Black Box Attacks on Explainable Artificial Intelligence(XAI) methods in Cyber Security

Aditya Kuppa
School of Computer Science
University College Dublin
Dublin, Ireland
aditya.kuppa@ucdconnect.ie

Nhien-An Le-Khac
School of Computer Science
University College Dublin
Dublin, Ireland
an.lekhac@ucd.ie

***Abstract*—Cybersecurity community is slowly leveraging Machine Learning (ML) to combat ever evolving threats. One of the biggest drivers for successful adoption of these models is how well domain experts and users are able to understand and trust their functionality. As these black-box models are being employed to make important predictions, the demand for transparency and explainability is increasing from the stakeholders.**

Explanations supporting the output of ML models are crucial in cyber security, where experts require far more information from the model than a simple binary output for their analysis. Recent approaches in the literature have focused on three different areas: (a) creating and improving explainability methods which help users better understand the internal workings of ML models and their outputs; (b) attacks on interpreters in white box setting; (c) defining the exact properties and metrics of the explanations generated by models. However, they have not covered, the security properties and threat models relevant to cybersecurity domain, and attacks on explainable models in black box settings.

In this paper, we bridge this gap by proposing a taxonomy for Explainable Artificial Intelligence (XAI) methods, covering various security properties and threat models relevant to cyber security domain. We design a novel black box attack for analyzing the consistency, correctness and confidence security properties of gradient based XAI methods. We validate our proposed system on 3 security-relevant data-sets and models, and demonstrate that the method achieves attacker's goal of misleading both the classifier and explanation report and, only explainability method without affecting the classifier output. Our evaluation of the proposed approach shows promising results and can help in designing secure and robust *XAI* methods.

***Index Terms*—Adversarial Attack, Explainable Artificial Intelligence, Cyber security, gradient-based XAI, deep learning**

I. Introduction

The term *Explainable Artificial Intelligence (XAI)* was framed by Lent et.al [1] to explain the behavior of AI-controlled entities in simulation game applications. The field of explanations of intelligent systems was active in the 1970s mainly focused around expert systems [2], and in the 1990s around neural networks [3]. The success of Machine Learning (ML) systems, mainly Deep Learning (DL) in various domains and challenge to intuitively understand the outputs of complicated models – how does a DL model arrived to a specific decision for a given input has spurred intensive research into *XAI* methods.

Adding interpretability at different stages of ML pipeline improves the design, implementability, and adaptation of the system for the following reasons: (a) Explainable results ensure data-driven decision-making, i.e. to detect, and consequently, correct from bias in the training dataset. This is important in security where data sets are highly imbalanced; (b) Explainable reports facilitates in improving the robustness of the system by highlighting potential adversarial examples that could change the prediction; (c) Interpretability can guarantee that only contextually correct variables infer the output, i.e., ensuring that an underlying truthful causality exists in the model reasoning. This is vital in security for use cases like attribution of threats [4] etc.

Interpretability methods coupled with the human in the loop improves the trust and security in the decision making process [10] of ML systems. However, recent studies have shown that the explainable methods are fragile and are potentially susceptible to malicious manipulations in the image recognition domain. Recent work [29] has showed that post-hoc methods are fragile, where small changes in the input cause significant changes in the interpretations. This fragility is undesirable from a safety and security perspective. Part of this fragility could be attributed to: (1) the black-box nature of the underlying models that post-hoc methods are trying to explain, and (2) the explanations themselves are models that can be fragile [30].

In the context of the cybersecurity domain, little work is done to understand the security robustness of explainable methods with a realistic threat models. Motivated by this, in this study we aim to conduct a security analysis of gradient-based XAI methods. More specifically, we seek to answer- How can an attacker deceive target classifiers and explainable methods both together, given only explainable model keeping the output of classifier similar? We first define the properties of the threat model for XAI methods into a unified attack framework and conduct both analytical and qualitative studies of the security properties of these methods under realistic assumptions of the real-world adversary. The contribution of this paper can be listed as:

- We propose a three dimension taxonomy for *XAI*, relevant to cyber security domain- (a) explanations of predictions/-data itself $X-PLAIN$; (b) explanations for security and privacy properties of predictions/data $XSP-PLAIN$; and (c) explanations covering threat model of prediction-

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

s/data under consideration $XT - PLAIN$.

- We propose a novel *black box* adversarial attack for testing the consistency, correctness and confidence properties of gradient based *XAI* methods.
- We test our proposed system on 3 security-relevant datasets and models, and show that the method achieves attacker goal's with threat models which reflect the real world settings.

The rest of paper is organized as follows. Section II presents the context of our research and related work in literature on XAI methods and attacks on XAI methods. We describe the role of XAI in cybersecurity in Section III. We also define the three dimensions of XAI space as well as illustrate the threat model for XAI in this section. We present and implement our adversarial-based approach of XAI in Section IV. We evaluate the experimental results and discuss our findings in Section V. Finally, we conclude this paper in the last section.

II. Related work

A. Attacks on XAI Methods

Very recently, some works [34]–[37] are beginning to study adversarial robustness by exploring the spectrum between classification accuracy and network interpretability. Zhang et al. present [35] a class of white-box attacks that generate adversarial inputs which not only mislead target deep learning classifiers but also their coupled interpretation models. They benchmark the proposed method with four different class of explainers. It was shown in [34] that an imperceptible adversarial perturbation to fool classifiers can lead to a significant change in a class-specific network interpretability map. In [36], it was demonstrated that explanation maps can be sensitive to small perturbations in the image domain. There has been some recent research on manipulating explanations in the context of image classification. Authors in [37] show modifying inputs in such a way that is imperceptible to humans.

In [30] showed that the post-hoc explanations are not faithful but only present correlations of the underlying computations. [13] showed that in the case of structured data, LIME and SHAP explanations are not intuitive. In recent work Dylan et al. [14] proposed a novel framework that can effectively hide discriminatory biases of any black-box classifier and fool the post-hoc explanation techniques such as LIME and SHAP.

