Czech Technical University in Prague
Faculty of Electrical Engineering
Department of Computer Science

Semestral Project

Adversarial Machine Learning for Detecting Malicious Behavior in Network Security

Bc. Michal Najman

Supervised by
Mgr. Viliam Lisý, MSc., Ph.D.

Submitted in May, 2019

Abstract

In this thesis, we elaborate on image captioning concerning especially dense image captioning. We present technical fundamentals of a model striving to solve such a task. Concretely, a detailed structure of DenseCap and Neural Image Caption is discussed. Experimentally, we examine results of DenseCap and analyse the model's weaknesses. We show that 92% of the generated captions are identical to a caption in the training set while the quality of those and the novel ones remains the same. We propose a criterion that significantly reduces a set of captions addressing an image whilst SPICE score of the set is maintained.

Keywords: image captioning, dense captioning, convolutional neural networks, long short-term memory

Abstrakt

Tato bakalářská práce se zaměřuje na automatickou tvorbu popisu obrázků (angl. image captioning), konkrétně na tzv. dense captioning. Problematika je ukázána ve světle současných modelů se zaměřením na stavbu DenseCap a Neural Image Caption. DenseCap zejména je prodoben experimentům, díky nimž jsou identifikovány nedostatky modelu. Pokusy ukazují, že 92 % generovaných popisků je identických vzorkům v trénovací množině. Je zjištěno, že jejich kvalita v porovnání s těmi, které v trénovací množině nejsou, je stejná. V neposlední řadě je navrženo kritérium, pomocí něhož lze významně zmenšit množinu popisků vztahujících se ke konkrétnímu obrázku, kdy SPICE skóre této menší množiny zůstává stejné.

Klíčová slova: automatická tvorba popisu obrázků, dense captioning, konvoluční neuronové sítě, long short-term memory

Český název: Automatická tvorba popisu obrázku pomocí konvolučních neuronových sítí

Author statement for undergraduate thesis:

I declare that the presented work was developed independently and that I have listed all sources of information used within it in accordance with the methodical instructions for observing the ethical principles in the preparation of university theses.

Prague, May 8, 2019

Michal Najman

Acknowledgements

I gratefully thank my supervisor Dr. Juho Kannala for his wise and critical comments as well as for his enriching attitude. Kiitos! Also, my appreciation goes to prof. Ing. Jiří Matas, Ph.D. who joined the thesis meetings and contributed immensely with novel ideas.

Last but not least, I thank my family for their support and my girlfriend Barbora for selecting the most pleasant shade of orange advisedly and for reviewing this text thoroughly.

Contents

1 Introduction

In recent years, computer science has seen an advent of powerful algorithms that are able to learn from examples. Despite the notion of learnable algorithms was recognised and studied in pioneering times of the field already, its wide-range real-world applications were to be implemented only with the presence of big available data collections and vast memory and computational resources. Therefore, nowadays one meets the abundance of machine learning techniques used to solve various problems. The field spans from theoretical research to practical applications in areas such as medical diagnosis, financial predictions and, most importantly in case of this work, computer security.

Most of the applications coin a similar scenario: a problem is formalised following a standard machine learning paradigm; a vast data set is collected and a proper algorithm giving the best results is found forming a model of the problem. However, in some applications once such a model is deployed to a complex real-world environment, one soon identifies the model performance deteriorates due to the key aspects of the reality that have been omitted in the standard machine learning point of view.

An example of such an observation is seen in computer vision. It was found that deep neural networks that reign competitions in image classification [1] are prone to so called adversarial images [2]. In particular, the state-of-the-art image classifiers based on deep neural networks score very well in terms of prediction accuracy, when given genuine images. However, such a classifier can be fooled with an image that was purposely adjusted. To put it simply, what is seen as a unambiguous cat by a human observer can be confidently labelled as a dog by a classifier. For instance, this phenomenon challenges traffic sign classification used in autonomous vehicles because it has been shown that a few well-placed stickers are able to fool the classifier and make it mis-recognise a yield sign for a main road sign [3].

To reflect such weakness, problems are reframed to a game-theoretic setting in which two autonomous rational players compete while following their mutually conflicting objectives. The aforementioned example with images is, consequently, extended in the following way. One of the players acts as an image classifier and aims to maximise classification accuracy, whereas the other player, an adversary, perturbs the images to lower prediction confidence or, better, to make the classifier misclassify the image.

Of course, the same is seen in computer security–the field defined by adversarial nature. Intruders desire to obfuscate a detector by adjusting their attacks [4]; malware is developed by optimising an executable binary [5], and spams are improved statistically to avoid detection [6].

The task central to this work is the problem of user classification in the adversarial setting. First, we examine the task as an instance of the user classification problem viewed by standard machine learning paradigm. Consequently, we show that omitting the adversarial nature in this particular problem exposes critical weaknesses, thus we

extend the model to incorporate game-theoretic notions. As an instance of the user classification task, we consider a detector of malicious users that is deployed by a computer security company to see which users exploit their public API service.

2 Background

2.1 Risk Minimisation

A classifier $h \in \mathcal{H}$ is a mapping $h : \mathbb{X} \mapsto \mathbb{C}$ that determines which class $c \in \mathbb{C}$ a sample $x \in \mathbb{X}$ belongs to. For the purposes of this work, we only describe binary classification in the following pages, however, the task is, naturally, expandable to a general discrete set $\mathbb{C}$. In the classical risk theory, the classifier h is a subject to minimisation of expected risk $R(h)$ given a cost function $\ell : \mathbb{C} \times \mathbb{C} \mapsto \mathbb{R}$.

$$R(h) = \mathop{\mathbb{E}}_{(x,c)\sim p} [\,[\ell(h(x), c)\,]] \tag{1}$$

Formally, the Expected Risk Minimisation (ERM) is given by:

$$\min_{h\in\mathcal{H}} R(h) \tag{2}$$

Typically when working with binary classification, ℓ is consider a *1-0 loss* which assigns an equal cost of magnitude *1* for misclassifying objects. The expected risk in this case accounts only for the rate of false positives and false negatives. If we employ *1-0 loss* into the expected risk, we arrive at the following form:

$$R(h) = \sum_{c\in\mathbb{C}} p(c) \int_{x:h(x)\neq \mathsf{c}} p(x|c)\, dx \tag{3}$$

The integral can be considered a probability of classifying objects x to an incorrect class given a correct class c, ie. $h(x) \neq c$. Let us consider binary classification in which $\mathbb{C} = \{\mathsf{B}, \mathsf{M}\}$ where M stands for a positive class (m for a malicious class) and B for a negative class (b for a benign class). In the context of this work, the positive class refers to malicious activity, ie. activity that is desired to be uncovered, and the negative class covers benign, legitimate or normal behaviours. To conclude, the risk $R(h)$ can be rewritten as a mixture of two types of errors: the probability of false positives and the probability of false negatives.

$$R(h) = p(\mathsf{B}) \cdot p(\mathsf{M}|\mathsf{B}) + p(\mathsf{M}) \cdot p(\mathsf{B}|\mathsf{M}) \tag{4}$$

In practice, the probabilities are not known and, moreover, computing the expected risk often involves intractable integrals. Therefore, the risk is empirically estimated from observed samples. The empirical risk $\hat{R}(h)$ estimated from a set of training samples $T_m = \{(x_i, c_i)\}_{i=1}^m$ is defined as follows:

$$\hat{R}_{T_m}(h) = \frac{1}{m} \sum_{(x_i,c_i)\in T_m} L(h(x_i), c_i) \tag{5}$$

Vapnik [7] showed that with increasing m the empirical risk $\hat{R}_{T_m}(h)$ approaches $R(h)$.

2.2 Regularisation

When examining possible classifiers, we usually have a priori knowledge of certain classifier instances being more suitable than others. Hence, some classifiers h correspond to models that are more likely to be inadequate, and some are a priori preferred. The reasons may vary, but mostly one desires to decrease models complexity to avoid overfitting. To capture this knowledge, a regularisation term $\Omega_D : \mathcal{H} \mapsto \mathbb{R}$ penalising some classifiers h is often added to the risk.

