Statistically Robust Neural Network Classification

Benjie Wang[1] **Stefan Webb**[2] **Tom Rainforth**[3]

[1]Department of Computer Science, University of Oxford
[2]Twitter Cortex, San Francisco
[3]Department of Statistics, University of Oxford

Abstract

Despite their numerous successes, there are many scenarios where adversarial risk metrics do not provide an appropriate measure of robustness. For example, test-time perturbations may occur in a probabilistic manner rather than being generated by an explicit adversary, while the poor train–test generalization of adversarial metrics can limit their usage to simple problems. Motivated by this, we develop a probabilistic robust risk framework, the ***statistically robust risk*** (SRR), which considers pointwise corruption distributions, as opposed to worst-case adversaries. The SRR provides a distinct and complementary measure of robust performance, compared to natural and adversarial risk. We show that the SRR admits estimation and training schemes which are as simple and efficient as for the natural risk: these simply require noising the inputs, but with a principled derivation for exactly how and why this should be done. Furthermore, we demonstrate both theoretically and experimentally that it can provide superior generalization performance compared with adversarial risks, enabling application to high-dimensional datasets.

1 INTRODUCTION

Since the discovery of the phenomenon of adversarial examples for neural networks [Szegedy et al., 2014, Goodfellow et al., 2015, Papernot et al., 2016], a variety of approaches for assessing and mitigating their impact on decision-making systems have been proposed [Gu and Rigazio, 2015, Moosavi-Dezfooli et al., 2016, Madry et al., 2018]. Much of this work has focused on the formal verification of neural network classifiers, such as the robustness of predictions under a L_p-norm perturbation set [Gehr et al., 2018, Wang et al., 2018], typically doing this in an input specific manner. Motivated by explicit adversarial attacks, these approaches are focused on worst-case robustness: they are based on the largest loss within the perturbed region.

Though highly appropriate in a variety of cases, this general approach is not universally applicable. Firstly, one is often concerned about robustness to naturally occurring, or ***random***, input perturbations, rather than an explicit adversary. For example, in self-driving cars we may not have access to the exact inputs due to sensor imperfections and wish to ensure our predictions are robust to such variations. Here our classifier must account for these variations, but some level of risk will usually be acceptable: it will typically be neither feasible nor necessary to guarantee there are ***no*** possible adversarial inputs, but we instead wish to ensure the ***probability*** of encountering such an input is sufficiently low.

Secondly, in practice, one is usually concerned with the ***overall*** robustness of the network, that is, its robustness across the range of possible inputs that it will see at test-time. This has motivated network-wide worst-case robustness definitions, such as average minimal adversarial distance [Fawzi et al., 2018] and adversarial risk [Madry et al., 2018], along with associated training schemes [Wong and Kolter, 2018, Madry et al., 2018]. However, whereas the motivation for requiring worst-case robustness for individual inputs is often clear, it is more difficult to motivate using worst-case robustness for the classifier as a whole; a classifier can only be perfectly worst-case robust if it is robust to all possible perturbations of all inputs, something which will very rarely be achievable in practice. Moreover, previous work has shown that worst-case robustness metrics can have very poor generalization from train to test time, both theoretically and in practice for real networks, substantially reducing their applicability [Schmidt et al., 2018, Yin et al., 2019].

To address these limitations, we suggest a class of alternative robust risk metrics, which we term ***statistically robust risks*** (SRRs), that naturally arise when relaxing worst-case adversaries to pointwise perturbation distributions. SRRs can be used to assess the overall probabilistic robustness

Proceedings of the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence (UAI 2021), PMLR 161:1735–1745.

of a classifier by averaging a loss function over both possible inputs and an input perturbation distribution. Unlike adversarial risks, our SRR framework naturally applies at a network-wide level due to the law of total expectation. We emphasize that this framework is not a replacement for adversarial risk, or a means to learn adversarially robust networks, but a distinct and complementary measure of robustness that will be more appropriate in some scenarios. Our work can be viewed as an extension and generalization of the pointwise statistical robustness work of Webb et al. [2019], which quantifies the expected robustness of an ***individual*** datapoint under a perturbation distribution.

Contributions Our contributions are as follows. Firstly, we provide theoretical and empirical results showing that SRRs have superior generalization performance to their corresponding adversarial risks, particularly in high-dimensions, with bounds on the generalization error respectively scaling as $O(\log(d))$ and $O(\sqrt{d})$ in the size of the network. This suggests that it may be possible to obtain statistically robust networks in a wide range of applications where adversarial robustness is still elusive or inappropriate. Further, we show that the SRR admits efficient estimation and training schemes which incur no extra computational cost over standard training. Indeed, training to a SRR requires only a noising of the inputs passed to the network, such that it encompasses, motivates, and formalizes many commonly-used heuristics.