Our work focuses on black box targeted attacks on two classes of adversaries. One aims to compromise the integrity of the underlying classifier and explainer and the other tries to attack only the explainer without changing the prediction of the classifier i.e. given a natural sample only change the explanation map.

III. *XAI* in Security Domain

A. Role of XAI in Security Domain

Explainable Security (XSec) - a extension of XAI to security domain was proposed by Luca et.al [26]. They discuss the "Six Ws" of XSec (Who? What? Where? When? Why? and How?). They argue that XSec has unique and complex characteristics that involve different consumers (i.e., the system's developers, analysts, users, and attackers) and is multi-faceted by nature (as it requires reasoning about system model, threat model and properties of security, privacy and trust [27]).

There are growing research efforts into methods to formally evaluate and compare explainers. In a recent survey, Murdoch et al. [31] proposed predictive accuracy, descriptive accuracy and relevancy(which is judged by humans) as three essential properties for evaluating explainers. Hall et al. [32] compiled a set of objective characteristics- effectiveness, versatility, constraints (i.e., privacy, computation cost, information collection effort) and the type of generated explanations without human evaluations. Metrics proposed by Alvarez-Melis [28] cover explicitness – intelligibility of explanations, faithfulness – feature relevance, and stability – consistency of explanations for similar or neighboring samples. Fidelity of explanations was evaluated by Yeh et al. [11] by quantifying the degree to which an explanation captures the underlying model changes. Yang et al. [33] proposed three complementary metrics to evaluate explainers: model contrast score – comparing two models trained to consider opposite concepts as important, input dependence score – comparing one model with two inputs of different concepts, and input dependence rate – comparing one model with two functionally identical inputs. These metrics aim to specifically cover aspects of false-positives.

B. Taxonomy of XAI in Security Domain

In the context of the security domain, we divide the explainability space into - (a) explanations of predictions/data itself $X - PLAIN$; (b) explanations covering security and privacy properties of predictions/data $XSP - PLAIN$; (c) explanations covering threat model of predictions/data under consideration $XT - PLAIN$.

1) $X - PLAIN$: This space covers the following type of explanations:

- *static* vs. *interactive* changes in explanations seen by user in response to feedback.
- *local* vs. *global* explanations.
- *in-model* vs. *post-hoc* model explanations that cover models, which are transparent by their nature vs. use of a auxiliary method to explain a model after it has been trained.
- *surrogate* model is a second, usually directly interpretable model that approximates a more complex model, while a *visualization* of a model may focus on parts of it and is not itself a full-fledged model.

2) $XSP - PLAIN$: The $XSP - PLAIN$ explanations include:

- Confidentiality properties of data and model e.g. which features of the data are protected by system owner.
- Integrity properties of data and model e.g. when and how the data was collected and model was trained to accommodate domain shifts etc. Fairness property can be part of model integrity in which explanations can help expose fairness violations by providing insights into possible biases in a model.

- Privacy properties of data and model in the explanations e.g. which part of the data/predictions is exposed to whom. For the publicly released training data and models, have noise added to them so that data rights or model privacy are not compromised? Global explainability methods need to investigate ways to provide explanations about the model without providing details on model weights (directly or via feature importance scores).

3) $XT - PLAIN$: This space captures the properties of threat models considered at the time of training and deployment. e.g. data poisoning protection, thresholds used, etc. Below we list some of the properties of *XAI* methods that are relevant to threat modeling in the security domain.

- *Correctness:* Correctness evaluates the ability of an explainer to correctly identify components of the input that contribute most to the prediction of the classifier.
- *Consistency:* It is the measure of the explainer's ability to capture the relevant components under various transformations to the input. More specifically, if the classifier predicts the same class for both the original and transformed inputs, consistency attempts to measure whether the generated explanation for the transformed input is similar to the one generated for the original input modulo the transformation.
- *Transferability:* Explainability is an advocate for transferability, since it may ease the task of elucidating the boundaries that might affect a model, allowing for a better understanding and implementation. Similarly, the mere understanding of the inner relations taking place within a model facilitates the ability of a user/attacker to reuse this knowledge craft an attack.
- *Confidence:* as a generalization of robustness and stability, confidence should always be assessed on a model in which reliability is expected. As stated in [23]–[25], stability is a must-have when drawing interpretations from a certain model. Trustworthy interpretations should not be produced by models that are not stable. Hence, an explainable model should contain information about the confidence of its working regime.
- *Fairness:* From a social standpoint, explainability can be considered as the capacity to reach and guarantee fairness in ML models. One of the objectives of XAI is highlighting bias in the data a model was exposed to [5], [22]. The support of algorithms and models is growing fast in fields that involve human lives, hence explainability should be considered as a bridge to avoid the unfair or unethical use of the algorithm's outputs.
- *Privacy:* One of the byproducts enabled by explainability in ML models is its ability to assess privacy. ML models may have complex representations of their learned patterns. Not being able to understand what has been captured by the model [6] and stored in its internal representation may entail a privacy breach. Contrarily, the ability to explain the inner relations of a trained model by non-authorized third parties may also compromise the differential privacy of the data origin. Ideally, XAI should be able to explain the knowledge within a model and it should be able to reason about what the model acts upon. However, the information revealed by XAI techniques can be used both to generate more effective attacks in adversarial contexts aimed at confusing the model, at the same time as to develop techniques to better protect against private content exposure by using such information.

IV. Proposed method

A. Attack method on Consistency, Correctness and Confidence

Attacks, which compromise the confidence of underlying system can be divided into two different categories based on which part of the system is compromised: (a) $CI-$ Attacks which target the classifier as well as its coupled interpreter, (b) $I-$ Attacks which target only underlying interpreter keeping the classifier's decision intact.