2.3 Neyman-Pearson Task

The Neyman-Pearson Task is a problem in which the false negative rate (FNR) is minimised while the false positive rate (FPR) is maintained lower than a given threshold.

$$\min_{f \in \mathcal{F}} \mathsf{FNR}(f) \qquad \text{s.t.} \quad \mathsf{FPR}(f) \tag{6}$$

3 Related Work

Examining adversarial aspects of various machine learning problems has currently been a popular topic. Mainly, this was triggered by [2] who showed that neural networks are susceptible to adversarial examples. Since then many endeavours have been carried out to enhance neural networks by making them robust. Some tried to develop a provably robust classifier [8], while others reframed the classification problem to incorporate aspects of game theory [9]. Despite most of the work deals with image classification, efforts to utilise the same notions in computer security have been seen too [5]. Susceptibility to adversarial examples is, however, not the only weakness adversaries exploit, they also are able to modify future training datasets in their favour [10].

3.1 Adversarial setting

[6] explore obfuscate strategies yielding spams that circumvent a spam filter. The authors consider attacks which are based on adding words to a spammy e-mail, while other modifications are not allowed. Three pools of words are defined: in the first obfuscate, random words from a dictionary are drawn; the second obfuscate utilises common legitimate e-mail words; and in the third obfuscate, words that are likely to appear in legitimate e-mails but are uncommon in spams are added.

To select the final set of words with the greatest effect from one of the three word pools, a black box threat model is used. In particular, the attacker repeatedly calls the detector to identify words which make the detector label the spam as benign. As expected, the last pool of words mentioned outperforms the others. Moreover, this shows that additive changes to a malicious object are sufficient for obfuscating the detector (within this domain). The authors claim they are able to add words to spams in such a way the tested detection models do not detect 50% of them.

To reflect the successful obfuscate algorithm, a defense strategy is proposed. It is shown that a robust detector which uncovers the adjusted spammy e-mails can be obtained by simply retraining the model on data now containing the attacks. However, the authors comment, a repeated obfuscate with a new set of effective words may again defeat the detector.

A similar notion is seen in more advanced classification models. For instance, deep neural networks are a popular class of classifiers nowadays for their performance in a great range of fields. They were shown to outperform other methods in image classification (ImageNet Challenge [1]), natural language processing [11] and in many others fields. However, it was found that neural networks are susceptible to artificially crafted images. In particular, [2] show an adversarial example may be labeled as an arbitrary class when accordingly adjusted. Moreover, despite the transformation of an input image is substantially bounded, for example by l_∞ norm, classifiers based on neural networks are prone to be circumvented anyway [3]. The susceptibility to adversarial samples follows the same observation in spam filtering – a good classifier is not

necessarily robust to test time data manipulation.

As soon as it was recognised the neural networks contain built-in vulnerabilities which are exploitable, endeavours to improve the architecture were carried out. To address the weakness, some of the following work focus on a model definition and consider possible attacks already in the model design. This approach is summarised by [12] who study adversarial examples in image classification. The authors identify that Empirical Risk Minimisation (ERM) does not necessarily give models robust to adversarially crafted samples.

Their work extends the training framework based on ERM by a threat model in which each data point $x \in \mathbb{R}^d$ is assigned a set of perturbations $S(x) \subseteq \mathbb{R}^d$ that is available to the adversary. The authors work with $S_\epsilon(x)$ that contains perturbations bounded by l_∞, creating an ϵ-hyper-cube around each x:

$$S_\epsilon(x) = \{x' \in \mathbb{R}^d \mid l_\infty(x - x') \leq \epsilon\} \tag{7}$$

The norm l_∞ is used for simplicity and roughly represents human-undetectable image perturbations. Other approaches, however, consider more complex bounds that capture domain-specific constraints [13].

To fully relate to an adversarial setting, [12] propose that the adversary maximises the classifier's loss function L by modifying an image x to an adversarial example $x' \in S_\epsilon(x)$. This is further incorporated into the ERM framework, arriving at a saddle point problem:

$$\min_{h \in \mathcal{H}} \mathbb{E}_{(x,c)\sim p}[\,[\,]. \max_{x' \in S_\epsilon(x)} L(h(x'),\, c)[\,]\,]. \tag{8}$$

In other words, a solution to the problem gives an optimal robust classifier $h \in \mathcal{H}$ that is likely to classify all objects $x \in \mathbb{R}^d$ and their neighbourhood $S_\epsilon(x)$ correctly.

The saddle point problem given above consists of two sub-problems: training the neural network and performing the inner maximisation. [12] approach the training part with Stochastic Gradient Descent (SGD) as it is commonly done in neural networks, while solving the inner maximisation task with Projected Gradient Descent (PGD) [14]. They conclude the ERM framework extended by this specific threat model gives a training method that is able to train neural networks in the adversarial setting and to produce classifiers robust to l_∞ bounded image perturbations. In addition, they find lower error is obtained with higher capacity models, suggesting that a robust model requires more parameters (eg. layers in neural networks).

To address the susceptibility to adversaries, several proposals of neural networks enhancements were submitted at ICLR 2018. However, seven out of nine were shown to be flawed due to following a similar ineffective scheme of masking the gradients [15].

In their paper, Athalye et al. [15] suggest there are three groups of gradient masking: first, a non-differentiable layer is inserted between the network layers; second, a classifier randomises its outputs; and third, a function transforms the input in such

a way backward gradient explodes or vanishes. Showing that the submitted defensive methods follow the schemes, the authors succeeded in circumventing 7 of 9 proposed models. Concretely, they replaced or removed defensive non-differentiable components accordingly to estimate the gradient and crafted adversarial samples with PGD.

3.2 Provable robustness

Until now, all presented efforts to improve the neural networks susceptibility were approached empirically and usually without providing provable defenses [12]. A method that aims to give provable resistance to adversarial samples was proposed by Kotler et al. [8] who examine a novel network architecture that provably classifies all objects in a convex neighbourhood of a given image correctly. To achieve that, Kotler et al. [8] redefine a ReLU [16] in such a way it is not a function anymore but rather a set of linear constrains yielding a convex polytope; i.e. a ReLU $y = \max\{0, x\}$ becomes:

$$y \geq x, \tag{9}$$

$$y \geq 0, \tag{10}$$

$$y(l - u) \leq -ux + ul \tag{11}$$

where u and l are an upper, respectively lower bound of x. The bounds are unknown and need to be estimated for each ReLU.

With a convex relaxation of ReLU, image classification can be rewritten as a linear program with all components of the network now being linear. In the training process, the weights of the relaxed neural network are optimised so that the network correctly classifies not only the input image but also its convex embedding. More specifically, using a l_∞ norm a ϵ-neighbourhood of an input sample is embedded by a convex polytope and the network learns to disallow any adversarial samples in it.

Solving the optimisation problem in its LP form with a standard LP solver is not tractable due to a great number of variables needed to express state-of-the-art deep neural networks. However, the LP can be conveniently used to form an upper bound on robust classification accuracy. Now, this upper bound combined with the ReLU input bounds estimation becomes fully differentiable. The training process follows standard SGD and gives a robust classifier that allows provably at most 6% error on MNIST. In contrast, a classical neural architecture is vulnerable up to 80% error [8].

3.3 Optimising malware

In contrast to image classification, the space of inputs is usually discrete in computer security. An image can be represented as a vector in $[0, 1]^n$, while executable binaries span a very sparse subset of the binary space $\{0, 1\}^n$. Similarly, a set of executable

source codes in a given programming language is a sparse subset of all character strings. Despite the theoretical difficulties several papers address the issue. [4] propose an obfuscate that optimises a malicious source code by applying some of the predefined modifications. The obfuscate method utilises the classifier's gradient to choose the most appropriate code modification. The set of plausible modifications is given beforehand and allows only additive changes. Although this significantly limits the attacker's action space, the authors claim reaching misclassification rates of up to 69%. [17] focus on static portable executables which they encode into a binary feature indicator vectors. Again, additive modifications are allowed only and malware is optimised with a bit gradient ascent. [5] take a different approach to malware optimisation and propose an agent which is trained with reinforcement learning. The agent is given a portable executable and its goal is to choose the most suitable modification of a piece of malware to lower probability of detection.