We justify the practical utility of our statistical robustness metric with a number of novel insights. Firstly, we demonstrate that SRRs can differ significantly from both their corresponding natural (i.e., non-robust) and adversarial risk, and as such provide a unique metric for both training and testing networks that helps ensure robustness to probabilistic input perturbations. Secondly, we find that SRRs generalize well *across different perturbation distributions*, meaning that it is not necessary to have knowledge of the precise test-time perturbation distribution. Finally, we show that practical safety properties encoded through bespoke loss functions can be tackled through SRRs, while standard training suffers from overfitting and instability.

2 BACKGROUND

2.1 ADVERSARIAL EXAMPLES

Although the general concept of adversarial examples (a perturbed input data point that is classified poorly) is well understood, the precise definition is often left implicit in the literature, despite many versions being present [Diochnos et al., 2018]. To formalize this, let f_θ represent the classifier (with parameters θ) and c label the true class. Let x be the original input point and x' the perturbed input point. Three different definitions of an adversarial example are now commonly used:

- **Prediction change (PC)** $f_\theta(x') \neq f_\theta(x)$;
- **Corrupted instance (CI)** $f_\theta(x') \neq c(x)$;
- **Error region (ER)** $f_\theta(x') \neq c(x')$.

The distinction between PC and CI is that the former is concerned with whether a perturbation changes the classification (regardless of whether it is correct), while the latter concerns whether the perturbed point is classified correctly. The ER definition is typically not measurable, since we usually will not have labels for perturbed points x'. Since we are interested in risk metrics, we take the CI definition.

2.2 NATURAL AND ADVERSARIAL RISKS

The ***risk*** of a classifier f_θ is a measure of its average performance with respect to the data distribution:

$$r_{\mathcal{D}}(f_\theta) \triangleq \mathbb{E}_{(X,Y)\sim p_{\mathcal{D}}}\left[L(X, Y, f_\theta)\right], \tag{1}$$

where (X, Y) is an input/target pair, $p_{\mathcal{D}}$ is the true data generating distribution, and L is some loss function. For non-robust risks, $L(X, Y, f_\theta)$ can usually be written in the form $\phi(f_\theta(X), Y)$, which we term the ***natural risk***. An empirical version of this, $R_N(f_\theta)$, can be obtained by replacing the expectation with a Monte Carlo average over a training dataset $\{(x_1, y_1), ..., (x_N, y_N)\}$. Training a classifier then corresponds to solving the optimization problem $\min_\theta R_N(f_\theta)$.

To model the effect of an adversary limited to additively perturbing inputs by a vector δ within a limited set Δ (e.g., an L_∞-ball), ***adversarial risk*** is defined as

$$r^{\text{adv}}_{\mathcal{D}}(f_\theta) \triangleq \mathbb{E}_{(X,Y)\sim p_{\mathcal{D}}}\left[\max_{\delta\in\Delta} \phi(f_\theta(X+\delta), Y)\right], \tag{2}$$

which is in fact a form of risk with loss function $L^{\text{adv}} \triangleq \max_{\delta\in\Delta} \phi(f_\theta(X+\delta), Y)$. When the 0-1 loss function $\phi(f_\theta(X), Y) = \mathbb{1}_{\arg\max_{i=1,\cdots,M} f_\theta(X)_i \neq Y}$ is used, this is known as ***adversarial accuracy***. Optimizing the empirical adversarial risk, $R^{\text{adv}}_N(f_\theta)$, corresponds to a robust optimization problem [Ben-Tal et al., 2009]. ***Adversarial training*** [Goodfellow et al., 2015, Kurakin et al., 2018, Madry et al., 2018] solves the problem by lower bounding the inner maximization using gradient-based methods to generate maximally adversarial examples, and training on this approximate loss.

2.3 STATISTICAL ROBUSTNESS

Webb et al. [2019] recently introduced a *statistical robustness* metric that provides a probabilistic alternative to formal verification of ***pointwise*** robustness. Standard verification

schemes target the binary 0-1 metric on whether an adversarial example exists in a perturbation region Δ around a point. Their statistical robustness metric instead corresponds to the ***probability*** of drawing an adversarial example from some ***perturbation distribution***. Concretely, for a perturbation distribution $p(\cdot|x)$ centred around x, they define their statistical robustness metric as

$$\mathcal{I}[p] \triangleq \mathbb{E}_{X' \sim p(\cdot|x)} \left[\mathbb{1}_{f_\theta(X') \neq f_\theta(x)} \right]. \tag{3}$$

This generalizes, and provides more information than, verification about the network's robustness around x: if there is no adversarial example in the support of $p(\cdot|x)$, then the probability is 0, whereas if there is an adversarial example, the metric indicates how likely we are to encounter one.

2.4 GENERALIZATION

The problem of generalization is fundamental in machine learning: we want classifiers to perform well on not just training points but unseen test points. It is well known in statistical learning theory that we can probabilistically upper bound the generalization error $r_{\mathcal{D}}(f) - R_N(f)$ of a learning algorithm using notions of complexity on the admissible set of classifiers (e.g. all parameterizations of a neural network) and loss function [Shalev-Shwartz and Ben-David, 2014]. Intuitively, if the admissible set of functions is less complex, then there is less capacity to overfit to the training data.

