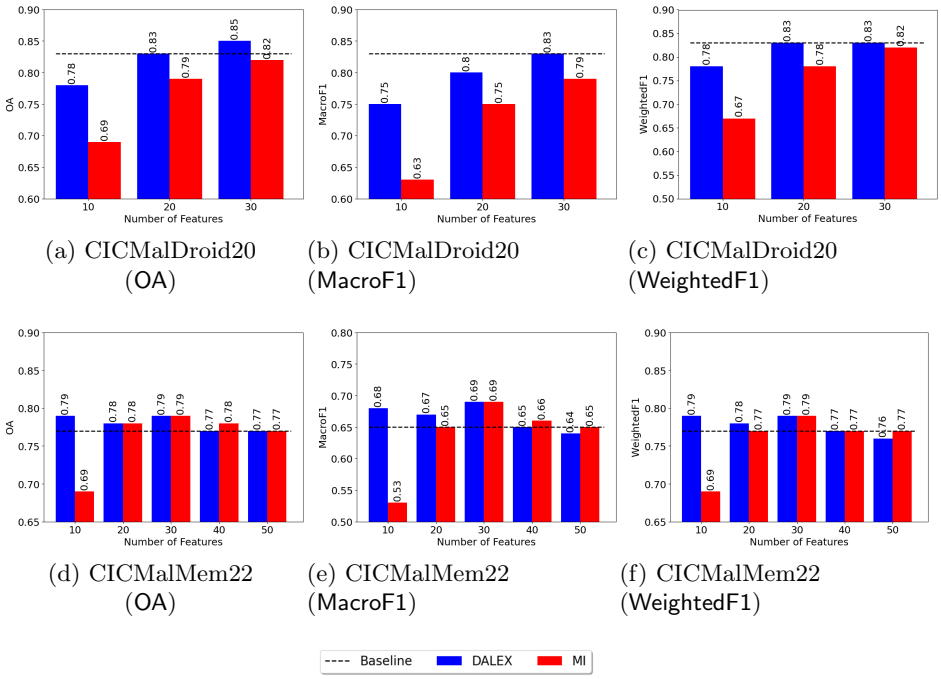


(a) CICMalDroid20 (b) CICMalMem22

Fig. 1: Relevance of the input characteristics measured with DALEX (D) and Mutual Information (MI) on both the training set and the testing set of CICMalDroid20 (Fig. 1a) and CICMalMem22 (Fig. 1b), respectively. For each dataset, the union of the top-10 input characteristics, selected according to the ranking provided by both DALEX and Mutual Information, is selected.

data-driven. Notably, the ranks assigned by the Mutual Information analysis to the input characteristics of both the training set and testing set are similar in both datasets. Instead, the ranks assigned by DALEX to the input characteristics of the training set and testing set are more similar in CICMalDroid20 than in CICMalMem22. In particular, the deep neural model trained for CICMalDroid20 makes decisions on testing samples in a similar way to how it makes decisions on training samples. Differently, the deep neural model trained for CICMalMem22 makes decisions on testing samples differently from how it makes decisions on testing samples.

Selecting input characteristics for training deep neural models Figure 2 shows the accuracy performance of deep neural models trained in both CICMalDroid20 and CICMalMem22 by using both DALEX and Mutual Info to select input characteristics for fueling the training stage. We performed experiments by varying the number of selected characteristics among 10, 20 and 30 in CICMalDroid20, 10, 20, 30, 40 and 50 in CICMalMem22 and we considered the deep neural models trained with all input characteristics of both training sets as baselines. The deep neural model trained with input characteristics selected with DALEX systematically outperforms the deep neural model trained with input characteristics selected with Mutual Information in CICMalDroid20. Notably, in this dataset, the deep neural model trained with 20 out of the 40 original input characteristics selected with DALEX achieves an accuracy performance comparable to that of the deep neural model trained with all input



(a) CICMalDroid20 (OA) (b) CICMalDroid20 (MacroF1) (c) CICMalDroid20 (WeightedF1)

(d) CICMalMem22 (OA) (e) CICMalMem22 (MacroF1) (f) CICMalMem22 (WeightedF1)

Fig. 2: CICMalDroid20 and CICMalMem22: OA (axis Y, Figs. 2a and 2d), MacroF1 (axis Y, Fig. 2b and 2e) and WeightedF1 (axis Y, Fig. 2c and ref-fig:MalMem WeightedF1) of deep neural models trained by selecting the top-n characteristics (axis X) of the original input space. The top-n characteristics are selected with respect to relevance values of input characteristics measured with DALEX and Mutual Information (MI). Baseline denotes the deep neural model learned with all characteristics of the original input space.