3.4 Stackelberg Prediction Game

As already shown, the problem of adversarial samples can be modelled as a game of two actors. However, [9] propose a more general game model compared to those already mentioned. In particular, the authors define the players as a classifier and a data generator consisting of *all* actors generating data – that is the second player aggregately covers both benign and malicious actors.

This setting is explored using a game-theoretical point of view. The authors propose a Stackelberg Prediction Game in which a classifier, acting as a leader, and a data generator, acting as a follower, optimise their action to meet their objectives. They argue the Stackelberg equilibrium is the most appropriate concept for trainable models, specifically compared to the Nash equilibrium. It is so, they claim, mainly because once a model is finalised and deployed, it is not changed anymore and thus the attacker can potentially learn all details of the model and adjust its actions to it.

In other words, the actions – the choice of model parameters and the test time data generation – are not carried out simultaneously, but instead the classifier commits to a specific parameters vector and the attacker utilises the information about the model and adjusts its attacking strategy accordingly. The later is modelled by a distribution shift at test time. The data generator transforms a probability of data p to a test time data probability $\dot{p}$ which maximises its objective function. In addition, the authors show that linear and kernel-based models together with suitable objective functions allow reformulating the problem to a quadratic program which yields the optimal model parameters.

3.5 Dataset poisoning

The Stackelberg Prediction Game assumes the model is fixed after deployment. In practice, however, engineers retrain the model on newly obtained data that might

better represent their population. As this might be done periodically, the adversary shall take advantage of it and adjust its obfuscate strategy. Concretely, Rubinstein et al. [10] elaborate on poisoning anomaly detectors.

The poisoning obfuscate consists of purposely providing pre-crafted samples to the detector over a long period of time in belief, that the samples will create a blind spot in which all samples are considered benign by the detector. The authors assume that the input space is usually governed by a distribution of benign samples concentrated only in certain areas, leaving the rest for anomalous activity. Given a substantial amount of time, the adversary is gradually able to poison the detector by targeting the large empty parts of the input space and populating them with benign samples. In future retraining, the anomaly detector may mistakenly consider those re-populated areas a new phenomenon and label them benign. The attacker then simply crafts an obfuscate near to the poisoned areas of the input space.

The authors present that such an obfuscate is possible with an anomaly detector based on principal component analysis (PCA) which determines directions of the sample space with greatest variance. Replacing variance in PCA with median absolute deviation, which in contrary is a robust scale estimator, their model is robust to data set poisoning and successfully performs anomaly detection in backbone networks.

4 Problem Analysis

What is a proper name for this section?

In the present state of Internet, it is common for a site owner to run models classifying users or their behaviour. The task spans from user's interests specification to detecting deviating activity. Since such applications are becoming more popular, one may expect the users to modify their behaviours once they know they are being tracked and classified. Moreover, behaviour modification may very well be of rational nature, especially when a malicious user exploits loopholes or carries out lawless activity in order to pursuit its goal.

In other words, if there is a cost for being disclosed or seen as a certain category, the users will examine their actions to optimise for lower cost. As a result, machine learning models of any kind aiming to capture behaviours of those users necessarily need to have the adversary nature incorporated in their design.

The straight-forward approach of solving this task would be to collect many examples of both kinds of user activity; that is to asses a dataset containing well-represented both malicious and benign users. This approach would follow the standard ERM framework and would give an activity classifier that minimises expected risk but omits the adversarial nature. However, one might arrive at difficulties during the construction of a balanced dataset for there is usually very few records of malicious activity, disproportionally less than the collection of normal, benign users. Also, and more importantly, the malicious actors modify their attack vectors once their method is exposed or they discover details concerning the detector.

Taking that into account, we consider the setting as a game of a classifier competing with a body of malicious users. This approach necessarily modifies the ERM framework and enhances it with game-theoretic notions.

This section first discusses the use-case which motivates this work. Then, a suitable threat model is prosed and the game is formally defined, supplemented with reasoning for given choices.

Emphasise the general adversarial machine learning problem as the core of this section.

revisit the section introduciton to inform about all subsecitons and refer to them

A malicious activity detection system is essentially a classifier that classifies users based on their behaviour. This in principle is a machine learning problem of finding a classifier $f \in \mathcal{F}$ minimising expectation of detection loss ℓ_{-1}. The detector is a mapping $f : \mathcal{X} \mapsto \mathcal{C}$ which takes vectors $x \; in \mathcal{X}$ on its input and produces a decision $d \in \mathcal{C}$. However, the ground variable representing a discrete input object is a user's activity history $h \in \mathcal{H}$ which is translated to a corresponding feature vector with a feature map $\Phi : \mathcal{H} \mapsto \mathcal{X}$.

All variables and functions related strictly to a detector are subscripted with -1, whereas we use $+1$ in the attacker's case. This choice follows Brückner et al [9].

As already mentioned, minimising the expected risk – as it is done in general classification problems – does not help to solve the problem of detection. The expected risk minimisation framework (ERM) consists of identifying an optimal classifier f that minimises expectation of $\ell_{\text{-}1} : \mathcal{C} \times \mathcal{C} \mapsto \mathbb{R}$ over the set $\mathcal{X} \times \mathcal{C}$.

The expected risk can be formulated as a convex combination of risks conditioned on a class.Assuming there is two classes, i.e. $\mathcal{C} = \{\mathsf{B}, \mathsf{M}\}$, the expected risk can be rewritten as a combination of the risk attained on the malicious class and the risk attained on the benign class.

Definition 4.1. *Let the risk attained on the malicious a class* M*,* $R_{\text{-}1}(f \,|\, \mathsf{M})$*, be the expectation of the loss conditioned on class* M*. Let the risk attained on the benign class* B*,* $R_{\text{-}1}(f \,|\, \mathsf{B})$*, be the expectation of the loss conditioned on a class* B*.*

$$R_{\text{-}1}(f \,|\, \mathsf{M}) = \mathop{\mathbb{E}}_{x}[\, \ell_{\text{-}1}(f(x), \mathsf{M}) \,|\, \mathsf{M}\,] \tag{12}$$

$$R_{\text{-}1}(f \,|\, \mathsf{B}) = \mathop{\mathbb{E}}_{x}[\, \ell_{\text{-}1}(f(x), \mathsf{M}) \,|\, \mathsf{B}\,] \tag{13}$$

Definition 4.2 (Detector's Expected Risk Minimisation)**.** *In standard classification, the optimal classifier* f^* *is the solution of the following problem:*

$$\underset{f \in \mathcal{F}}{\textit{minimise}} \quad p(\mathsf{B}) \cdot R_{\text{-}1}(f \,|\, \mathsf{B}) + p(\mathsf{M}) \cdot R_{\text{-}1}(f \,|\, \mathsf{M}) \tag{14}$$

4.1 Specifics of Adversarial Machine Learning

In this section, we examine adversarial machine learning in the domain of network security in general terms. The central task is to detect malicious users in the network without ideally affecting legitimate users.

The expectations in Def. 4.2 are usually estimated from a set of examples of each class. However, in the detection problem there are not enough examples of malicious behaviour and, in addition, this behaviour changes reflecting the current detector. This imposes two critical properties of adversarial machine learning:

- The priori class probabilities are not known.
- An individual attacker follows its private objective and (possibly rationally) chooses actions minimising its cost.

4.2 Property 1: Unknown Class Probabilities

To reflect the first property, we redefine the detection problem to comply with the Neyman-Pearson Task 2.3. Since we aim to detect malicious activity, the false positives comprise benign users classified as malicious. And, vice-versa, the false negatives are malicious users classified as benign.

