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 February, 2019

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 (Russakovsky et al. 2014) are prone to so called adversarial images (I. J. Goodfellow, Shlens, and Szegedy 2015). 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 (Nguyen, Yosinski, and Clune 2014).

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 (Grosse et al. 2017); malware is developed by optimising an executable binary (Anderson et al. 2018), and spams are improved statistically to avoid detection (Lowd and Meek 2005).

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 $\ell : \mathbb{C} \times \mathbb{C} \mapsto \mathbb{R}$.

$$R(h) = \mathbb{E}_{(x,c) \sim p}\, \ell(h(x), c)$$

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

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

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:

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

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.

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

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:

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

Vapnik (1998) 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.

3 Related Work

Examining adversarial aspects of various machine learning problems has currently been a popular topic. Mainly, this was triggered by I. J. Goodfellow, Shlens, and Szegedy (2015) 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 (Kolter and Wong 2017), while others reframed the classification problem to incorporate aspects of game theory (Brückner and Scheffer 2011). Despite most of the work deals with image classification, efforts to utilise the same notions in computer security have been seen too (Anderson et al. 2018). Susceptibility to adversarial examples is, however, not the only weakness adversaries exploit, they also are able to modify future training datasets in their favour (Rubinstein et al. 2009).

3.1 Adversarial setting

Lowd and Meek (2005) explore attack 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 attack, random words from a dictionary are drawn; the second attack utilises common legitimate e-mail words; and in the third attack, 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 attack 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 attack 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; Russakovsky et al. 2014), natural language processing (Vaswani et al. 2017) and in many others fields. However, it was found that neural networks are susceptible to artificially crafted images. In particular, I. J. Goodfellow, Shlens, and Szegedy (2015) 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 (Nguyen, Yosinski, and Clune 2014). 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 Madry et al. (2017) 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:

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

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 (Evtimov et al. 2017).

To fully relate to an adversarial setting, Madry et al. (2017) 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:

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

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. (Madry et al. 2017) 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) (Kurakin, Goodfellow, and Bengio 2016). 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 (Athalye, Carlini, and Wagner 2018).

In their paper, Athalye, Carlini, and Wagner (2018) 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 (Madry et al. 2017). A method that aims to give provable resistance to adversarial samples was proposed by Kolter and Wong (2017) who examine a novel network architecture that provably classifies all objects in a convex neighbourhood of a given image correctly. To achieve that, Kolter and Wong (2017) redefine a ReLU (Glorot, Bordes, and Bengio 2011) 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:

$$y \geq x,$$

$$y \geq 0,$$

$$y(l - u) \leq -ux + ul$$

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 (Kolter and Wong 2017).

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. Grosse et al. (2017) propose an attack that optimises a malicious source code by applying some of the predefined modifications. The attack 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%. A. Huang et al. (2018) 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. Anderson et al. (2018) 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 it's 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, Brückner and Scheffer (2011) 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 attack strategy. Concretely, Rubinstein et al. (2009) elaborate on poisoning anomaly detectors.

The poisoning attack 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 attack near to the poisoned areas of the input space.

The authors present that such an attack 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

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 behaviour 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 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.

4.1 Motivation

In this work, we consider a computer security company that runs an API service which returns rating of a queried URL. This type of service is usually deployed by such companies to provide their security software with access to most up-to-date database of URL ratings.

The typicall 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 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. In this work, we focus on malware producers that asses a set of URLs that serve as a 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 needs to regularly check whether any of its URLs has been exposed – by querying the publicly available URL rating system.

The task now is as follows: the computer security company desires to distinguish malicious users of the URL rating service from benign ones.

In principle, the task is an instance a broader family of problems. Agents query a service in order to discern information about a set of critical objects and we are given a specificity function which evaluates those objects. In other words, the specificity function f heuristically assigns a high value number to salient objects that we are interested in and a near-to-zero number to unrelated objects. Consequently, we use the f-values to specify what queries (or the queried objects, respectively) are relevant in the given instance of the user activity classification problem.

Therefore, the communication between an agent and a service, consisting of the agent querying the service with the objects, can be seen as a sequence of f-values of those objects. Based on f, we design a classifier that sorts agents into two groups; one given by agents carrying out communication which is somewhat prone to high values of f, and the other grouping agents with low f-values communication.