More formally, we consider a neural network $f : \mathbb{R}^d \rightarrow \mathbb{R}^K$ with relu non-linearities that classifies a sample $x \in \mathbb{R}^d$ in K categories with the predicted class given by $k = \text{argmax}_i\, f(x)_i$. The explanation map is denoted by $g : \mathbb{R}^d \rightarrow \mathbb{R}^d$ and associates a sample with a vector of the same dimension whose components encode the relevance score of each feature for the neural network's prediction.

Let $g^t \in \mathbb{R}^d$ be a given target explanation map and $x \in \mathbb{R}^d$ an input sample. For $I-$ attack we want to construct a manipulated sample $x_{adv} = x + \delta x$ such that it has an explanation very similar to the target g^t but the output of the network stays approximately constant,i.e. $f(x_{adv}) \approx f(x)$. For $CI-$ attack both $g(x_{adv})^t \approx g(x)^t$ and $f(x_{adv}) \neq f(x)$.

The norm of the perturbation δx added to the input is small, i.e. $\|\delta x\| = \|x_{adv} - x\| \ll 1$.

As per threat model, there are additional constraints on the attacker as follows:

- The attack is performed in black-box settings i.e., attacker has no knowledge of the underlying architecture of the model, hyperparameters used, and training data distributions.
- The perturbation δ changing the original instance x into $x_{\text{adv}} = x + \delta$ should be sparse.
- The adversarial sample x_{adv} needs to be interpretable. We consider an instance x_{adv} interpretable if it lies close to the model's training data distribution. Let us illustrate this with an intuitive example. Assume we are predicting given traffic flow as malicious or not with features including the protocol type and the Time-to-live (TTL) of the packet. Most of the benign flows have a TTL below a predefined value and we would like to know what needs to change about the flow in order to make it adversarial. By simply increasing the TTL and leaving the other features unchanged, the method outputs adversarial sample, and classifier and corresponding explainer can output that our *adv. flow* is malicious. This sparse adversarial sample lies fairly close to the overall training distribution since only one feature value was changed. The adversarial sample is however out-of-distribution with regards to the subset of benign flows in the training data valued above pre-defined

value because other relevant features like the protocol still resemble a typical normal flow. As a result, we do not consider this adversarial sample to be very interpretable.

With these constraints in place, we now present the attack steps. We follow a two-step strategy. For step 1, we use the black-box model attack proposed in [12]. The attacker collects a small set of n test data and start querying the prediction and explanation report for each sample. One can train a surrogate classifier with collected outputs of test samples but data n may be limited and also the attacker has no knowledge of distributions of original training data of which the model was trained. We run Manifold Approximation Algorithm (MAA) [9] on n data points to find the best piece wise spherical manifold or subspace $\hat{M}$ and the projection map $p : \alpha^p \rightarrow \hat{M}$ such that the mean square error $\sum_{i=1}^{n} \|x_i - p(x_i)\|^2$ is minimized, where $x_i \in \mathbb{R}^{\alpha}$.

The output of step 1 helps us to divide the collected n samples into various data distributions $z_1, z_2..z_i$. Similarly, We can run the MAA on explanation reports to find explanation distributions $g_1, g_2...g_i$ for each data distribution. At the end of step 1, we will have data and explanation distributions of dataset (n samples) that the attacker has collected.

For the success of our attack, in Step 2 we need to induce minimal distortions on the input distribution z_a to move the decision boundary to z_i along with g_a to move to g_i where a is the natural sample distribution and i is the target sample distribution. Perturbations based on data distribution helps us satisfying the data manifold constraint i.e. adversarial sample x_{adv} needs to be interpretable.

To achieve this, we solve Equation 1.

$$\begin{aligned} \min_{\boldsymbol{d}} \quad & \Delta(\boldsymbol{z_a}, \boldsymbol{z_i}) + \Delta(\boldsymbol{g_a}, \boldsymbol{g_i}) + C\|\boldsymbol{d}\| \\ \text{s.t.} \quad & S_p^a(V_a, c_a, r_a) \leq \boldsymbol{x} + \boldsymbol{d} \leq S_p^i(V_i, c_i, r_i) \end{aligned} \tag{1}$$

where C is the regularisation constant that balances approaching the target and limiting the input distortion and Kullback-Leibler (KL)-divergence as Δ and $s(X) = S(V, c, r)$ is sphere centered at c with radius r and V determines an affine subspace the sphere lies in for all data X. In case of $I-$ attack $\Delta(\boldsymbol{z_a}, \boldsymbol{z_i})$ = 0, here we aim to change the explanation of natural sample to target sample keeping the label intact.

We measure the success of attack using attack success rate, which is defined by

$$\text{Attack Success Rate (ASR)} = \frac{\#\,\text{successful attempts}}{\#\,\text{total attempts}}$$

V. Evaluation and Discussion

A. Experimental Setting

Here we introduce the setting of our experiments. We choose three different security-relevant systems for our evaluation.

System 1: Mimicus a multilayer perceptron that is capable of detecting malicious PDF documents. We use the same features as discussed in the original paper [20] and implemented the system based on the work of [18]. The network architecture consists of two hidden layers with 200 nodes each. 135 features were extracted from PDF documents. These features cover properties about the document structure, such as the number of sections and fonts in the document, and are mapped to binary values as described by [18]. For training the model we use the original dataset, which consists of 5,000 benign and 5,000 malicious PDF files. For adversarial feature value perturbation, we also ensure that the generated files are valid and realistic by applying domain-specific constraints. For example, generated PDFs need to follow the PDF specification to ensure that a PDF viewer can open the test file. Explanation reports are generated by gradient-based methods- Input*Gradient(I*G), Layer-Wise Relevance Propagation(LRP), Guided Back Propagation(GBP), SmoothGrad(SG), Gradient(GRAD), and Integrated Gradients(IG). We adopt their open-source implementation in our evaluation.