Definition 4.3 (Neyman-Pearson Task)**.** *The Neyman-Pearson Task translates to minimising the expected loss conditioned on the malicious class while the expected loss conditioned on the benign class is maintained lower than a threshold.*

$$\begin{aligned} \underset{f \in \mathcal{F}}{\textit{minimise}} \quad & R_{\text{-}1}(f \mid \mathsf{M}) \\ \textit{subject to} \quad & R_{\text{-}1}(f \mid \mathsf{B}) \leq \tau_0 \end{aligned} \tag{15}$$

Using this formulation, prior class probabilities are omitted and, in addition, the task reflects the nature of security detection problems in which there is a hard constraint on false positives. In other words, we aim to find a classifier f that does not affect legitimate activity but given this constraint is the best detector of malicious activity. As shown later, this form is particularly useful if the detector is a neural network.

4.3 Property 2: Adversarial Setting

By assuming an attacker is a rational actor that pursuits its goal, the setting of statistical learning changes to an adversarial game of two players: a detector and an attacker.

We sort sampled activity into two classes: benign (B) and malicious (M). The former is activity generated by legitimate users and the later is activity produced solely by attackers in pursuit of their objectives.

To avoid detection, each individual attacker obfuscates its primary activity by acting nearly as a legitimate user. However, if a good detector is deployed the obfuscation requires a large quantity of legitimate activity. If the obfuscation effort is too costly the attacker may seize to attack at all.

The obfuscated activity of an attacker, in consequence, is recorder by the detector and stored as an activity history based on which the detector assigns a label to it. The intuitive goal of the detector is to label legitimate activity history as benign, i.e. B, and the activity history generated by an attacker as malicious, i.e. M.

In the following sections, we closely explore and examine the motivations and aspects implied by the adversarial setting in network security.

4.3.1 Stackelberg Game

In practice, the detector is fixed after deployment and the choice of its particular form and parameters necessarily occurs before the deployment. This is a case of a Stackelberg game [9] in which the detector is a leader and the attacker is a follower. For the sake of simplicity we assume the Strong Stackelberg Equilibrium is played.

In a Stackelberg game, the follower plays a best response to the leader's public strategy and the leader optimises this strategy accounting the follower's best response. In such a setting, the leader optimally plays a mixed strategy.

This means, the attacker obfuscates its activity optimally without a need of randomisation by playing a best response to detector's strategy.

As mentioned, the detector may necessarily randomise its actions to achieve the optimal cost. This translates to a detector playing a mixed strategy $\sigma(f) : \mathcal{F} \mapsto [0, 1]$ instead of a single particular f.

Definition 4.4. *A stochastic detector $D : \mathcal{X} \mapsto \mathcal{C}$ is a probability distribution $p(d \mid x)$ generating a decision $d \in \mathcal{C}$ conditioned on an observed sample $x \in \mathcal{X}$. D_σ is given by a detector playing a mixed strategy $\sigma : \mathcal{F} \mapsto [0, 1]$:*

$$D_\sigma(d|x) = \sum_{f:f(x)=d} \sigma(f) \tag{16}$$

4.3.2 Information Available to Attacker

We assume the attacker has full knowledge of the detector's structure and parameters.

4.3.3 Attacker

We model the attacker as a rational actor which plays the action minimising its expected costs. The particular form of costs and actions depends largely on the domain. Therefore, here we only present general notions defining the attacker and, later in Section 5.3, we propose the attacker's model that suits the running example of attacks to a URL reputation service.

We propose all attackers follow the same objective and they differ only in their particular primary goal. That is, the activity obfuscation is practically the same task shared by all attackers and two attackers differ in what they aim to obfuscate.

For that reason, the model of an attacker considers a common body of attacker instances in which an individual attacker instance is thoroughly defined by its primary goal $g \in \mathcal{G}$. The common shared obfuscation function $\psi : \mathcal{G} \mapsto \mathcal{H}$ takes a primary goal g on its input and maps it to an activity history $h = \psi(g)$ that obfuscates the primary goal.

obfs function maps to S(g), maybe make definition of psi

Definition 4.5. *The attacker's risk $R_{+1} : \Psi \times \mathcal{F} \mapsto \mathbb{R}$ is given as the expectation of its loss $\ell_{+1} : \mathcal{G} \times \Psi \times \mathcal{C} \mapsto \mathbb{R}$. That is:*

$$R_{+1}(\psi, f) = \mathop{\mathbb{E}}_{g}[\, \ell_{+1}(g, \psi, f(\Phi(\psi(g))))\,] \tag{17}$$

couple of words what the loss comprises.

Relating to game theory, the obfuscation function ψ is an attacker's action and its best response to σ is given by minimising the attacker's expected risk $\mathbb{E}_{f\sim\sigma}\, R_{+1}(\psi, f)$.

Proposition 4.1 (Attacker's best response)**.** *The attacker's best response* $\boldsymbol{BR}(\sigma)$ *to a mixed strategy* σ *is a set of obfuscation functions* $\psi : \mathcal{G} \mapsto \mathcal{H}$ *that are the minimisers of the expectation of the attacker's loss* ℓ_{+1}*.*

$$\boldsymbol{BR}(\sigma) = \underset{\psi}{\mathit{argmin}}\, \mathop{\mathbb{E}}_{g,d}[\, \ell_{+1}(g, \psi, d)\,] \tag{18}$$

Proof.

$$\begin{aligned}
\mathbf{BR}(\sigma) &= \operatorname*{argmin}_{\psi} \mathop{\mathbb{E}}_{f\sim\sigma} R_{+1}(\psi, f) && (19)\\
&= \operatorname*{argmin}_{\psi} \mathop{\mathbb{E}}_{f,g}[\, \ell_{+1}(g, \psi, f(\Phi(\psi(g))))\,] && (20)\\
&= \operatorname*{argmin}_{\psi} \sum_{f}\sum_{g} \ell_{+1}(g, \psi, f \circ \Phi \circ \psi(g)) \cdot p(g) \cdot \sigma(f) && (21)\\
&= \operatorname*{argmin}_{\psi} \sum_{g}\sum_{d} \sum_{f:f\circ\Phi\circ\psi(g)=d} \ell_{+1}(g, \psi, d) \cdot p(g)\sigma(f) && (22)\\
&= \operatorname*{argmin}_{\psi} \sum_{g}\sum_{d} \ell_{+1}(g, \psi, d) \cdot p(g) \cdot \sum_{f:f\circ\Phi\circ\psi(g)=d} \sigma(f) && (23)\\
&= \operatorname*{argmin}_{\psi} \sum_{g}\sum_{d} \ell_{+1}(g, \psi, d) \cdot p(g) \cdot D_{\sigma}(d \,|\, \Phi \circ \psi(g)) && (24)\\
&= \operatorname*{argmin}_{\psi} \mathop{\mathbb{E}}_{g,d}[\, \ell_{+1}(g, \psi, d)\,] && (25)
\end{aligned}$$

□

4.3.4 Stochastic Detector

The detector's pure strategy consists of a particular detector f. However, as proposed above, its optimal strategy is generally mixed and the detector, therefore, randomises its final decision $d \in \mathcal{C}$.

A mixed strategy in case of the detector is a probability distribution $\sigma : \mathcal{F} \mapsto [0, 1]$ which assigns a probability to each particular detector f. The decision d representing the estimated class of a sample x is, consequently, a random variable whose probability distribution is the aggregate of probabilities $\sigma(f)$ for which $f(x) = d$. To capture that, we defined a decision distribution $D(d|x)$ in Def. 4.4.

In this work, we model $D(d|x)$ with a neural network which fruitfully allow us to bypass potentially infinite enumeration of detectors from $\mathcal{F}$. The detector's mixed strategy is, in conclusion, represented by the distribution $D_\theta(d|x)$ where θ is a parameters vector.

As proposed by Brückner et al. [9], the attacker's impact on the setting can be modelled by a distribution shift. However, in contrast to [9], in this work we assume only the malicious class activity is governed by adversarial objectives and benign activity is maintained unchanged irrespective to the detector's presence. Taking that into account,

we define that the distribution of samples produced by attackers $p(x \mid \mathsf{M})$ is shifted in reaction to the presence of a deployed detector D and changes to $\dot{p}(x \mid \mathsf{M}, D)$.