To be more precise, we define the ***empirical Rademacher complexity*** (ERC) for function class $\mathcal{F} : \mathbb{R}^d \to \mathbb{R}$ and sample set $\mathcal{S} = \{x_1, ..., x_N\}$, $x_i \in \mathbb{R}^d$ to be [Shalev-Shwartz and Ben-David, 2014]:

$$\text{Rad}_{\mathcal{S}}(\mathcal{F}) := \frac{1}{N} \mathbb{E}_\sigma \left[\sup_{f \in \mathcal{F}} \sum_{n=1}^{N} \sigma_n f(x_n) \right], \tag{4}$$

where $\sigma_1, ..., \sigma_N$ are independent Rademacher random variables, which take either the value -1 or $+1$, each with probability $1/2$. Intuitively, this measures the complexity of the class by determining how many different ways functions $f \in \mathcal{F}$ can classify the sample $\mathcal{S}$.

3 FROM ADVERSARIAL TO STATISTICAL RISKS

Adversarial examples originally captured the attention of the machine learning community by demonstrating a discrepancy between the behaviour of NNs and human reasoning. Given that, in domains such as computer vision and natural language processing, the long-term goal is to attain models which can reason as humans do, it is natural to define robustness in terms of all semantically meaningful perturbations. However, given the rapid adoption of machine learning systems in applications such as autonomous vehicles and medical diagnosis, robustness is now also a vital practical requirement: brittleness to input perturbations can have severe consequences. These goals, while seemingly aligned, can in fact sometimes be conflicting. We argue that the latter agenda requires its own treatment and develop an associated robustness framework that arises naturally when considering how ideas from adversarial robustness can be transferred to probabilistic settings.

3.1 THE SHORTFALLS OF ADVERSARIAL APPROACHES

Obtaining ***fully*** adversarially robust networks (in the sense of being robust to all meaningful perturbations to all possible points) is a typically infeasible task, even for simple datasets such as MNIST and when the set of semantically consistent perturbations is known [Schott et al., 2019, Schmidt et al., 2018, Yin et al., 2019]. As such, one must rely on robustness metrics, such as adversarial risk/accuracy.

However, this can have significant issues. First, adversarial risk loses information about ***how*** robust a point is. By taking the worst point within a perturbation set, adversarial risk is by definition independent of performance on the vast majority of the perturbed input space (so long as it is better than the worst point). Adversarial risk thus places stringent $0/1$ requirements on each point from the data distribution, such that it favors a greater *number* of adversarially (completely) robust points, without guarantees of any degree of robustness on other points. When considering applications where perturbations are randomly generated, this can be very misleading, or even dangerous.

Consider, for instance, a network trained to classify disease based on medical imaging. Due to imperfections in the imaging equipment, as well as variation in equipment across different hospitals, random noise may be introduced to the test dataset. Optimising for adversarial risk might make the network adversarially robust on 80% of test points; however, if it is often fooled by random noise on the remaining 20%, this could result in misdiagnoses for many patients. In contrast, a network is robust against 95% of random perturbations overall would be preferable, even if it is adversarially robust on many fewer points. In these cases, somewhat counterintuitively, adversarial risk does not correspond well with the required notion of safety or robustness.

Second, learning with adversarial risk has proven to be very difficult from a statistical learning perspective due to poor generalization. Though adversarial training is effective for reducing the adversarial risk of neural networks on small datasets like MNIST, success has been limited in scaling up to higher-dimensional datasets. Schmidt et al. [2018] show this is due to a generalization gap, whereby, for CIFAR-10, it is possible to achieve adversarial accuracy of 97% on the training set, yet just 47% on the test set. This overfitting is in contrast to the natural case, where well-tuned networks

rarely overfit on CIFAR-10 [Yin et al., 2019].

In view of this, in the practical robustness agenda there is thus a need for robustness metrics which are not as conservative as adversarial risk, taking into account performance on a larger subset of input space, whilst also being amenable to training and tractable in the sense of generalization.

3.2 TOTAL STATISTICAL ROBUSTNESS METRIC

The pointwise statistical robustness framework introduced in Section 2.3 provides an indication of how we might construct some form of robust statistical risk. However, it is not directly applicable as a) we require a mechanism for assessing the ***overall*** robustness of a network; b) it only examines ***changes*** in predictions (PC), such that it can be satisfied by trivial networks which always make the same incorrect prediction; and c) it does not provide any practical mechanism for ***training*** networks.