System 2: Drebin proposed [19] uses a multilayer perceptron for identifying Android malware. The network consists of two hidden layers, each comprising 200 neurons. It builds on features developed by [21]. The dataset has 129,013 Android applications among which 123,453 are benign and 5,560 are malicious. There is a total of 545,333 binary features categorized into eight sets including the features captured from manifest files (e.g., requested permissions and intents) and disassembled code (e.g., restricted API calls and network addresses). For adversarial feature value perturbation, we enforce a constraint that only allows modifying features related to the Android manifest file and thus ensures that the application code is unaffected. The features are only added but never deleted from the manifest files to ensure that no application functionality is changed due to insufficient permissions. Explanation reports are generated by gradient-based methods- Input*Gradient(I*G), Layer-Wise Relevance Propagation(LRP), Guided Back Propagation(GBP), SmoothGrad(SG), Gradient(GRAD), and Integrated Gradients(IG). We adopt their open-source implementation in our evaluation.

System 3: The third system is an Intrusion Detection System, which uses Adversarial Auto Encoder(AAE) [15] for Anomaly Detection(AD) task on recently published UGR16 dataset [16]. The dataset consists of NetFlow traces captured from a real network. For our experiment, we use dataset which consists of 5,990,295 data points. We extract 53 aggregated features, which include mean and standard deviation of flow duration, number of packets, number of bytes, packet rate; and byte rate; entropy of protocol type, destination IP addresses, source ports, destination ports, and TCP flags; and proportion of ports used for common applications (e.g. WinRPC, Telnet, DNS, SSH, HTTP, FTP, and POP3).

An AAE is a generative model that is trained with dual objectives i.e., a traditional reconstruction error criterion and an adversarial training criterion. The encoder in AAE learns to convert the data distribution to a latent representation with an arbitrary prior distribution, attempting to minimize the reconstruction error. In other words, a GAN is attached to the latent layer. AAE used four layers of (conv-batch-normalization-elu) in the encoder and decoder part of the

network. The AAE network parameters such as (number of filters, filter size, strides) are chosen to be (53,10,1) for first and second layers and (53,10,1) for third and fourth layers of both encoder and decoder layers. The middle hidden layer size is set to be same as rank $K = 20$ and the model is trained using Adam. Once the parameters are optimized after training, the AAE model is used for anomaly detection [15], [17], where an IP address and its time window is recognized as abnormal when the reconstruction error of its input features is high. Here, the reconstruction error is the mean square difference between the observed features and the expectation of their reconstruction as given by the AAE. The threshold we used is 5% of the data as anomalies as this reflect the actual data set distribution. For adversarial feature value perturbation, we impose the constraint that the distortion added should not create malformed packets when features are transformed back to *pcap*. To solve this constraint we selectively add noise to a small set of features instead of perturbing any of 53 features and we use Scapy[1] to generate valid packets for changed values keeping other properties of the flow intact. For explanations reports, we analyze the gradients contributed by each feature of the data point. Given the trained AAE and an anomalous data point, the gradient map between the feature and reconstruction error is used to explain the contribution of a particular feature to the anomaly.

Table I summarizes the performance of 3 systems we use to evaluate our proposed method in our experiments.

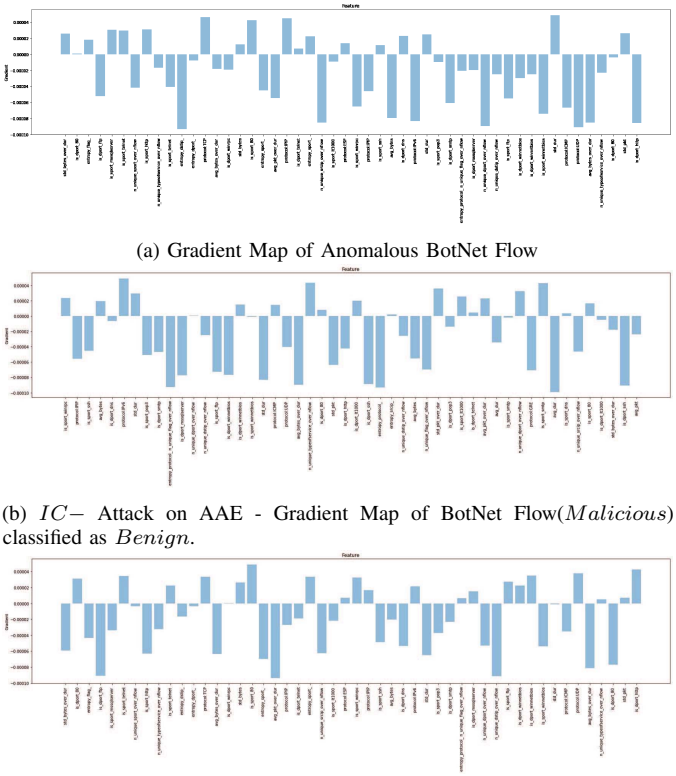
TABLE I: Performance of models used in Experiments.

System	Accuracy	Precision	Recall	F1-Score
S1(Mimicus)	0.975	0.991	0.998	0.994
S2(Drebin)	0.980	0.926	0.924	0.925
S3(AAE-BotNet AD)	0.967	0.931	0.965	0.948

TABLE II: Results for the proposed CI and I attack on explainable method vs three systems with n(1000(S1,S2),10000(S3)). For S3 we only report gradient method n was sampled randomly from all dataset and removed from the training set to reflect the real attacker.