In a standard classification problem, we find f minimising the expected risk. In our adversarial setting, the detector is necessarily a distribution D that is a solution to the Neyman-Pearson Task with a non-stationary distribution of samples $\dot{p}(x \mid \mathsf{M})$.

Proposition 4.2. *Let the attacker play a best response $\boldsymbol{BR}(\sigma)$ to a mixed strategy σ, then the detector's risk of a mixed strategy σ attained on malicious activity, $R_{\text{-}1}(\sigma \mid \mathsf{M})$, is given by the best-case expectation of its loss attained on malicious activity.*

$$R_{\text{-}1}(\sigma \mid \mathsf{M}) = \mathop{\mathbb{E}}_{f}[\, R_{\text{-}1}(f) \mid \mathsf{M}\,] = \min_{\psi \in \boldsymbol{BR}(\sigma)} \mathop{\mathbb{E}}_{q,d}[\, \ell_{\text{-}1}(d, \mathsf{M}) \mid \mathsf{M}\,] \tag{26}$$

Similarly, the detector's risk of mixed strategy σ attained on benign activity, $R_{\text{-}1}(\sigma \mid \mathsf{B})$, is given by the expectation of its loss attained on benign activity.

$$R_{\text{-}1}(\sigma \mid \mathsf{B}) = \mathop{\mathbb{E}}_{f}[\, R_{\text{-}1}(f) \mid \mathsf{B}\,] = \mathop{\mathbb{E}}_{h,d}[\, \ell_{\text{-}1}(d, \mathsf{B}) \mid \mathsf{B}\,] \tag{27}$$

Proof. For the risk of a mixed strategy attained on malicious activity, it holds that:

$$\begin{aligned} R_{\text{-}1}(\sigma \mid \mathsf{M}) &= \mathop{\mathbb{E}}_{f}[\, R_{\text{-}1}(f) \mid \mathsf{M}\,] && (28)\\ &= \mathop{\mathbb{E}}_{f,x}[\, \ell_{\text{-}1}(f(x), \mathsf{M}) \mid \mathsf{M}\,] && (29)\\ &= \sum_{f} \sum_{x} \ell_{\text{-}1}(f(x), \mathsf{M}) \cdot \dot{p}(x \mid \mathsf{M}) \cdot \sigma(f) && (30) \end{aligned}$$

Consider a sample x is generated solely by the attacker (due to the M class in the conditional probability). We substitute x for $\Phi \circ \psi(g)$. Assuming a feature map $\Phi : \mathcal{H} \mapsto \mathcal{X}$ projects each h to one particular feature vector x and a malicious activity history h is given by a primary goal g obfuscated by a best response obfuscation function $\psi \in \mathbf{BR}(\sigma)$, the sum of probabilities $p(g)$ for which $\Phi \circ \psi(g) = x$ gives the non-stationary probability $\dot{p}(x \mid \mathsf{M})$.

$$\begin{aligned} \dot{p}(x \mid \mathsf{M}) &= \sum_{h:\Phi(h)=x} \dot{p}(h \mid \mathsf{M}) && (31)\\ &= \sum_{h:\Phi(h)=x} \sum_{g:\psi(g)=h} p(g) && (32)\\ &= \sum_{g:\Phi\circ\psi(g)=x} p(g) && (33) \end{aligned}$$

Using the substitution and considering the best-case, we arrive at:

$$R_{\text{-}1}(\sigma \,|\, \mathsf{M}) = \sum_f \sum_x \ell_{\text{-}1}(f(x), \mathsf{M}) \cdot \dot{p}(x \,|\, \mathsf{M}) \cdot \sigma(f) \tag{34}$$

$$= \min_{\psi \in \mathbf{BR}(\sigma)} \sum_f \sum_g \ell_{\text{-}1}(f \circ \Phi \circ \psi(g), \mathsf{M}) \cdot p(g) \cdot \sigma(f) \tag{35}$$

$$= \min_{\psi \in \mathbf{BR}(\sigma)} \sum_g \sum_d \ell_{\text{-}1}(d, \mathsf{M}) \cdot p(g) \cdot \sum_{f : f \circ \Phi \circ \psi(g) = d} \cdot \sigma(f) \tag{36}$$

$$= \min_{\psi \in \mathbf{BR}(\sigma)} \sum_g \sum_d \ell_{\text{-}1}(d, \mathsf{M}) \cdot p(g) \cdot D_\sigma(d \,|\, \Phi \circ \psi(g)) \tag{37}$$

$$= \min_{\psi \in \mathbf{BR}(\sigma)} \mathop{\mathbb{E}}_{q,d} [\, \ell_{\text{-}1}(d, \mathsf{M}) \,|\, \mathsf{M} \,] \tag{38}$$

Similarly for the risk of a mixed strategy attained on benign activity:

$$R_{\text{-}1}(\sigma \,|\, \mathsf{B}) = \mathop{\mathbb{E}}_f [\, R_{\text{-}1}(f) \,|\, \mathsf{B} \,] \tag{39}$$

$$= \mathop{\mathbb{E}}_{f,x} [\, \ell_{\text{-}1}(f(x), \mathsf{B}) \,|\, \mathsf{B} \,] \tag{40}$$

$$= \sum_f \sum_x \ell_{\text{-}1}(f(x), \mathsf{B}) \cdot p(x \,|\, \mathsf{B}) \sigma(f) \tag{41}$$

$$= \sum_f \sum_h \ell_{\text{-}1}(f \circ \Phi(h)), \mathsf{B}) \cdot p(h \,|\, \mathsf{B}) \cdot \sigma(f) \tag{42}$$

$$= \sum_h \sum_d \ell_{\text{-}1}(d, \mathsf{B}) \cdot p(h \,|\, \mathsf{B}) \cdot D_\sigma(d \,|\, \Phi(h)) \tag{43}$$

$$= \mathop{\mathbb{E}}_{h,d} [\, \ell_{\text{-}1}(d, \mathsf{B}) \,|\, \mathsf{B} \,] \tag{44}$$

□

Definition 4.6. *For simplicity, we interchangeably use σ and D_σ and D_θ as the detector's strategy. Thus:*

$$R_{\text{-}1}(D_\sigma \,|\, \cdot) = R_{\text{-}1}(\sigma \,|\, \cdot) = R_{\text{-}1}(\theta \,|\, \cdot) \tag{45}$$

Proposition 4.3 (Detector's optimisation problem)**.** *Let the detector minimise the expected risk attained on malicious activity, while maintaining the expected risk attained on benign activity upper-bounded by τ_0. Let the attacker minimise its expected risk. Then the stochastic detector D_θ parametrised by θ and the obfuscation function ψ which are the solution to the following bi-level optimisation problem are the Stackelberg*

equilibrium.

$$\begin{aligned} \underset{\theta,\psi}{\textit{minimize}} \quad & \mathop{\mathbb{E}}_{q,d}[\,\ell_{\text{-}1}(d,\mathsf{M})\,|\,\mathsf{M}\,] \\ \textit{subject to} \quad & \mathop{\mathbb{E}}_{h,d}[\,\ell_{\text{-}1}(d,\mathsf{B})\,|\,\mathsf{B}\,] \leq \tau_0 \\ & \psi \in \underset{\psi'}{\textit{argmin}}\, \mathop{\mathbb{E}}_{g,d}[\,\ell_{+1}(g,\psi',d)\,] \end{aligned} \tag{46}$$

Proof. The proposition follows directly from the definitions and propositions above. □

This is cool as it follows directly from ERM when three assumptions are added: Neyman-Pearson, Rational Attacker, Stackelberg Game

Smooth out the sequence of those propositions by mix-in explanations. This is a key train thought and it is important to stress it out clearly.

4.4 Assumption on Losses

As it is common in ERM, we expect the detector's loss $\ell_{\text{-}1}$ is a zero-one loss. This simplifies the primary objective in the detector's optimisation problem.