To address the first two issues, we now introduce the ***total statistical robustness metric*** (TSRM) as follows:

$$\mathcal{I}_{\text{total}}[p] = \mathbb{E}_{(X,Y)\sim p_{\mathcal{D}}}\left[\mathbb{E}_{X'\sim p(\cdot|X)}\left[\mathbb{1}_{f_\theta(X')\neq Y}\right]\right], \quad (5)$$

where $p_{\mathcal{D}}$ is the true data generating distribution and $Y = c(X)$ is the true label of X. As it includes an expectation over the data, the TSRM is a measure for the overall robustness of the network. Intuitively, it can be thought of as the classification error under test-time input corruptions $p(\cdot|X)$.

Notice that the TSRM also varies from (3) in that it compares $f_\theta(X')$ to Y instead of $f_\theta(X)$. That is, we now consider CI examples. This is because we want to assess how the network performs over distribution $\mathcal{D}$. Analogously, while pointwise adversarial metrics are usually defined in terms of prediction change (PC), adversarial accuracy is defined in terms of misclassification (CI).

3.3 STATISTICALLY ROBUST RISK

The TSRM forms a useful metric for pre-trained networks, but it is not suitable as a training objective due to the difficulty of taking gradients through the identity function. Further, it explicitly assumes we are interested in probability of failure, rather than a more general loss. To address this, we note that it can be thought of as a specific risk and thus generalized to

$$r^{\text{stat}}_{\mathcal{D}}(f_\theta) \triangleq \mathbb{E}_{(X,Y)\sim p_{\mathcal{D}}}\left[\mathbb{E}_{X'\sim p(\cdot|X)}\left[\phi(f_\theta(X'), Y)\right]\right] \quad (6)$$

where ϕ represents a natural, pointwise, loss function as per Section 2.2.[1] We refer to $r^{\text{stat}}_{\mathcal{D}}(f_\theta)$ as a ***statistically robust risk*** (SRR). The TSRM now constitutes the special case of $\phi(f_\theta(X'), Y) = \mathbb{1}_{f_\theta(X')\neq Y}$. Note that the SRR corresponds to using the loss function

$$L^{\text{stat}}(X, Y, f_\theta) \triangleq \mathbb{E}_{p(X'|X)}\left[\phi(f_\theta(X'), Y)\right]. \quad (7)$$

Given a dataset $\{(x_1, y_1), (x_2, y_2), ..., (x_N, y_N)\}$, we can also define the ***empirical SRR***:

$$R^{\text{stat}}_N(f_\theta) \triangleq \frac{1}{N}\sum_{n=1}^{N}\mathbb{E}_{p(X'|x_n)}\left[\phi(f_\theta(X'), y_n)\right]. \quad (8)$$

The SRR framework provides a mechanism for linking statistical robustness back to the conventional notions of natural and adversarial risk, as well as a basis for theoretical analysis (see Section 4). The natural risk can be viewed as a special case of a SRR for which $p(X'|X)$ collapses to a Dirac delta measure about X, such that it does not take into account robustness to perturbations. On the other hand, by using the expected loss over $p(\cdot|X)$, instead of just the single worst-classified perturbation, a SRR contains important information that is not captured by an adversarial risk.

At a high-level, training using a SRR has the effect of "smoothing" the decision boundaries relative to using the corresponding natural risk. This can be useful when we want to be sensitive to certain classes or events, as it allows us to train our classifier to take conservative actions when the input is close to potentially problematic regions. For example, a self-driving car needs to ensure it avoids false negatives when predicting the presence of a pedestrian.

3.4 ESTIMATION AND TRAINING

The empirical SRR cannot be evaluated exactly as it contains an expectation over a perturbation distribution. A simple approximation approach is to use Monte Carlo estimation for each inner expectation, that is:

$$R^{\text{statMC}}_{N,C}(f_\theta) = \frac{1}{N}\sum_{n=1}^{N}\frac{1}{C}\sum_{m=1}^{C}\phi(f_\theta(x'_{n,m}), y_n) \quad (9)$$

where $\{x'_{n,m}\}$ is a sample from the perturbation distribution $p(\cdot|x_n)$ for $m \in \{1, \ldots, C\}$. Estimating $\mathbb{E}_{p(X'|x_n)}\left[\phi(f_\theta(X'), y_n)\right]$ in this way can be a challenging task when this expectation is very small (i.e. when the point is very robust), typically requiring specialist rare-event estimation methodology to avoid large ***relative errors*** [Webb et al., 2019].

Perhaps surprisingly though, this difficulty actually resolves itself when considering the empirical SRR as an overall estimation problem. This is firstly because, for practical networks and tasks, the empirical SRR is typically dominated by a small subset of the inputs x_n for which the pointwise statistical robustness loss is large ("non-robust" points), as opposed to the majority of inputs where the pointwise statistical robustness loss is very small ("robust" points). Consequently, large relative errors for these robust points do

[1] In certain cases, we may further require ϕ to also take X directly as an input. This potential dependency is not problematic and omitted simply for notational clarity.

not significantly impact on the overall error. Secondly, we usually have relatively large datasets ($N > 10^5$) for tasks involving neural networks, meaning that we do not necessarily need accurate estimates of pointwise robustness around each individual datapoint to obtain an accurate average: by the law of large numbers, the errors in our (unbiased) estimates will cancel each other when averaged.