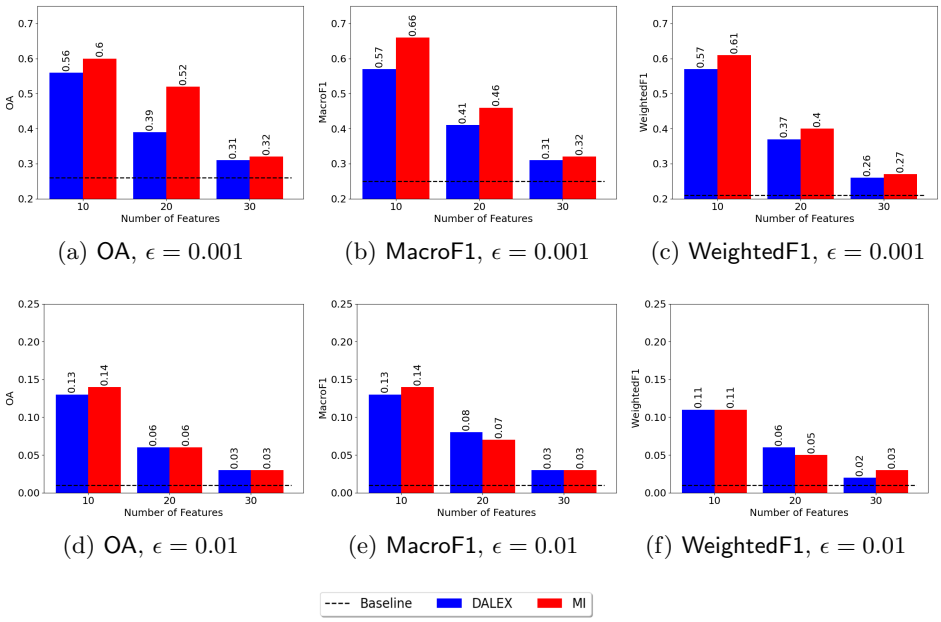
characteristics. On the other hand, in CICMalMem22, the deep neural model trained with 10 out of the 55 original input characteristics selected with DALEX achieves higher accuracy than the deep neural model trained with all input characteristics. The same is not observed by selecting 10 input characteristics with Mutual Information. We must select at least 20 input characteristics with Mutual Information to train a deep neural model that outperforms the deep neural model trained with all input characteristics. In short, both DALEX and Mutual Information allows us to select a subspace of input characteristics to learn a simpler deep neural model that achieves the same accuracy or even better accuracy that the deep neural model trained with all input characteristics. However, DALEX outperforms Mutual Information in this task.

Selecting input characteristics for generating adversarial samples Figures 3 and 4 show the accuracy performance of the deep neural models trained

with all characteristics from the training sets of both CICMalDroid20 and CICMalMem22 and tested on the adversarial testing sets of the same datasets. For each testing set, the adversarial testing set was produced by perturbing samples with FGSM with $\epsilon = 0.001$ and $\epsilon = 0.01$. We used FGSM on the input characteristics selected according to the ranking of characteristics determined on the testing set with both DALEX and Mutual Information. We performed experiments by the number of selected characteristics among 10, 20 and 30 in CICMalDroid20, 10, 20, 30, 40 and 50 in CICMalMem22 and we considered adversarial testings sets constructed perturbing all input characteristics as baselines. Results show that, in CICMalDroid20, DALEX is more effective than the Mutual Information analysis to allow attackers to identify sub-spaces of input characteristic to select for perturbing testing samples and fooling the deep neural model. On the other hand, in CICMalMem22, the Mutual Information analysis helps more than DALEX the attackers to understand which input characteristics to perturb to fool the deep neural model. We explain this result by recalling that DALEX found two completely different rankings for the input characteristics of the training set and testing set of MalMem22. This suggests that DALEX can actually disclose useful knowledge for the attackers when it produces similar explanations of the deep neural model decisions on both the data used for training the model and data considered to fool the model.