This work's use-case takes the URL rating as the specificity function f, and each URL is considered an object which the service is queried with. Thus, we aim to classify the agents which tend to query high-danger URLs to one group and the other agents to a the other–benign–group.

4.2 Game Model

Formally, the service is queried with a query $q \in \mathbb{Q}$ where $\mathbb{Q}$ is the set of all queries. The service securely assigns each query to a user. Thus, we define a query profile $\pi \in \Pi$ built using the queries of the user, possibly supplied with additional query properties. For example, if a user sends a sequence of queries to the service supplemented with a timestamp, a source IP or possibly other information, this is recorded and integrally stored in a corresponding user's query profile π.

$$(q_1, t_1, \mathsf{IP}_1, \dots),\ (q_2, t_2, \mathsf{IP}_2, \dots),\ \dots,\ (q_k, t_k, \mathsf{IP}_k, \dots) \longrightarrow \pi$$

For practical reasons, we define a partially inverse mapping $\sigma(\pi) \subseteq \mathbb{Q}$ which denotes the queries comprising the profile π. Using the example above $\sigma(\pi) = \{q_1, q_2, \dots, q_k\}$.

A single malicious user posses a private set of critical queries $Q^{\mathsf{cr}} \subset \mathbb{Q}$ that contains queries which the malicious user will necessarily employ to achieve its goal. In other words, there is a piece of information which is sought by the malicious user and which is obtainable only by querying the service with critical queries Q^{cr}. Given its critical queries, a malicious user sends requests to the service with queries $Q \subseteq \mathbb{Q}$ that may next to its critical queries also contain legitimate queries which it uses to cover-up its activity.

$$Q^{\mathsf{cr}} \subseteq Q$$

If there was no classifier and malicious users were not motivated to adjust their behaviour, they would simply query the service with Q resembling critical queries, presumably containing just a little overhead, ie. $Q \cong Q^{\mathsf{cr}}$. A user query activity comprising only queries in Q^{cr} creates a strictly critical query profile π^{cr} such that $\sigma(\pi^{\mathsf{cr}}) = Q^{\mathsf{cr}}$. However, taking into account the overhead queries in Q, the probability of a query profile of a malicious user $p(\pi|M)$ is given by the underlying distribution of critical queries Q^{cr} and overhead queries. Assuming there is only little overhead, $p(\pi|M)$ becomes the probability of strictly critical query profiles $p(\pi^{\mathsf{cr}})$.

$$p(\pi|M) \cong p(\pi^{\mathsf{cr}})$$

Nonetheless, once there actually is a classifier deployed, implying a cost for disclosure of a malicious user, the malicious users rationally query the service with additional legitimate queries to obfuscate the classifier. Since we allow only *adding* (legitimate) queries to a query profile, each strictly critical query profile induces a bounded set of profiles $S(\pi^{\mathsf{cr}})$ that contains profiles derivable from π^{cr} by adding queries.

$$S(\pi^{\mathsf{cr}}) = \{\pi \in \Pi \mid \sigma(\pi^{\mathsf{cr}}) \subseteq \sigma(\pi)\}$$

This is thoroughly captured by an obfuscation function $g : \Pi^{\mathsf{cr}} \mapsto \Pi$ which a malicious user employs to transform its original strictly critical query profile π^{cr} to an obfuscated query profile $g(\pi^{\mathsf{cr}}) \in S(\pi^{\mathsf{cr}})$. In this case, the presence of the classifier changes the probability distributions. Concretely, the distribution of malicious query profiles is now governed by the distribution of obfuscated strictly critical query profiles.

$$\dot{p}(\pi|M) = p(g(\pi^{\mathsf{cr}}))$$

To conclude, 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 profiles of malicious users in such a way they get misclassified by the classifier. The detector is a -1 player and the adversary is a $+1$ player.

4.2.1 Detector

Mathematically, the task of the detector is to find a mapping $h \in \mathcal{H}$ which classifies a query profile'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 profile is given by a feature map $\Phi : \Pi \mapsto \mathcal{X}$ which takes a query profile π 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:

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

where $\dot{p}(\pi, c)$ represents the joint probability of a query profile $\pi \in \Pi$ 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.