Explainable Method	LRP	SG	GRAD	I*G	GBP	IG
Attack Success Rate for $IC-$ Attack						
Mimicus(S1)	0.98	0.94	1.00	0.99	0.98	0.99
Drebin(S2)	0.98.4	0.93	1.00	0.984	0.982	0.987
AAE(S3)	X	X	1.00	X	X	X
Attack Success Rate for $I-$ Attack						
Mimicus(S1)	0.983	0.93	0.99	0.98	0.984	0.978
Drebin(S2)	0.97	0.92	0.99	0.975	0.969	0.967
AAE(S3)	X	X	0.98	X	X	X

To solve equation 1 we use SciPy [8] L-BFGS-B [7] optimiser, with initial disturbance sampled from a uniform distribution $\mathcal{U}(10^{-8}, 10^{8})$, 50 corrections on the memory matrix, and termination test tolerance of 10. We set n to 1000 for Mimicus and Drebin and for BotNet flows we set to 10000. We randomly sampled 2% of each malicious type dataset and 8% of benign data from original dataset for n. We discarded n from the training data to reflect the threat model, i.e., attacker has no access to training data.



(a) Gradient Map of Anomalous BotNet Flow

(b) $IC-$ Attack on AAE - Gradient Map of BotNet Flow($Malicious$) classified as $Benign$.

(c) $I-$ Attack on AAE - Gradient Map of BotNet Flow($Malicious$) changed from original explanation map

Fig. 1: Gradient Map changes of AAE for $IC-$ and I attack

B. Results and Discussion

Table II summarizes Attack Success Rate(ASR) for each system against the gradient-based explainable methods. SmoothGrad(SG) method performed better compared to its counterparts.

For the qualitative evaluation of our attack, we visualize the relevance vector of an explanation. First, we normalize the scores to -1 and 1, and highlight features according to whether they support the decision (green color) or contradict the decision (red color). The brightness reflects the importance of the features.

Table III visualizes small set of feature vectors of malicious pdf file with hash[2] with each explanation method. This file exploits certain vulnerabilities in Adobe Acrobat and Reader to obfuscate and hide, and subsequently run, malicious JavaScript and shellcode. Table IV and V illustrates explanation map of same file after $CI-$ and I attack respectively.

Table VI visualizes set of feature vectors of malicious Android file with hash of [3] with each explanation method. This Android APK malware appear to be installers for other applications; when executed however, the malware send SMS

[1]https://scapy.readthedocs.io/en/latest

[2]268aabdc7d2dcebdc42775e36e05ad6c842044b799176d16b3c42105931e7e6

[3]eb1bcca87ab55bd0fe0cf1ec27753fddcd35b6030633da559eee42977279b8db

Id	LRP	SG	G	I*G	GBP	IG
0	createdate_ts	createdate_ts	createdate_ts	createdate_ts	createdate_ts	createdate_ts
1	creator_dot	creator_dot	creator_dot	creator_dot	creator_dot	creator_dot
2	creator_lc	creator_lc	creator_lc	creator_lc	creator_lc	creator_lc
3	creator_uc	creator_uc	creator_uc	creator_uc	creator_uc	creator_uc
4	moddate_ts	moddate_ts	moddate_ts	moddate_ts	moddate_ts	moddate_ts
5	producer_uc	producer_uc	producer_uc	producer_uc	producer_uc	producer_uc
6	title_dot	title_dot	title_dot	title_dot	title_dot	title_dot
7	title_lc	title_lc	title_lc	title_lc	title_lc	title_lc
8	title_oth	title_oth	title_oth	title_oth	title_oth	title_oth

TABLE III: Explanation Map of $Malicious$ PDF file *268aabdc7d2dcebdc42775e36e05ad6c842044b799176d16b3c42105931e7e6*

Id	LRP	SG	G	I*G	GBP	IG
0	createdate_ts	createdate_ts	createdate_ts	createdate_ts	createdate_ts	createdate_ts
1	creator_dot	creator_dot	creator_dot	creator_dot	creator_dot	creator_dot
2	creator_lc	creator_lc	creator_lc	creator_lc	creator_lc	creator_lc
3	creator_uc	creator_uc	creator_uc	creator_uc	creator_uc	creator_uc
4	moddate_ts	moddate_ts	moddate_ts	moddate_ts	moddate_ts	moddate_ts
5	producer_uc	producer_uc	producer_uc	producer_uc	producer_uc	producer_uc
6	title_dot	title_dot	title_dot	title_dot	title_dot	title_dot
7	title_lc	title_lc	title_lc	title_lc	title_lc	title_lc
8	title_oth	title_oth	title_oth	title_oth	title_oth	title_oth

TABLE IV: $IC-$ Attack on Mimicus System, where the classification is changed to $Benign$. Explanation map of the original file changed compared to Table III

Id	LRP	SG	G	I*G	GBP	IG
0	count_acroform	count_acroform	count_acroform	count_acroform	count_acroform	count_acroform
1	count_trailer	count_trailer	count_trailer	count_trailer	count_trailer	count_trailer
2	count_xref	count_xref	count_xref	count_xref	count_xref	count_xref
3	createdate_ts	createdate_ts	createdate_ts	createdate_ts	createdate_ts	createdate_ts
4	creator_dot	creator_dot	creator_dot	creator_dot	creator_dot	creator_dot
5	creator_lc	creator_lc	creator_lc	creator_lc	creator_lc	creator_lc
6	creator_uc	creator_uc	creator_uc	creator_uc	creator_uc	creator_uc
7	moddate_ts	moddate_ts	moddate_ts	moddate_ts	moddate_ts	moddate_ts
8	producer_uc	producer_uc	producer_uc	producer_uc	producer_uc	producer_uc

TABLE V: $I-$ Attack on Mimicus System, where classification of original file $Malicious$ is unchanged and Explanation map of the original file changed compared to Table III

messages to premium-rate numbers or services. Table VII and VIII illustrates explanation map of same file after $CI-$ and $I-$ attack respectively. Figure 1 summarizes proposed attack on AAE anomaly detector.

Our results show that security properties of XAI methods can be compromised by proposed $CI-$ and $I-$ attacks. Both attacks compromises the correctness (the ability of an explainer to correctly identify components of the input that contribute most to the prediction) by changing the prediction of the input without loosing functionality, consistency (the explainer's capability to capture the transformations to the input) by changing the explanation map via feature additions, and confidence of the corresponding explainers.