Because of these effects, we found that Monte Carlo estimation with $C = 1$ (sampling a single point x'_n from $p(\cdot|x_n)$, then averaging the loss over these points) is often sufficiently accurate in practice for training and evaluation.

To use the SRR as a ***training objective***, we need to use a differentiable loss ϕ, such as the cross-entropy, so that we can perform stochastic gradient descent. We can then iterate through the training data by taking mini-batches $B \subset \{1, \dots, N\}$, drawing corresponding sample perturbations $x'_{n,m} \sim p(\cdot|X = x_n)$, and updating the network using the unbiased gradient updates

$$\nabla_\theta r^{\text{stat}}(p, f_\theta) \approx \frac{1}{\|B\|} \nabla_\theta \sum_{n \in B} \frac{1}{C} \sum_{m=1}^{C} \phi(f_\theta(x'_{n,m}), y_n), \tag{10}$$

noting that this is equivalent to conventional training but with the inputs randomly perturbed.

These insights mean that we can estimate and train on SRRs accurately with no additional cost to standard neural network training. This also provides justification for data augmentation schemes which sample a single perturbation for each datapoint, as accurately minimizing a statistically robust risk (without needing to sample many different perturbations, or use the original datapoint). Though this scheme is simple and efficient, our framework is itself agnostic to how we estimate/train (see Appendix C for discussion).

3.5 CHOICE OF PERTURBATION DISTRIBUTION

The aim of the perturbation distribution used in a SRR is not to be a completely accurate model of test-time perturbations; finding such a perturbation distribution is typically infeasible. Instead, it should be chosen to reflect what kind of perturbations we wish to be robust to. For example, if we are concerned about large-magnitude perturbations, we might use a heavy-tailed distribution such as a Cauchy. We may instead have known invariances in our inputs (e.g. rotations of an image) and construct our perturbation distribution to encapsulate these. Generally speaking, the most important consideration is to ensure that the perturbation distribution places mass over ***any*** test-time perturbations of concern; our results will later demonstrate that we generally achieve good test-time robustness when we train with higher–variance perturbations than those we consider at test–time.

4 THEORETICAL GENERALIZATION ANALYSIS

The poor generalization properties of adversarial risk in high-dimensions are a fundamental limitation on its applicability and utility: regardless of the tractability of the optimization procedure of training, we are left with no guarantees (or even an expectation) that our classifier will be robust at test-time. We will now show that SRRs do not suffer from the same limitation.

Given a neural network function class $\mathcal{F}$ and loss function class $L_\mathcal{F} \triangleq \{(X, Y) \to L(X, Y, f) : f \in \mathcal{F}\}$, we can bound the generalization error of a classifier using the following theorem [Mohri et al., 2012]:

Theorem 1. *Suppose $0 \leq L(X, Y, f) \leq c$ for all X, Y, f. Suppose further that the samples $S = \{(x_1, y_1), ..., (x_N, y_N)\}$ are i.i.d. from a distribution $p_\mathcal{D}$. Then for any $\delta \in (0, 1)$, with probability at least $1 - \delta$ the following holds for all $f \in \mathcal{F}$:*

$$\begin{aligned} &r_\mathcal{D}(f) - R_N(f) \\ &\quad \leq 2c \,\text{Rad}_S(L_\mathcal{F}) + 3c\sqrt{\log(2/\delta)/(2N)}. \end{aligned} \tag{11}$$

This bound is probabilistic, data-dependent and uniform over all $f \in \mathcal{F}$. This means it holds for all $f \in \mathcal{F}$, including those trained on the dataset S. Informally, this means that with high probability (in the formal sense) the empirical risk on the training dataset will be "close" to the true risk (i.e. difference bounded by the term on the RHS).

To take advantage of this bound, we need to be able to compute $\text{Rad}_S(L_\mathcal{F})$. The ERC (see Eq. 4) of the neural network function class $\text{Rad}_S(\mathcal{F})$ can be upper bounded [Bartlett et al., 2017, Yin et al., 2019] by an expression $O(\log(d_{\text{max}}))$ in dimension, where d_{max} is the maximal number of nodes in a single layer. Thus we simply need to relate $\text{Rad}_S(L_\mathcal{F})$ to $\text{Rad}_S(\mathcal{F})$.

Consider first the natural case, for which $L^{nat}(X, Y, f) \triangleq \phi(f(X), Y)$. If $\phi(\cdot, \cdot)$ is $\gamma-$Lipschitz in the first argument, we can use the Talagrand Contraction Lemma [Ledoux and Talagrand, 2013], which gives that $\text{Rad}_S(L_\mathcal{F}) \leq \gamma \,\text{Rad}_S(\mathcal{F})$. Thus, substituting this inequality into (11), we have

$$\begin{aligned} &r_\mathcal{D}(f) - R_N(f) \\ &\quad \leq 2c\gamma \,\text{Rad}_S(\mathcal{F}) + 3c\sqrt{\log(2/\delta)/(2N)} \end{aligned} \tag{12}$$

such that our generalization error bound scales as $O(\log(d_{\text{max}}))$ in dimension (as $\text{Rad}_S(\mathcal{F})$ is $O(\log(d_{\text{max}}))$).