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

4.2.2 Adversary

Since the objectives of all malicious users are equivalent, ie. they aim to obfuscate their private set of critical queries Q^{cr}, the final query profile of each of them is strictly a function of Q^{cr}. 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 $g : \Pi^{\mathsf{cr}} \mapsto \Pi$. 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 by Brückner and Scheffer (2011)) and these are inherently given by the distribution of strictly critical query profiles $p(\pi^{\mathsf{cr}})$. This simplifies the adversary's risk $R_{+1}(g)$ to a new form:

$$R_{+1}(g) = \mathop{\mathbb{E}}_{\pi^{\mathsf{cr}} \sim p_{\pi^{\mathsf{cr}}}} \ell_{+1}(h \circ \Phi \circ g(\pi^{\mathsf{cr}}), \mathsf{M})$$

where $p_{\pi^{\mathsf{cr}}}$ gives the probability distribution from which π^{cr} 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:

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

4.3 Zero-one Loss Assumption

Let us now assume that both ℓ_{-1} and ℓ_{+1} are *1-0 losses* that penalise misclassification, or detection, respectively. Then the adversary's risk simplifies to:

$$R_{+1}(g) = \sum_{\substack{\pi^{\mathsf{cr}} \in \Pi^{\mathsf{cr}} \\ h \circ \Phi \circ g(\pi^{\mathsf{cr}}) = \mathsf{M}}} p(\pi^{\mathsf{cr}})$$

Note that this means the optimal $g^*(\pi^{\mathsf{cr}})$ gives an obfuscated query profile $\pi \in S(\pi^{\mathsf{cr}})$ that is classified as B and minimises a regulariser $\Theta_{+1}(\pi, \pi^{\mathsf{cr}})$ which is induced by $\Omega_{+1}(g)$. The adversary's minimisation problem can be rewritten to:

$$g^*(\pi^{\mathsf{cr}}) \in \min_{\pi \in S(\pi^{\mathsf{cr}})} \Theta_{+1}(\pi, \pi^{\mathsf{cr}}) \qquad \text{s.t.} \quad h \circ \Phi(\pi) = \mathsf{B}$$

In cases in which there is no obfuscated profile that the classifier labels benign, $g^*(\pi^{\mathsf{cr}})$ simply returns π^{cr} or any element of $S(\pi^{\mathsf{cr}})$ depending on what type of Stackelberg equilibrium is played.

Additionally, if $\Theta_{+1}(\pi, \pi^{\mathsf{cr}})$ is convex in π, there is a single optimal value $g^*(\pi^{\mathsf{cr}})$. However, note that this for example does not hold true for $\Theta_{+1}(\pi, \pi^{\mathsf{cr}}) = |\sigma(\pi)|$ as there might exist a tuple π_1 and π_2, s.t. $|\sigma(\pi_1)| = |\sigma(\pi_2)|$ but $\sigma(\pi_1) \neq \sigma(\pi_2)$.

The *1-0 loss* also simplifies the detectors risk. It is convenient to decompose the risk into two components: the first stands for benign users expectation and the second for malicious users expectation.

$$R_{-1}(h) = \sum_{\substack{\pi \in \Pi \\ h \circ \Phi(\pi) = \mathsf{M}}} p(\pi, \mathsf{B}) + \sum_{\substack{\pi^{\mathsf{cr}} \in \Pi^{\mathsf{cr}} \\ h \circ \Phi \circ g^*(\pi^{\mathsf{cr}}) = \mathsf{B}}} p(g^*(\pi^{\mathsf{cr}}), \mathsf{M})$$

4.4 Additional remarks

What is to be done in the game specification:

- finalise the derivation of the optimal $h^*(x)$
- introduce a naive approach based on the assumption the URL ratings, f, are independent, ie. $p(\mathsf{B}|\pi) = \prod_{q \in \sigma(\pi)} f(q)$
- show how this translates to empirical risks and actual classifier optimisation criterion

The threat model is extendable by the following notions:

- a time window τ is used to collect query profiles; the detection repeats at the end of each collection window
- a malicious user buys a new license once it is detected and continues querying the service with the remaining queries of Q^{cr}
- a malicious user is allowed to create strictly benign query profiles in order to poison future retraining datasets (most likely, will not be eventually explored)

5 Datasets

The problem definition proposed in the previous section gives a mathematical program which returns a solution to the problem based on empirical data. The empirical data are needed because the theoretical fundaments of the problem definition concern terms that are principally unknown. In particular, the probabilities of query profiles generated by both benign and malicious users $p(\pi, c)$ and the specificity function $f : \mathbb{Q} \mapsto [0, 1]$ are unknown.