5 Conclusion

In this paper, we illustrate a study that we have conducted to explore how a global, post-hoc XAI technique can help in learning simpler deep neural models by possibly gaining accuracy. The study is conducted in the field of malware detection by exploring the performance of DALEX, that is a post-hoc XAI technique. We use DALEX to explain how input characteristics may condition output decisions of deep neural models trained for malware detection and classification. Specifically, we use decision explanations produced by DALEX to select the subset of input characteristics that can allow us to train simpler deep neural models, that preserve, or even gain, accuracy compared to deep neural models trained with all input characteristics. In addition, we explore how DALEX can help attackers in identifying the input characteristics to perturb with FGSM, in order to fool a deep neural model. We describe an empirical study conducted considering two benchmark malware datasets. In this study, we compare the performance of the input characteristic ranking performed with DALEX to the ranking performed with the Mutual Information analysis. The results show that DALEX helps in training a simpler, more accurate deep neural models in both problems. However, DALEX helps attackers to identify the better subset of input characteristics to perturb only when a similar ranking is retrieved on both the training data and the testing data. As future work, we plan to extend this study to various cybersecurity problems (e.g., network intrusion detection, review spam detection) and various XAI techniques (e.g., SHAP).



(a) OA, $\epsilon = 0.001$ (b) MacroF1, $\epsilon = 0.001$ (c) WeightedF1, $\epsilon = 0.001$

(d) OA, $\epsilon = 0.01$ (e) MacroF1, $\epsilon = 0.01$ (f) WeightedF1, $\epsilon = 0.01$

Fig. 3: CICMalDroid20: OA (axis Y, Figs. 3a and 3d), MacroF1 (axis Y, Figs. 3b and 3b) and WeightedF1 (axis Y, Figs. 3c and 3f) of deep neural models trained on the original training set and evaluated on the testing set perturbed using FGSM with $\epsilon = 0.001$ and 0.01. Perturbations are performed on the top-n input characteristics selected according to relevance values measured with DALEX and Mutual Information (MI). Baseline denotes the evaluation performed on the testing set with the perturbation applied to all the 40 input characteristics.

Acknowledgment

Malik AL-Essa is supported by PON RI 2014-2020 - Machine Learning per l'Investigazione di Cyber-minacce e la Cyber-difesa - CUP H98B20000970007. Giuseppina Andresini is supported by the project FAIR - Future AI Research (PE00000013), Spoke 6 - Symbiotic AI, under the NRRP MUR program funded by the NextGenerationEU. Annalisa Appice and Donato Malerba are partially supported by project SERICS (PE00000014) under the NRRP MUR National Recovery and Resilience Plan funded by the European Union - NextGenerationEU.

References

1. AL-Essa, M., Andresini, G., Appice, A., Malerba, D.: An XAI-based adversarial training approach for cyber-threat detection. In: 2022 IEEE International Con-

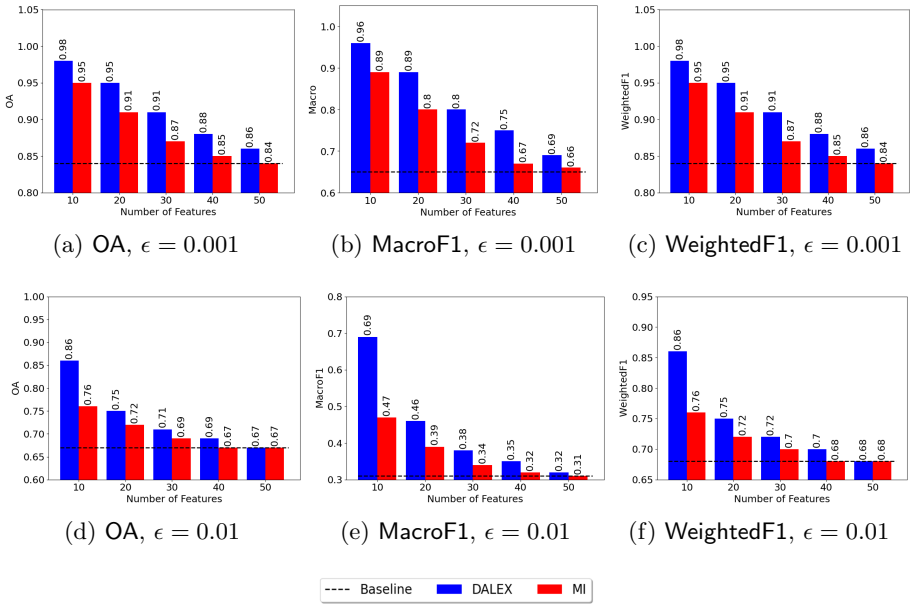


Fig. 4: CICMalMem22: OA (axis Y, Figs. 4a and 4d), MacroF1 (axis Y, Figs. 4b and 4e) and WeightedF1 (axis Y, Figs. 4c and 4f) of deep neural models trained on the original training set and evaluated on the testing set perturbed using FGSM with $\epsilon = 0.001$ and 0.01. Perturbations are performed on the top-n input characteristics selected according to relevance values measured with DALEX and Mutual Information (MI). Baseline denotes the evaluation performed on the testing set with the perturbation applied to all the 55 input characteristics.