We now introduce an analogous result for SRRs. In this case, the empirical risk we will use is the MC estimate $R^{\text{statMC}}_{N,C}$ in (9) since this is what we actually compute.

Theorem 2. *Suppose ϕ is bounded in $[0, c]$, and γ-Lipschitz in the first argument. For $m \in \{1, ..., C\}$, define $S'_m =$*

$\{(x'_{1,m}, y_1), ..., (x'_{N,m}, y_N)\}$, *such that it contains the* m*-th perturbed point from each of the* N *original input points. For any* $\delta \in (0, 1)$*, with probability at least* $1 - \delta$*, the following holds for all* $f \in \mathcal{F}$*:*

$$r^{stat}_{\mathcal{D}}(f) - R^{statMC}_{N,C}(f) \leq 2c\gamma\,\overline{\mathrm{Rad}_{S'}(\mathcal{F})} + 3c\sqrt{\log(2/\delta)/(2N)} \tag{13}$$

where

$$\overline{\mathrm{Rad}_{S'}(\mathcal{F})} \triangleq \frac{1}{C}\sum_{m=1}^{C} \mathrm{Rad}_{S'_m}(\mathcal{F}) \tag{14}$$

Proof. See Appendix A. □

Thus the SRR generalization error is upper bounded by an expression that varies as $O(\log(d_{\max}))$.

In contrast, for the adversarial risk, where (in binary classification) $L^{adv}(X, Y, f) \triangleq \max_{\delta\in\Delta} \phi(f_\theta(X + \delta), Y)$, $\mathrm{Rad}_S(L^{adv}_{\mathcal{F}})$ is ***lower bounded*** by an expression containing explicit dependence on $\sqrt{d_{in}}$, where d_{in} is the dimension of the input layer to the NN [Yin et al., 2019]. While this lower bound does not allow us to directly bound the generalization error using (11), it does suggest that in high dimensions the adversarial generalization error can be much greater than the natural and statistically robust generalization error. This indicates it will typically be difficult to train networks that are adversarially robust at test time for high-dimensional datasets. Our analysis thus shows that statistically robust networks may be easier to obtain.

5 RELATED WORK

Probabilistic robustness Compared to the vast body of work on adversarial metrics for neural network robustness, there has been relatively little work examining robustness to probabilistic perturbations. Fawzi et al. [2018] introduced a risk metric based on finding the largest possible uniform perturbation distribution that still maintains a target level of accuracy. Hendrycks and Dietterich [2019] experimentally evaluated different network architectures by averaging their accuracy over a discrete set of common image corruptions. Weng et al. [2019] and Webb et al. [2019], suggested probabilistic metrics for ***pointwise*** robustness based on verification and statistical sampling approaches respectively. Our work extends these ideas to provide a comprehensive robust risk framework that applies to the whole network and which can be used for ***training*** networks.

Use of noise in achieving adversarial robustness Some recent papers [Zantedeschi et al., 2017, Li et al., 2019, Cohen et al., 2019] have examined the use of noise/random corruptions as a mechanism for achieving adversarial robustness. For instance, randomized smoothing [Gilmer et al., 2019] can be used to harden modes against adversarial attack post-hoc with guarantees. Our work instead focuses on statistical robustness as the goal in its own right.

Distributional shift Defining metrics for—and obtaining classifiers robust to—distributional shift from train to test is a related problem [Quionero-Candela et al., 2009, Duchi and Namkoong, 2018, Lipton et al., 2018]. We instead are not assuming uncertainty in the population distribution, but that individual datapoints are probabilistically corrupted.

Data augmentation for generalization Training neural networks with randomly perturbed inputs is, of course, not a new concept. Many works examine this form of data augmentation as a means for improving generalization [Elman and Zipser, 1988, An, 1996]. Other work has investigated training neural networks by perturbing other components of the neural network such as weights [An, 1996, Graves et al., 2013], targets [Szegedy et al., 2016, Vaswani et al., 2017], and gradients [Neelakantan et al., 2015], with similar motivations. Chapelle et al. [2001] introduced an empirical metric—vicinal risk—as a means to better approximate the true natural risk by using a kernelized density estimate for the data distribution $p_{\mathcal{D}}$, rather than just taking the standard MC approximation (empirical risk). This leads to training schemes equivalent to randomly perturbing the inputs.