As shown, the game model defines a malicious user that posses a set of critical queries Q^{cr} which it inevitably uses in communication with a service. We assume if there is no classifier deployed on the service side, the malicious users query the service with Q, approximately consisting of queries in Q^{cr} only. If there is a classifier, a malicious user changes its behaviour to cover up its activity while still pursuing its goal. Therefore, to asses Q^{cr} instances, our assumption suggests data collected without any detector deployed may suitably represent the critical query sets.

In contrast, a benign user produces same activity regardless of any classification employed. Thus, any activity recordings that capture what queries are used in communication by benign, legitimate users are essentially a good source of data.

Finally, we propose to utilise values of a specificity function f in order to construct the feature map Φ. The specificity, given by f, estimates the level of benignity of a query. The values of this function can either be collected from a service that actually provides them, for example the aforementioned URL rating system, or, less suitably, manually crafted based on the information in each query. An example of a crafted specificity function may be any anomaly measure of a query. However, an authentic specificity function is preferred.

In conclusion, to plausibly estimate the unknown values, we require a data collection contains:

1. query profiles of benign users, to estimate $p(\pi|\mathsf{B})$;
2. different instances of Q^{cr}, to estimate $p(\pi^{\mathsf{cr}})$;
3. a specifity function value $f(q)$ for each query in the dataset.

5.1 Publicly Availble Data Sources

The motivation to the problem concerns a URL rating service that is exploited by attackers trying to find rating of a set of malicious URLs. Therefore, ideally, we aim to work with a dataset recording user communication with a service that explicitly contains URLs and other, potentially useful, attributes of HTTP requests. Also, we need plausible ratings of the URLs captured in the data set.

To best of our knowledge, unfortunately, any of publicly available data sources does not fully provide all types of data aspects needed. Especially, we lack data allowing to assign or record values of f. The main reason is, understandably, the level of data anonymisation. Usually, recorded activity is anonymised in terms of both the actor and the query for privacy concerns. From the many data sets publicly available we present two that, despite being mildly inappropriate, at least partially fit our requirements.

The data collected in Lawrence Berkeley Laboratory (Paxson 1994)[1] consists of a tuple: a user id and a target IP address. Other TCP connection information is hidden making the data set inappropriate for the lack of specificity function and critical queries availability. The second example is the dataset collected at Boston University (Cunha, Bestavros, and Crovella 1995)[2] in which HTTP connections are recorded. However, the dataset contains mostly records with repeated access to university department sites. Thus, we question the actual user activity distribution essential to our problem is recorded in this dataset.

5.2 FEE-CTU Dataset

In terms of sources that are not publicly available, we have access to HTTP connections recorded within the network of The Faculty of Electrical Engineering at Czech Technical University (FEE-CTU). This dataset comprises a public IP of a request sender, a target URL and other request attributes useful in features construction. Also, we are given access to URL ratings recorded by the company Trend Micro[3] which, in fact, runs a URL rating service that is attacked in a way proposed in the game model.

The HTTP data from CTU contain following attributes: an IP address of the sender; a full-length URL; a referrer; browser specification; a timestamp; and other partially unrelated attributes to our problem. The data were recorded on June 1, 2016 for the entire day. To relate to our problem, we assume a user is identified by the IP address of the sender and the queried object is the accessed URL.

Removing users exhibiting less than 10 queries, we arrive at a dataset of 2300 users who sent nearly $800,000$ queries. The total number of unique URLs is nearly $10,000$. However, this only represents benign users activity. Therefore, we randomly assign instances of critical query sets based on both the URL rating and what users accessed the URL. However, an exact form of the critical query sets assessment is a subject to change and will be determined once URL ratings are obtained.

[1] Available at: ita.ee.lbl.gov/html/contrib/LBL-CONN-7.html

[2] Available at: ita.ee.lbl.gov/html/contrib/BU-Web-Client.html

[3] See link for more information: www.trendmicro.com

6 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 by Brückner and Scheffer (2011). 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

Anderson, Hyrum S., Anant Kharkar, Bobby Filar, David Evans, and Phil Roth. 2018. "Learning to Evade Static PE Machine Learning Malware Models via Reinforcement Learning." *CoRR* abs/1801.08917. `http://arxiv.org/abs/1801.08917`.