Proposition 4.4. *Let the detector's loss $\ell_{\text{-}1}$ be a zero-one loss. Then the detector's risk attained on malicious activity $R_{\text{-}1}(\theta\,|\,\mathsf{M})$ is the expectation of the posteriori probability of a benign class conditioned on malicious activity. Similarly for the risk attained on benign activity $R_{\text{-}1}(\theta\,|\,\mathsf{M})$:*

$$R_{\text{-}1}(\theta\,|\,\mathsf{M}) = \min_{\psi\in\boldsymbol{BR}(\sigma)} \mathop{\mathbb{E}}_{g}[\,D_\theta(\mathsf{B}\,|\,\Phi\circ\psi(g))\,|\,\mathsf{M}\,] \tag{47}$$

$$R_{\text{-}1}(\theta\,|\,\mathsf{B}) = \mathop{\mathbb{E}}_{h}[\,D_\theta(\mathsf{M}\,|\,\Phi(h))\,|\,\mathsf{B}\,] \tag{48}$$

Proof. The proof is straight-forward.

$$R_{\text{-}1}(\theta\,|\,\mathsf{M}) = \min_{\psi\in\mathbf{BR}(\sigma)} \mathop{\mathbb{E}}_{q,d}[\,\ell_{\text{-}1}(d,\mathsf{M})\,|\,\mathsf{M}\,] \tag{49}$$

$$= \min_{\psi\in\mathbf{BR}(\sigma)} \mathop{\mathbb{E}}_{q}[\sum_{d} \ell_{\text{-}1}(d,\mathsf{M}) D_\theta(d\,|\,\Phi\circ\psi(q))\,|\,\mathsf{M}\,] \tag{50}$$

$$= \min_{\psi\in\mathbf{BR}(\sigma)} \mathop{\mathbb{E}}_{q}[\,D_\theta(\mathsf{B}\,|\,\Phi\circ\psi(q))\,|\,\mathsf{M}\,] \tag{51}$$

$$R_{\text{-}1}(\theta\,|\,\mathsf{B}) = \mathop{\mathbb{E}}_{h,d}[\,\ell_{\text{-}1}(d,\mathsf{B})\,|\,\mathsf{B}\,] \tag{52}$$

$$= \mathop{\mathbb{E}}_{h}[\sum_{d} \ell_{\text{-}1}(d,\mathsf{B}) D_\theta(d\,|\,\Phi(h))\,|\,\mathsf{B}\,] \tag{53}$$

$$= \mathop{\mathbb{E}}_{h}[\,D_\theta(\mathsf{M}\,|\,\Phi(h))\,|\,\mathsf{B}\,] \tag{54}$$

□

d	$\ell_{+1}(g, \psi, d)$
B	$\Omega_{+1}(g, \psi(g))$
M	$L_0 + \Omega_{+1}(g, \psi(g))$

Table 1: Table to test captions and labels

The posteriori probability of the stochastic detector $D_\theta(d \,|\, x)$ is explicitly modelled by neural network in this work. Thus we prefer the risk explicitly contains the term. However, this does not hold generally and in some cases it is more fruitful to estimate the risk as expectation of loss values (e.g. reinforcement learning).

The same trick which was used in case of the defender cannot by applied to the attacker. The attacker's loss $\ell_{+1} : \mathcal{G} \times \Psi \times \mathcal{C} \mapsto \mathbb{R}$ is more complex. Naturally, it consists of two components: a public and a private term. The public cost reflects the adversarial objective of escaping detection (e.g. detection probability). The private cost penalises the attacker for too costly obfuscation and is not necessarily adversarial to the detector's cost.

This also shows the game is a non-zero sum game as the private term in the attacker's loss does not have an adversarial equivalent in the detector's loss.

Following Brückner et al. [9], we defined the attacker as a shared body of attacker instances. However, if the attacker's loss is defined conveniently, the attacker's optimisation problem decomposes and it can be solved independently for each attacker's instance. The convenient form of the loss is shown Tab. 1.

The motivation of this particular form of the loss is simple. If an attacker is detected it pays the amount L_0 for acquiring a new license or an account so that it is able to carry out further activity. However, the more complex activity histories it creates to obfuscate its primary goal, the more costly carrying out such activity is. This is represented by $\Omega_{+1} : \mathcal{G} \times \mathcal{H} \mapsto \mathbb{R}$.

Recall that the obfuscation function $\psi(g)$ is constrained by the set of activity histories $S(g)$ such that $\psi(g) \in S(g)$, i.e. $\psi(g)$ can only create activity histories in $S(g)$. The set $S(g)$ in practice defines activity histories that the attacker is able to construct from g.

Proposition 4.5. *Let the attacker's loss be defined by Tab. 1. Let the attacker's private cost be a function $\Omega_{+1} : \mathcal{G} \times \mathcal{H} \mapsto \mathbb{R}$. Then the attacker's best response problem of finding the optimal obfuscation function ψ^* decomposes to identifying $\Psi^*(g)$ such that $\psi^*(g) \in \Psi^*(g) \subset S(g)$ and $\Psi^*(g)$ is the set of solution to the following problem.*

$$\Psi^*(g) = \underset{h \in S(g)}{\mathit{argmin}}\, L_0 \cdot D_\sigma(\mathsf{M} \,|\, \Phi(h)) + \Omega_{+1}(g, h) \tag{55}$$

Proof. Proposition 4.1 defines the attacker's best response problem in which the expectation is over variables g and d.

$$\psi^* \in \mathbf{BR}(\sigma) = \operatorname*{argmin}_{\psi} \mathop{\mathbb{E}}_{g,d}[\,\ell_{+1}(g,\psi,d)\,] \tag{56}$$

$$= \operatorname*{argmin}_{\psi} \mathop{\mathbb{E}}_{g} \mathop{\mathbb{E}}_{d}[\,\ell_{+1}(g,\psi,d)\,] \tag{57}$$

Let us substitute the loss ℓ_{+1} for its tabular form in Tab. 1. The inner expectation in Eq. 57 simplifies and becomes:

$$\begin{aligned}
\mathop{\mathbb{E}}_{d}[\,\ell_{+1}(g,\psi,d)\,] &= \Omega_{+1}(g,\psi(g)) \cdot D_\sigma(\mathsf{B} \,|\, \Phi \circ \psi(g)) + \\
&\quad + (L_0 + \Omega_{+1}(g,\psi(g))) \cdot D_\sigma(\mathsf{M} \,|\, \Phi \circ \psi(g)) \\
&= \Omega_{+1}(g,\psi(g)) \cdot (1 - D_\sigma(\mathsf{M} \,|\, \Phi \circ \psi(g))) + \\
&\quad + (L_0 + \Omega_{+1}(g,\psi(g))) \cdot D_\sigma(\mathsf{M} \,|\, \Phi \circ \psi(g)) \\
&= L_0 \cdot D_\sigma(\mathsf{M} \,|\, \Phi \circ \psi(g)) + \Omega_{+1}(g,\psi(g))
\end{aligned}$$

This gives us a simplified of the best response problem:

$$\operatorname*{argmin}_{\psi} \mathop{\mathbb{E}}_{g}[\, L_0 \cdot D_\sigma(\mathsf{M} \,|\, \Phi \circ \psi(g)) + \Omega_{+1}(g,\psi(g))\,]$$

The best response problem now contains only $\psi(g)$ and the expectation can be decomposed. The criterion is minimised if we set $\psi(g)$ to h that minimises $L_0 \cdot D_\sigma(\mathsf{M} \,|\, \Phi(h)) + \Omega_{+1}(g,h)$. However, the obfuscation function ψ is constrained by $S(g)$. Taking $S(g)$ into account, we arrive at the following form.

$$\psi^*(g) \in \Psi^*(g) = \operatorname*{argmin}_{h \in S(g)} L_0 \cdot D_\sigma(\mathsf{M} \,|\, \Phi(h)) + \Omega_{+1}(g,h)$$

$\Psi^*(g)$ simply denotes the solution of the optimisation problem.