For the given trained model, and adversarial sample, explanation methods do not agree on the same set of feature vectors that influence the decision of the model. We can infer that the transferability property of adversarial inputs across different explanation methods is low. We speculate that one of the reasons for our attack success is because explanation methods for a given model do not exactly reflect the true state of the model, which allows the adversary to exploit both models and explainers simultaneously. Specifically, gradient-based methods only rely on the gradient information, which can deviate from the true behavior of underlying models.

VI. Conclusion

In this paper, we propose a three dimension taxonomy for *XAI*, relevant to cybersecurity domain. We extend the threat model for XAI methods to reflect the real world settings. We design a novel black-box attacks to study the security properties- consistency, correctness and confidence of gradient based *XAI* methods.

Our attack focuses on two classes of adversaries - CI and I attack. CI attack aims to compromise the integrity of the underlying classifier and explainer simultaneously and I attack tries to attack the only explainer without changing the prediction of the classifier i.e. given a natural sample only change the explanation map. Through empirical and qualitative evaluation, we show the effectiveness of the attack on multiple gradient-based explainers and on 3 security-relevant data-sets and models. Our analysis highlights the gaps between model inner workings and their corresponding explainers, and the adversarial samples are weakly transferable.

Id	LRP	SG	GRAD	I*G	GBP	IG
0	activity::.Ma	activity::.Ma	activity::.Ma	activity::.Ma	activity::.Ma	activity::.MainActivity
1	activity::Mai	activity::Mai	activity::Mai	activity::Mai	activity::Mai	activity::MainActivity
2	feature::andr	feature::andr	feature::andr	feature::andr	feature::andr	feature::android.hardware.telephony
3	feature::andr	feature::andr	feature::andr	feature::andr	feature::andr	feature::android.hardware.touchscreen
4	intent::andro	intent::andro	intent::andro	intent::andro	intent::andro	intent::android.intent.action.BOOT_COMPLETED
5	intent::andro	intent::andro	intent::andro	intent::andro	intent::andro	intent::android.intent.action.MAIN
6	intent::andro	intent::andro	intent::andro	intent::andro	intent::andro	intent::android.intent.action.PHONE_STATE
7	intent::andro	intent::andro	intent::andro	intent::andro	intent::andro	intent::android.intent.action.USER_PRESENT
8	intent::andro	intent::andro	intent::andro	intent::andro	intent::andro	intent::android.intent.category.LAUNCHER
9	permission::a	permission::a	permission::a	permission::a	permission::a	permission::android.permission.ACCESS_NETWORK_STATE
10	permission::a	permission::a	permission::a	permission::a	permission::a	permission::android.permission.DELETE_PACKAGES
11	permission::a	permission::a	permission::a	permission::a	permission::a	permission::android.permission.INSTALL_PACKAGES
12	permission::a	permission::a	permission::a	permission::a	permission::a	permission::android.permission.INTERNET
13	permission::a	permission::a	permission::a	permission::a	permission::a	permission::android.permission.READ_CONTACTS
14	permission::a	permission::a	permission::a	permission::a	permission::a	permission::android.permission.READ_PHONE_STATE
15	permission::a	permission::a	permission::a	permission::a	permission::a	permission::android.permission.RECEIVE_BOOT_COMPLETED
16	permission::a	permission::a	permission::a	permission::a	permission::a	permission::android.permission.RECEIVE_SMS
17	permission::a	permission::a	permission::a	permission::a	permission::a	permission::android.permission.SEND_SMS
18	permission::a	permission::a	permission::a	permission::a	permission::a	permission::android.permission.WRITE_EXTERNAL_STORAGE
19	provider::and	provider::and	provider::and	provider::and	provider::and	provider::android.provider.Telephony.SMS_RECEIVED
20	service_recei	service_recei	service_recei	service_recei	service_recei	service_receiver::AlarmReceiver
21	service_recei	service_recei	service_recei	service_recei	service_recei	service_receiver::AutorunReceiver
22	service_recei	service_recei	service_recei	service_recei	service_recei	service_receiver::SmsReciver

TABLE VI: Explanation Map of $Malicious$ Android file *eb1bcca87ab55bd0fe0cf1ec27753fddcd35b6030633da559eee42977279b8db*

Id	LRP	SG	GRAD	I*G	GBP	IG
0	activity::.Ma	activity::.Ma	activity::.Ma	activity::.Ma	activity::.Ma	activity::.MainActivity
1	activity::Mai	activity::Mai	activity::Mai	activity::Mai	activity::Mai	activity::MainActivity
2	feature::andr	feature::andr	feature::andr	feature::andr	feature::andr	feature::android.hardware.telephony
3	feature::andr	feature::andr	feature::andr	feature::andr	feature::andr	feature::android.hardware.touchscreen
4	intent::andro	intent::andro	intent::andro	intent::andro	intent::andro	intent::android.intent.action.BOOT_COMPLETED
5	intent::andro	intent::andro	intent::andro	intent::andro	intent::andro	intent::android.intent.action.MAIN
6	intent::andro	intent::andro	intent::andro	intent::andro	intent::andro	intent::android.intent.action.PHONE_STATE
7	intent::andro	intent::andro	intent::andro	intent::andro	intent::andro	intent::android.intent.action.USER_PRESENT
8	intent::andro	intent::andro	intent::andro	intent::andro	intent::andro	intent::android.intent.category.LAUNCHER
9	permission::a	permission::a	permission::a	permission::a	permission::a	permission::android.permission.ACCESS_NETWORK_STATE
10	permission::a	permission::a	permission::a	permission::a	permission::a	permission::android.permission.DELETE_PACKAGES
11	permission::a	permission::a	permission::a	permission::a	permission::a	permission::android.permission.INSTALL_PACKAGES
12	permission::a	permission::a	permission::a	permission::a	permission::a	permission::android.permission.INTERNET
13	permission::a	permission::a	permission::a	permission::a	permission::a	permission::android.permission.READ_CONTACTS
14	permission::a	permission::a	permission::a	permission::a	permission::a	permission::android.permission.READ_PHONE_STATE
15	permission::a	permission::a	permission::a	permission::a	permission::a	permission::android.permission.RECEIVE_BOOT_COMPLETED
16	permission::a	permission::a	permission::a	permission::a	permission::a	permission::android.permission.RECEIVE_SMS
17	permission::a	permission::a	permission::a	permission::a	permission::a	permission::android.permission.SEND_SMS
18	permission::a	permission::a	permission::a	permission::a	permission::a	permission::android.permission.WRITE_EXTERNAL_STORAGE
19	provider::and	provider::and	provider::and	provider::and	provider::and	provider::android.provider.Telephony.SMS_RECEIVED
20	service_recei	service_recei	service_recei	service_recei	service_recei	service_receiver::AlarmReceiver
21	service_recei	service_recei	service_recei	service_recei	service_recei	service_receiver::AutorunReceiver
22	service_recei	service_recei	service_recei	service_recei	service_recei	service_receiver::SmsReciver