Athalye, Anish, Nicholas Carlini, and David Wagner. 2018. "Obfuscated Gradients Give a False Sense of Security: Circumventing Defenses to Adversarial Examples." *arXiv Preprint arXiv:1802.00420*.

Brückner, Michael, and Tobias Scheffer. 2011. "Stackelberg Games for Adversarial Prediction Problems." In *Proceedings of the 17th Acm Sigkdd International Conference on Knowledge Discovery and Data Mining*, 547–55. ACM.

Cunha, Carlos, Azer Bestavros, and Mark Crovella. 1995. "Characteristics of Www Client-Based Traces." Boston, MA, USA: Boston University.

Evtimov, Ivan, Kevin Eykholt, Earlence Fernandes, Tadayoshi Kohno, Bo Li, Atul Prakash, Amir Rahmati, and Dawn Song. 2017. "Robust Physical-World Attacks on Machine Learning Models." *CoRR* abs/1707.08945. `http://arxiv.org/abs/1707.08945`.

Glorot, Xavier, Antoine Bordes, and Yoshua Bengio. 2011. "Deep Sparse Rectifier Neural Networks." In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, edited by Geoffrey Gordon, David Dunson, and Miroslav Dudík, 15:315–23. Proceedings of Machine Learning Research. Fort Lauderdale, FL, USA: PMLR. `http://proceedings.mlr.press/v15/glorot11a.html`.

Goodfellow, Ian J, Jonathon Shlens, and Christian Szegedy. 2015. "Explaining and Harnessing Adversarial Examples. CoRR (2015)."

Grosse, Kathrin, Nicolas Papernot, Praveen Manoharan, Michael Backes, and Patrick McDaniel. 2017. "Adversarial Examples for Malware Detection." In *European Symposium on Research in Computer Security*, 62–79. Springer.

Huang, Alex, Abdullah Al-Dujaili, Erik Hemberg, and Una-May O'Reilly. 2018. "Adversarial Deep Learning for Robust Detection of Binary Encoded Malware." *CoRR* abs/1801.02950. `http://arxiv.org/abs/1801.02950`.

Kolter, J Zico, and Eric Wong. 2017. "Provable Defenses Against Adversarial Examples via the Convex Outer Adversarial Polytope." *arXiv Preprint arXiv:1711.00851* 1 (2):3.

Kurakin, Alexey, Ian J. Goodfellow, and Samy Bengio. 2016. "Adversarial Machine Learning at Scale." *CoRR* abs/1611.01236. `http://arxiv.org/abs/1611.01236`.

Lowd, Daniel, and Christopher Meek. 2005. "Good Word Attacks on Statistical Spam Filters." In *CEAS*.

Madry, Aleksander, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2017. "Towards Deep Learning Models Resistant to Adversarial Attacks." *arXiv Preprint arXiv:1706.06083*.

Nguyen, Anh Mai, Jason Yosinski, and Jeff Clune. 2014. "Deep Neural Networks Are Easily Fooled: High Confidence Predictions for Unrecognizable Images." *CoRR* abs/1412.1897. `http://arxiv.org/abs/1412.1897`.

Paxson, Vern. 1994. "Empirically Derived Analytic Models of Wide-Area Tcp Connections." *IEEE/ACM Trans. Netw.* 2 (4). Piscataway, NJ, USA: IEEE Press:316–

36. https://doi.org/10.1109/90.330413.

Rubinstein, Benjamin IP, Blaine Nelson, Ling Huang, Anthony D Joseph, Shing-hon Lau, Satish Rao, Nina Taft, and J Doug Tygar. 2009. "Antidote: Understanding and Defending Against Poisoning of Anomaly Detectors." In *Proceedings of the 9th Acm Sigcomm Conference on Internet Measurement*, 1–14. ACM.

Russakovsky, Olga, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, et al. 2014. "ImageNet Large Scale Visual Recognition Challenge." *CoRR* abs/1409.0575. http://arxiv.org/abs/1409.0575.

Vapnik, Vladimir N. 1998. *Statistical Learning Theory.* Wiley-Interscience.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. "Attention Is All You Need." *CoRR* abs/1706.03762. http://arxiv.org/abs/1706.03762.