□

Having

Proposition 4.6. *Considering the losses in Propositions 4.4 and 4.5, the detector's optimisation problems 46 becomes:*

$$\begin{aligned}
\underset{\theta}{\textit{maximise}} \quad & \mathop{\mathbb{E}}_{g}[\, \max_{h^{\mathsf{adv}} \in \Psi^*(g)} D_\theta(\mathsf{M} \,|\, \Phi(h^{\mathsf{adv}})) \,|\, \mathsf{M}\,] \\
\textit{subject to} \quad & \mathop{\mathbb{E}}_{h}[\, D_\theta(\mathsf{M} \,|\, \Phi(h)) \,|\, \mathsf{B}\,] \leq \tau_0 \\
& \Psi^*(g) = \operatorname*{argmin}_{h' \in S(g)} L_0 \cdot D_\sigma(\mathsf{M} \,|\, \Psi(h')) + \Omega_{+1}(g,h')
\end{aligned}$$

Proof. todo □

4.5 Anomaly Detection

Note the aforementioned problem is related to (unsupervised) anomaly detection in which is is assumed no information about the malicious class is known. Thus, formulating the problem as anomaly detection, we aim to identify a detector for which the expected loss conditioned on benign class is lower than a threshold. The anomaly detection view of the problem is utilised for example if a detector consists of a k nearest neighbours distance estimator.

For our purposes we define anomaly detection as ...

Definition 4.7 (Anomaly Detection)**.** *Let the optimal anomaly detector be a distribution $D_\theta(d|x)$, solely parametrised by a vector θ, if it is a solution to the following problem:*

$$\begin{aligned} &\textit{find} && \theta \\ &\textit{such that} && \mathop{\mathbb{E}}_{h}[\, D_\theta(\mathsf{M} \,|\, \Phi(h)) \,|\, \mathsf{B} \,] \leq \tau_0 \end{aligned} \tag{58}$$

5 Game Definition

The activity classification problem is modelled as a game of two players: a detector and an adversary. The goal of the detector is to identify the best user activity classifier, while the adversary seeks to optimally modify query historys of malicious users in such a way they get misclassified by the classifier.

More introductory words.

5.1 Use Case

write: See, this is a running example.

In this work, we consider a network security company that runs a reputation service which returns rating of a queried URL. For example, if we query the service's API with www.google.com, the url is rated with high score whereas the malicious URL www.malicious-url.com is rated poorly. This type of a service is usually deployed by network security companies to provide their security software with access to most up-to-date database of URL ratings.

The typical usage scenario is coined as follows. A client running on an end-user's device encounters the user is about to enter a website. To evaluate the danger level of the website, the client queries the API of the reputation service with the website URL. Accordingly, the client may show a warning message notifying the user of expected danger or carry out an appropriate action.

Usually, URL rating systems aim to identify various danger types of a URL. As a running example in this work, we focus one particular type of malicious misuse of the URL reputation system: malware producers that asses a set of URLs which are used as communication entry-points for deployed malware units. With one of these URLs, a unit of deployed malware is able to receive commands and adjust its actions. However, to maintain consistency and availability of its malware units, the malware producer must regularly check whether any of its URLs has been exposed – by querying the publicly available URL rating system.

To conclude, the task is as follows: the computer security company desires to distinguish malicious users of the URL rating service from benign ones based on the URLs each user queried the service with.

5.1.1 Formal Definition

In the this section, we formally define the running example of an attack to a reputation system.

The service is queried with a URL $u \in \mathbb{U}$. The query consists of a typical HTTP requests properties and the url as the subject of the query. The service securely assigns each query to a user based on a license the user used. Thus, we define an activity history $h \in \mathcal{H}$ as a collection of queries of the user. For example, if a user sends a sequence

of queries for which we record a queried URL, an arrival timestamp, a source IP or possibly other information, this is recorded and integrally stored in a corresponding user's query history h.

$$(u_1, t_1, \mathsf{IP}_1, \dots), (u_2, t_2, \mathsf{IP}_2, \dots), \dots, (u_k, t_k, \mathsf{IP}_k, \dots) \longrightarrow h \tag{59}$$

Note that a user's query history h represents the ground objects based on which the detector classifies users. Note that the inner structure of h is discrete and also $\mathcal{H}$ is generally discrete. This is problematic for attackers as there is no direct way of computing gradients with respect to elements of h.

In the previous section, we defined a malicious user posses a primary goal that thoroughly defines its individual instance. In this example a single malicious user posses a private set of primary URLs $U^{\mathsf{pr}} \subset \mathbb{U}$. The primary url set contains urls which the malicious user necessarily employs to achieve its primary goal. That is to obtain the current reputation rating for each URL in U^{pr}. In consequence the primary goal $g \in \mathcal{G}$ is composed solely of the primary URLs.

$$g = U^{\mathsf{pr}} \tag{60}$$

Given its primary URLs, a malicious user queries the service with URLs U that may next to its primary URLs also contain legitimate queries which it uses to obfuscate its activity.

$$U^{\mathsf{pr}} \subseteq U \tag{61}$$

Recall we assume the malicious user is a rational player thus the particular content of U changes depending on the classifier. If there was no classifier and, therefore, malicious users were not motivated to adjust their behaviour, they would presumably query the service with U resembling primary URLs and perhaps containing just a little overhead, ie. $U \cong U^{\mathsf{pr}}$. No detector also means there is no need to strategies with the values of other properties request properties. This activity would be recorder in a corresponding activity history h.

Nonetheless, once there actually is a classifier deployed, implying a cost for disclosure, the malicious users rationally query the service with additional legitimate URLs to obfuscate its primary goal. There are essentially to types of primary URLs U^{pr} obfuscation: adding legitimate queries and adjusting properties of each query. Each primary URLs set U^{pr} induces a bounded set of histories $S(U^{\mathsf{pr}})$ that contains histories derivable from U^{pr} by obfuscation.

$$S(U^{\mathsf{pr}}) = \{h \in \mathcal{H} \,|\, U^{\mathsf{pr}} \subseteq \mathsf{urls\ in}\ h\} \tag{62}$$

We capture this with the obfuscation function $\psi : \mathcal{G} \mapsto \mathcal{H}$ which a malicious user employs to transform its original primary goal g to an obfuscated activity history. Since a primary goal g is solely defined by a primary URLs set U^{pr}, we can redefine

the obfuscation function for this use case to: $\psi : 2^{\mathbb{U}} \mapsto \mathcal{H}$. The obfuscation is naturally bounded by the aforementioned types, thus:

$$\psi(g) = \psi(U^{\mathsf{pr}}) \in S(U^{\mathsf{pr}}) \tag{63}$$

The presence of a classifier changes the probability distribution of activity histories generated by malicious users. Concretely, the distribution of malicious activity histories is now governed by the distribution of obfuscated primary goals.

$$\dot{p}(h|M) = \sum_{U^{\mathsf{pr}}:\psi(U^{\mathsf{pr}})=h} p(U^{\mathsf{pr}}) \tag{64}$$

This illustrates how the general adversarial machine notions map to a problem instance. In the previous section we used h as a general variable that represents the ground discrete objects that are classified by the detector. However, as shown above, the inner structure of h can be ... blabla.

note that primary goal g comprises only partially h, thus there is room for obfuscation.

5.2 Detector

Reflect D(d, x) in the following definition

Mathematically, the task of the detector is to find a mapping $h \in \mathcal{H}$ which classifies a query history's feature vector $x \in \mathcal{X}$ to a class $\mathbb{C} = \{\mathsf{B}, \mathsf{M}\}$, ie. $h : \mathcal{X} \mapsto \mathbb{C}$. Note that B stands for benign users, while M denotes malicious users.

A feature vector representing a query history is given by a feature map $\Phi : h \mapsto \mathcal{X}$ which takes a query history h as an argument and maps it to a real vector $x \in \mathcal{X} \subseteq \mathbb{R}^d$. Naturally, Φ is surjective and is given a priori to task solving.