ference on Cyber Science and Technology Congress, CyberSciTech 2023. pp. 1–8 (2022)

2. AL-Essa, M., Andresini, G., Appice, A., Malerba, D.: XAI to explore robustness of features in adversarial training for cybersecurity. In: Foundations of Intelligent Systems - 26th International Symposium, ISMIS 2022, Proceedings. Lecture Notes in Computer Science, vol. 13515, pp. 117–126. Springer (2022)
3. Andresini, G., Appice, A., Caforio, F.P., Malerba, D., Vessio, G.: ROULETTE: A neural attention multi-output model for explainable network intrusion detection. Expert Systems with Applications p. 117144 (2022)
4. Andresini, G., Appice, A., Mauro, N.D., Loglisci, C., Malerba, D.: Exploiting the auto-encoder residual error for intrusion detection. In: 2019 IEEE European Symposium on Security and Privacy Workshops, EuroS&P Workshops 2019. pp. 281–290. IEEE (2019)
5. Andresini, G., Pendlebury, F., Pierazzi, F., Loglisci, C., Appice, A., Cavallaro, L.: INSOMNIA: towards concept-drift robustness in network intrusion detection. In: 14th ACM Workshop on Artificial Intelligence and Security. pp. 111–122. ACM (2021)

6. Andriushchenko, M., Flammarion, N.: Understanding and improving fast adversarial training. In: Advances in Neural Information Processing Systems: Annual Conference on Neural Information Processing Systems, NeurIPS 2020. pp. 16048–16059 (2020)
7. Asha, S., Vinod, P.: Evaluation of adversarial machine learning tools for securing AI systems. Clust. Comput. **25**(1), 503–522 (2022)
8. Biecek, P.: DALEX: Explainers for complex predictive models in R. Journal of Machine Learning Research **19**(84), 1–5 (2018)
9. Capuano, N., Fenza, G., Loia, V., Stanzione, C.: Explainable artificial intelligence in cybersecurity: A survey. IEEE Access **10**, 93575–93600 (2022)
10. Carrier, T., Victor, P., Tekeoglu, A., Lashkari, A.H.: Detecting obfuscated malware using memory feature engineering. In: Mori, P., Lenzini, G., Furnell, S. (eds.) Proceedings of the 8th International Conference on Information Systems Security and Privacy, ICISSP 2022. pp. 177–188. SCITEPRESS (2022)
11. Fisher, A., Rudin, C., Dominici, F.: All models are wrong, but many are useful: Learning a variable's importance by studying an entire class of prediction models simultaneously. Journal of Machine Learning Research **20**, 1–81 (2019)
12. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. In: 3rd International Conference on Learning Representations, ICLR 2015, Conference Track Proceedings. pp. 1–11 (2015)
13. Huynh, M.T., Le, H.T., Nguyen, X.H., Le, K.H.: Deep feature selection for machine learning based attack detection systems. In: 2022 IEEE International Conference on Communication, Networks and Satellite, COMNETSAT 2022. pp. 339–344. IEEE (2022)
14. Kuppa, A., Le-Khac, N.: Black box attacks on explainable artificial intelligence(xai) methods in cyber security. In: 2020 International Joint Conference on Neural Networks, IJCNN 2020. pp. 1–8. IEEE (2020)
15. Kuppa, A., Le-Khac, N.A.: Adversarial XAI methods in cybersecurity. IEEE Transactions on Information Forensics and Security **16**, 4924–4938 (2021)
16. Liang, H., He, E., Zhao, Y., Jia, Z., Li, H.: Adversarial attack and defense: A survey. Electronics **11**(8) (2022)
17. Mahdavifar, S., Alhadidi, D., Ghorbani, A.A.: Effective and efficient hybrid android malware classification using pseudo-label stacked auto-encoder. J. Netw. Syst. Manag. **30**(1), 22 (2022)
18. Singla, A., Bertino, E.: How deep learning is making information security more intelligent. IEEE Secur. Priv. **17**(3), 56–65 (2019)
19. Vergara, J., Estevez, P.: A review of feature selection methods based on mutual information. Neural Computing and Applications **24** (01 2014)
20. Wang, M., Zheng, K., Yang, Y., Wang, X.: An explainable machine learning framework for intrusion detection systems. IEEE Access **8**, 73127–73141 (2020)