Our work differs from these in that training with random perturbations emerges from a principled risk minimization framework, rather than being taken as the starting point of algorithmic development. Moreover, we use input perturbations not only during training but also as a means of evaluating the robustness at ***test–time***. We have also drawn novel connections and comparisons between existing adversarial/robustness methods and probabilistic input perturbations, providing conceptual, theoretical, and empirical arguments for why the latter is an important component in the greater arsenal of robust classification approaches.

6 EXPERIMENTS

To empirically investigate our SRR framework, we now present experiments comparing it with natural and adversarial approaches. For SRR training, we follow the approach from Section 3.4, generating perturbations X'_n to points X_n in the training dataset and then using a mini-batch version of the gradient update in (10). Unless otherwise stated, we train using the cross-entropy loss, $\phi^{\mathrm{CE}}(f_\theta(X'), Y) \triangleq -\log(f_\theta(X')_Y)$, referring to training on the resulting SRR as corruption training. Analogously to testing on accuracy in natural settings, we evaluate using the TSRM , i.e. (5).

For Experiments 6.1 and 6.3, we used a dense ReLU network architecture with one hidden layer, while for Experiment 6.2, we use a wide residual network architecture [Zagoruyko and Komodakis, 2016]. Full details are provided in Appendix E.

Table 1: Train/test set evaluations of different networks on CIFAR-10, scores given in % and averaged over 5 runs. The best test set performance for each evaluation metric is highlighted in bold.

		Training Method			
		Natural, $\epsilon = 0$	Corruption $\epsilon = 0.157$	Corruption $\epsilon = 0.5$	PGD $\epsilon = 0.157$
Evaluation Metric	Natural, $\delta = 0$	98.7/**92.9**	98.0/92.2	93.7/87.9	96.3/88.1
	TSRM, $\delta = 0.157$	94.9/89.4	98.1/**92.4**	94.1/88.1	96.3/88.1
	TSRM, $\delta = 0.5$	60.0/57.6	79.9/76.0	95.6/**89.9**	94.9/86.1
	Adversarial, $\delta = 0.157$	0.0/0.0	0.2/0.2	3.1/3.0	67.3/**40.1**

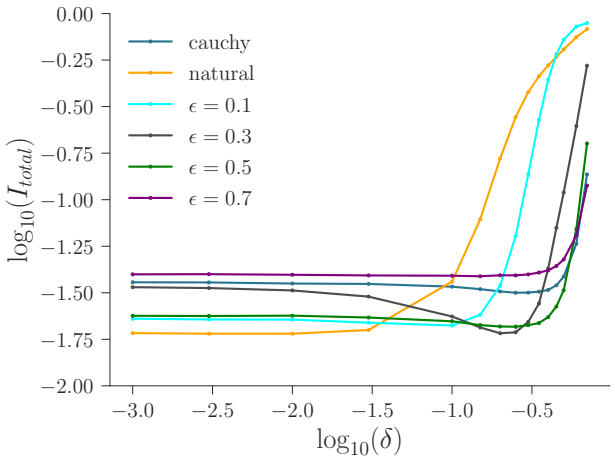

Figure 1: TSRM computed on the MNIST test set for different uniform perturbations ϵ. Each line represents a different training objective. Results are averaged over 5 runs (error bars are imperceptibly small).

6.1 COMPARISON TO NATURAL ACCURACY

First, we show that naturally trained networks are vulnerable under the TSRM metric and that corruption training can alleviate this. We train separate networks using 6 methods: corruption training with a Cauchy distribution with scale $\gamma = 0.5$, corruption training with the uniform perturbations over radius-ϵ L_∞ balls ($\epsilon = 0.1, 0.3, 0.5, 0.7$), and natural training ($\epsilon = 0$). We evaluate these networks using natural accuracy, and TSRM with uniform perturbation distributions on radius-δ L_∞ balls with δ from 10^{-3} to 0.7.

The results, shown in Figure 1, provide several interesting insights. Firstly, as expected, networks corruption trained with more severe perturbations (larger ϵ) performed better when evaluated on more severe perturbations (larger δ), though this comes at the cost of a lower natural accuracy. Secondly, these gains are often more than an order of magnitude in size, confirming that TSRM can be highly distinct from natural accuracy ($\delta = 0$), and corruption training can provide significant benefits under this robust metric. Finally, training with a qualitatively distinct distribution (Cauchy) provided decent performance when evaluated on TSRM with uniform perturbations, supporting our intuition in Section 3.5.

6.2 EMPIRICAL GENERALIZATION ERROR

As previously noted, it has proved challenging to train networks to achieve high test-time adversarial accuracy on higher-dimensional datasets such as CIFAR-10 due to poor generalization from training. By contrast, our analysis in Section 4 suggests that the gap will be more similar to natural accuracy for SRR approaches. We thus investigate the generalization gap experimentally for TSRM on CIFAR-10. Additionally, we compare corruption training with the PGD adversarial training method of Madry et al. [2018], which is designed to maximize adversarial accuracy.