TABLE VII: $IC-$ Attack on Drebin System, where the explanation map and corresponding classification is changed to $Benign$. Explanation map of the original file changed compared to Table VI

Our future work will focus on three areas: (a) Study the defense mechanisms against the attack proposed; (b) Extend the proposed method to compromise the privacy and confidentiality properties of explainable methods, and (c) Examine the security robustness of other *XAI* with different neural network architectures.

REFERENCES

[1] Van Lent M., Fisher W., Mancuso M. (2004), An explainable artificial intelligence system for small-unit tactical behavior Proceedings of the national conference on artificial intelligence 900–907 2004 Menlo Park, CA; Cambridge

[2] Swartout W. R. (1983), XPLAIN: A system for creating and explaining expert consulting programs Artificial intelligence Elsevier 1983

[3] Andrews R., Diederich J., Tickle, A.B. (1995), Survey and critique of techniques for extracting rules from trained artificial neural networks , Knowledge-based systems 8 ,1995 Elsevier

[4] Kuppa A., Grzonkowski S., Le-Khac N.-A. (2018), RiskWriter: Predicting Cyber Risk of an Enterprise, In: Ganapathy V., Jaeger T., Shyamasundar R. (eds) Information Systems Security. ICISS 2018. Lecture Notes in Computer Science, vol 11281. Springer, Cham, DOI:10.1007/978-3-030-05171-6_5

[5] Burns K. et al. (2018), Women also Snowboard: Overcoming Bias in Captioning Models (2018). arXiv preprint arXiv:1803.09797.

[6] Castelvecchi D. (2016), Can we open the black box of AI?, Nature News 538 (7623) (2016) 20.

[7] Zhu, C. et al. (1997), Algorithm 778: L-BFGS-B: Fortran subroutines

Id	LRP	SG	GRAD	I*G	GBP	IG
0	activity::.Ma	activity::.Ma	activity::.Ma	activity::.Ma	activity::.Ma	activity::.MainActivity
1	activity::Mai	activity::Mai	activity::Mai	activity::Mai	activity::Mai	activity::MainActivity
2	feature::andr	feature::andr	feature::andr	feature::andr	feature::andr	feature::android.hardware.telephony
3	feature::andr	feature::andr	feature::andr	feature::andr	feature::andr	feature::android.hardware.touchscreen
4	intent::andro	intent::andro	intent::andro	intent::andro	intent::andro	intent::android.intent.action.BOOT_COMPLETED
5	intent::andro	intent::andro	intent::andro	intent::andro	intent::andro	intent::android.intent.action.MAIN
6	intent::andro	intent::andro	intent::andro	intent::andro	intent::andro	intent::android.intent.action.PHONE_STATE
7	intent::andro	intent::andro	intent::andro	intent::andro	intent::andro	intent::android.intent.action.USER_PRESENT
8	intent::andro	intent::andro	intent::andro	intent::andro	intent::andro	intent::android.intent.category.LAUNCHER
9	permission::a	permission::a	permission::a	permission::a	permission::a	permission::android.permission.ACCESS_NETWORK_STATE
10	permission::a	permission::a	permission::a	permission::a	permission::a	permission::android.permission.DELETE_PACKAGES
11	permission::a	permission::a	permission::a	permission::a	permission::a	permission::android.permission.INSTALL_PACKAGES
12	permission::a	permission::a	permission::a	permission::a	permission::a	permission::android.permission.INTERNET
13	permission::a	permission::a	permission::a	permission::a	permission::a	permission::android.permission.READ_CONTACTS
14	permission::a	permission::a	permission::a	permission::a	permission::a	permission::android.permission.READ_PHONE_STATE
15	permission::a	permission::a	permission::a	permission::a	permission::a	permission::android.permission.RECEIVE_BOOT_COMPLETED
16	permission::a	permission::a	permission::a	permission::a	permission::a	permission::android.permission.RECEIVE_SMS
17	permission::a	permission::a	permission::a	permission::a	permission::a	permission::android.permission.SEND_SMS
18	permission::a	permission::a	permission::a	permission::a	permission::a	permission::android.permission.WRITE_EXTERNAL_STORAGE
19	provider::and	provider::and	provider::and	provider::and	provider::and	provider::android.provider.Telephony.SMS_RECEIVED
20	service_recei	service_recei	service_recei	service_recei	service_recei	service_receiver::AlarmReceiver
21	service_recei	service_recei	service_recei	service_recei	service_recei	service_receiver::AutorunReceiver
22	service_recei	service_recei	service_recei	service_recei	service_recei	service_receiver::SmsReciver

TABLE VIII: $I-$ Attack on Drebin System, where classification of original file $Malicious$ is unchanged and Explanation map of the original file changed compared to Table VI

for large-scale bound-constrained optimization ACM Transactions on Mathematical Software (TOMS),1997

[8] Jones E., Oliphant T., Peterson P. (2014), {SciPy}: Open source scientific tools for {Python}, 2014

[9] Li D., Dunson D.B. (2017), Efficient manifold and subspace approximations with spherelets, arXiv preprint arXiv:1706.08263,2017

[10] Tao G., Ma S., Liu Y., Zhang X. (2018), Attacks Meet Interpretability: Attribute-Steered Detection of Adversarial Samples. In *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, 2018.