Following the ERM framework, the optimal classifier h^* is given by minimising its expected risk:

$$R_{-1}(h) = \mathop{\mathbb{E}}_{(h,c)\sim\dot{p}} [\, \ell_{-1}(h \circ \Phi(h),\, c) \,] \tag{65}$$

where $\dot{p}(h, c)$ represents the joint probability of a query history $h \in h$ and a class $c \in \mathbb{C}$, partly modified by the adversary as explained above. ℓ_{-1} stands for a classification loss function.

Generally, we prefer some classifier instances to others, therefore, we employ a regularisation term $\Omega_{-1}(h)$. In conclusion, the optimal classifier h^* is given by the following equation.

$$h^* = \min_{h\in\mathcal{H}} R_{-1}(h) + \Omega_{-1}(h) \tag{66}$$

Taking into account the Neyman-Pearson Task:

5.3 Attacker

Since the objectives of all malicious users are equivalent, ie. they aim to obfuscate their private set of primary queries Q^{pr}, the final query history of each of them is strictly a function of Q^{pr}. Due to the shared goal, we represent the malicious users as a single-body aggregate player, the adversary.

The adversary aims to identify an obfuscation function $g : h^{\mathsf{pr}} \mapsto h$. This is done for a fixed h and Φ by minimising the risk of the adversary $R_{+1}(g)$. However, following the threat model we restrict the adversary to produce only malicious samples (in contrast to the general form of the Stackelberg Prediction Game in [9]) and these are inherently given by the distribution of strictly primary query historys $p(h^{\mathsf{pr}})$. This simplifies the adversary's risk $R_{+1}(g)$ to a new form:

$$R_{+1}(g) = \mathop{\mathbb{E}}_{h^{\mathsf{pr}} \sim p_{h^{\mathsf{pr}}}} [\, \ell_{+1}(h \circ \Phi \circ g(h^{\mathsf{pr}}),\, \mathsf{M}) \,] \tag{67}$$

where $p_{h^{\mathsf{pr}}}$ gives the probability distribution from which h^{pr} are drawn. $[\, \ell_{+1} : \mathbb{C} \times \mathbb{C} \mapsto \mathbb{R}$ is the adversary's cost function.

Similarly to h-regularisation, we prefer some obfuscation functions to others which is articulated by a regulariser $\Omega_{+1}(g)$. In conclusion, the optimal obfuscation function g^* is given by the following equation:

$$g^* = \min_{g \in \mathcal{G}} R_{+1}(g) + \Omega_{+1}(g) \tag{68}$$

5.3.1 Attacker's optimisation problem

In practical terms, if the detector's decision is differentiable, then the attacker may use gradient descent to optimise its actions. And if the cost of the optimal action is larger then the cost of its detection, it chooses not to attack at all.

Three layers: X,H,G non differentiable transitions.

Link to Gradient Obfuscation.

Let $\Phi : \mathcal{H} \mapsto \mathcal{X}$ be a feature map that creates feature vectors $\Phi(h) \in \mathcal{X} \subset \mathbb{R}^N$. Assume Φ constructs features of h based on various numerical properties and attributes $p \in \mathbb{P}$.

Definition 5.1. *Let $V_\Phi = \{v_1, \ldots, v_K\}$ be a set of vectors that compose a basis of $\mathbb{P}$. We call V_Φ a basal mixtures of Φ. Let $\phi_V : \mathcal{G} \times \mathbb{R}^L \mapsto \mathcal{H}$ be a mapping $\phi_V(g, t)$ which is differentiable in t and composes activity histories h mixing the primary goal and vectors from V_Φ based on t.*

Note that vectors from V_Φ must extis. Sometimes full vdoes nt aexist.

IN this way we can bypass S(g) by simply chossing a proper t.

For example histogram.

Proposition 5.1.

6 Experiments

7 Conclusions

This work is a draft of a diploma thesis. It is an evaluated outcome of a semestral project that precedes the thesis. In the draft, a motivation to the problem and its definition is proposed. Concretely, we deal with a user behaviour classification problem that incorporates adversarial nature of some of the actors. The proposed threat model introduces a set of critical objects (here URLs) that a malicious user necessarily employs in communication with a service. This is a fundamental building block which imposes necessary modifications of the existing threat models met commonly in literature.

The draft explores related work and gives a formal definition of the problem, specifying a threat model that is inspired by the Stackelberg Prediction Game in [9]. However, some modifications to SPG are proposed. Last but not least, the game definition is augmented by each players' actions analysis, arriving at the conclusion the ERM framework is an excellent starting point for solving adversarial machine learning problems and gives, when combined with game theory, mathematical programs that are related to robust optimisation.

During thesis preparation, the draft will be enriched with the remaining parts of players' actions analysis. This will, *hopefully*, give a mathematical program for which an algorithm will be proposed. It goes without saying that the algorithm will then be compared to baseline approaches on real-world data and the experiment results will be evaluated. For now, it seems the HTTP data from the university's DNS logs will be used. Yet, this is a subject to change.

Finally, let us naïvely believe the proposed models and experimental findings will serve the common good, rather then the truly malicious actors.

References

[1] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. S. Bernstein, A. C. Berg, and F. Li, "Imagenet large scale visual recognition challenge," *CoRR*, vol. abs/1409.0575, 2014.

[2] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples. corr (2015)," 2015.

[3] A. M. Nguyen, J. Yosinski, and J. Clune, "Deep neural networks are easily fooled: High confidence predictions for unrecognizable images," *CoRR*, vol. abs/1412.1897, 2014.

[4] K. Grosse, N. Papernot, P. Manoharan, M. Backes, and P. McDaniel, "Adversarial examples for malware detection," in *European Symposium on Research in Computer Security*, pp. 62–79, Springer, 2017.

[5] H. S. Anderson, A. Kharkar, B. Filar, D. Evans, and P. Roth, "Learning to evade static PE machine learning malware models via reinforcement learning," *CoRR*, vol. abs/1801.08917, 2018.

[6] D. Lowd and C. Meek, "Good word attacks on statistical spam filters," in *CEAS*, 2005.

[7] V. N. Vapnik, *Statistical Learning Theory*. Wiley-Interscience, 1998.

[8] J. Z. Kolter and E. Wong, "Provable defenses against adversarial examples via the convex outer adversarial polytope," *arXiv preprint arXiv:1711.00851*, vol. 1, no. 2, p. 3, 2017.

[9] M. Brückner and T. Scheffer, "Stackelberg games for adversarial prediction problems," in *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 547–555, ACM, 2011.

[10] B. I. Rubinstein, B. Nelson, L. Huang, A. D. Joseph, S.-h. Lau, S. Rao, N. Taft, and J. D. Tygar, "Antidote: understanding and defending against poisoning of anomaly detectors," in *Proceedings of the 9th ACM SIGCOMM conference on Internet measurement*, pp. 1–14, ACM, 2009.

[11] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *CoRR*, vol. abs/1706.03762, 2017.

[12] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," *arXiv preprint arXiv:1706.06083*, 2017.

[13] I. Evtimov, K. Eykholt, E. Fernandes, T. Kohno, B. Li, A. Prakash, A. Rahmati, and D. Song, "Robust physical-world attacks on machine learning models," *CoRR*, vol. abs/1707.08945, 2017.

[14] A. Kurakin, I. J. Goodfellow, and S. Bengio, "Adversarial machine learning at scale," *CoRR*, vol. abs/1611.01236, 2016.

[15] A. Athalye, N. Carlini, and D. Wagner, "Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples," *arXiv preprint arXiv:1802.00420*, 2018.

[16] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics* (G. Gordon, D. Dunson, and M. Dudík, eds.), vol. 15 of *Proceedings of Machine Learning Research*, (Fort Lauderdale, FL, USA), pp. 315–323, PMLR, 11–13 Apr 2011.

[17] A. Huang, A. Al-Dujaili, E. Hemberg, and U. O'Reilly, "Adversarial deep learning for robust detection of binary encoded malware," *CoRR*, vol. abs/1801.02950, 2018.

List of Figures

List of Tables