[11] Yeh C.-K. et al. (2019), On the (In) fidelity and Sensitivity of Explanations, Advances in Neural Information Processing Systems,2019

[12] Kuppa, A. and Grzonkowski, S. and Asghar, M. R. and Le-Khac, N-A. (2019), Black Box Attacks on Deep Anomaly Detectors, Proceedings of the 14th International Conference on Availability, Reliability and Security, Aug. 2019, UK DOI: 10.1145/3339252.3339266

[13] Mittelstadt B., Russell C., Wachter S. (2019), Explaining explanations in AI, Proceedings of the conference on fairness, accountability, and transparency 2019

[14] Slack D. et al. (2019), How can we fool LIME and SHAP? Adversarial Attacks on Post hoc Explanation Methods, arXiv preprint arXiv:1911.02508,2019

[15] Kuppa A., Grzonkowski S., Asghar M.R., Le-Khac N.-A. (2019), Finding rats in cats: Detecting stealthy attacks using group anomaly detection, (2019) 18th IEEE International Conference on Trust, Security and Privacy in Computing and Communications, Roturua, New Zealand, Oct 2019, DOI:10.1109/TrustCom/BigDataSE.2019.00066

[16] Fernández G.M. et al. (2018), "UGR'16: A new dataset for the evaluation of cyclostationarity-based network IDSs," *Computers & Security*, vol. 73, pp. 411–424, 2018.

[17] Nguyen Thi N., Cao V.L., Le-Khac N.-A. (2017), One-Class Collective Anomaly Detection Based on LSTM-RNNs. In: Hameurlain A., Kung J., Wagner R., Dang T., Thoai N. (eds) Transactions on Large-Scale Data- and Knowledge-Centered Systems XXXVI. Lecture Notes in Computer Science, vol 10720. Springer, Berlin, Heidelberg

[18] Guo W. et al. (2018), LEMNA: Explaining deep learning based security applications. In *Proc. of the ACM Conference on Computer and Communications Security (CCS)*, pages 364–379, 2018.

[19] Grosse K. et al. (2017), Adversarial examples for malware detection. In *Proc. of the European Symposium on Research in Computer Security (ESORICS)*, pages 62–79, 2017.

[20] Smutz C., Stavrou A. (2012), Malicious PDF detection using metadata and structural features. In *Proc. of the Annual Computer Security Applications Conference (ACSAC)*, pages 239–248, 2012.

[21] Arp D. et al. (2014), Drebin: Efficient and explainable detection of Android malware in your pocket. In *Proc. of the Network and Distributed System Security Symposium (NDSS)*, Feb. 2014.

[22] Bennetot A., Laurent J.-L., Chatila R., Díaz-Rodríguez N. (2019), Towards explainable neural-symbolic visual reasoning, in: NeSy Workshop IJCAI 2019, Macau, China, 2019.

[23] Ruppert D. (1987), Robust statistics: The approach based on influence functions, Taylor & Francis, 1987.

[24] Yu B. et al. (2013), Stability, Bernoulli 19 (4) (2013) 1484–1500.

[25] Basu S., Kumbier K., Brown J.B., Yu B. (2018), Iterative random forests to discover predictive and stable high-order interactions, Proceedings of the National Academy of Sciences 115 (8) (2018) 1943–1948.

[26] Viganò L., Magazzeni D. (2018), Explainable security, arXiv preprint arXiv:1807.04178,2018

[27] Kuppa A., Grzonkowski S., Le-Khac N.-A. (2018), Enabling Trust in Deep Learning Models: A Digital Forensics Case Study, (2018) 17th IEEE International Conference on Trust, Security and Privacy in Computing and Communications, New York, USA, Aug 2018, DOI:10.1109/TrustCom/BigDataSE.2018.00172

[28] Melis D.A., Jaakkola T. (2018), Towards robust interpretability with self-explaining neural networks. In *NeurIPS'18: Neural Information Processing Systems.*, pp. 7775–7784, 2018.

[29] Ghorbani A., Abid A., Zou, J. (2017), Interpretation of neural networks is fragile. In *32nd AAAI Conference on Artificial Intelligence*, 2019.

[30] Rudin C. (2019), Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1(5):206, 2019.

[31] Mohseni S., Niloofar Z., Ragan E.D. (2018), A survey of evaluation methods and measures for interpretable machine learning, arXiv preprint arXiv:1811.11839

[32] Hall M. et al. (2019), A Systematic Method to Understand Requirements for Explainable AI (XAI) Systems, Proceedings of the IJCAI 2019 Workshop on Explainable Artificial Intelligence (XAI) 2019

[33] Yang M., Kim B. (2019), BIM: Towards quantitative evaluation of interpretability methods with ground truth arXiv preprint arXiv:1907.09701

[34] Xu K. et al. (2018), Structured adversarial attack: Towards general implementation and better interpretability, arXiv preprint arXiv:1808.01664

[35] Zhang X. et al. (2018), Interpretable Deep Learning under Fire, arXiv preprint arXiv:1812.00891,2018

[36] Ghorbani A.,Abubakar A., Zou J. (2019), Interpretation of neural networks is fragile, Proceedings of the AAAI Conference on Artificial Intelligence, 2019

[37] Dombrowski A.-K. et al. (2019), Explanations can be manipulated and geometry is to blame, arXiv preprint arXiv:1906.07983 2019