We train using four different methods: natural training, corruption training with $\epsilon = 0.157$ and $\epsilon = 0.5$, and PGD adversarial training (7 gradient steps) with $\epsilon = 0.157$. Correspondingly, we then evaluate these networks using natural accuracy, TSRM with $\delta = 0.157$ and $\delta = 0.5$, and adversarial accuracy with $\delta = 0.157$ (computed using 7-step PGD). Here 0.157 corresponds roughly 8/255 in pixel values, which is used as the corruption set by Madry et al. [2018] for adversarial training, while 0.5 represents a more extreme corruption model.

The results in Table 1 demonstrate generalization performance in line with our theoretical analysis: the natural/TSRM evaluation metrics have fairly small generalization gaps (up to about 8%), while we see a much larger 27.2% gap for adversarial accuracy (on the PGD trained network). To reiterate, this is a limitation of adversarial risk compared to SRR, ***regardless of the training method***.

Regarding training methods, for the natural and TSRM metrics, we notice that the best test-set performance was achieved using the corresponding natural/corruption training method. As can be expected, corruption training does not perform well on adversarial risk, since it targets SRR rather than adversarial risk. However, as discussed previously, adversarial risk is not a relevant or accurate metric to use for probabilistic perturbations. We also see that PGD training is fairly effective for improving the TSRM, recording con-

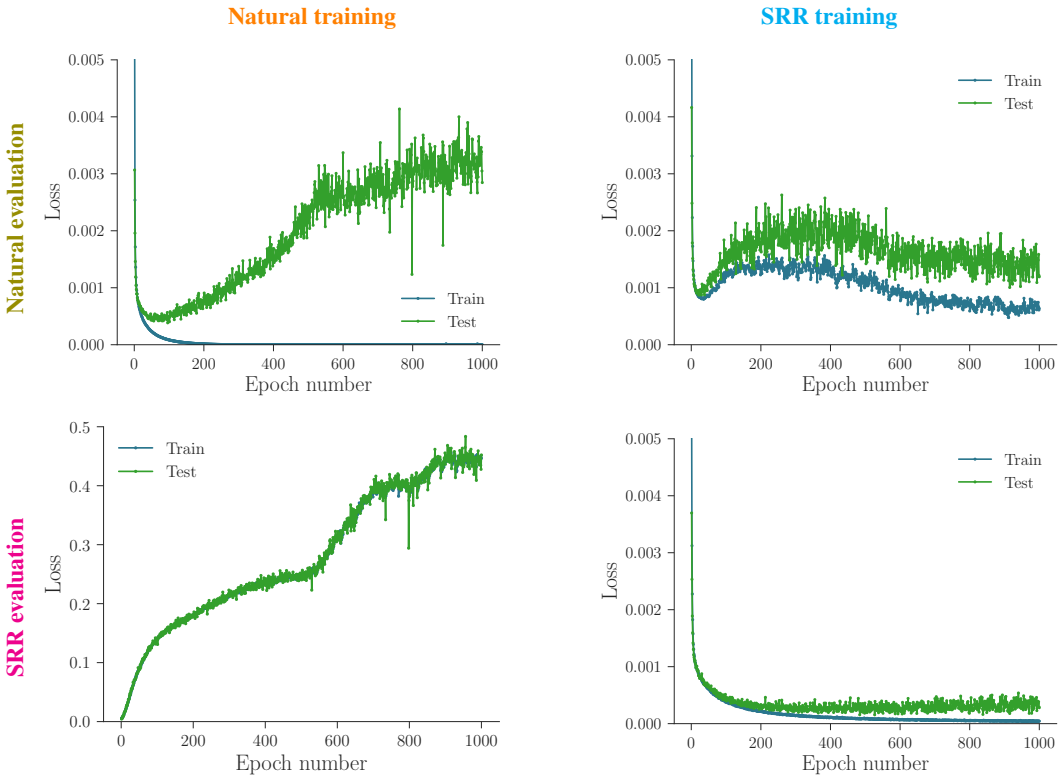

Figure 2: Learning curves using weighted cross-entropy loss on MNIST with $w(8) = 100$. Each plot represents a combination of training method and evaluation metric. For the bottom left plot, note the different y-axis scaling, and that the training and test SRR almost exactly overlap.

sistently good test-set TSRM for all values of δ. However, there is still a consistent gap of $3-4\%$ between PGD training and corruption training on these statistical risk metrics. This shows that adversarial risk can indeed be non-optimal for average-case performance, as we argued in Section 3.1 (even if it can be a reasonable approximation). Further, PGD training incurs significant additional computational expense ($\sim$6 times slower). We thus see that corruption training can be the better choice due to its simplicity and efficiency, when we are concerned with probabilistic perturbations.

6.3 TAILORED LOSS FUNCTIONS

In risk frameworks, we often wish to tailor the loss function ϕ to better represent a particular problem, such as a safety property. For example, a self-driving car predicting the road is clear when there is actually a pedestrian will be far more damaging than predicting there is a pedestrian when the road is clear. The SRR can be particularly useful in such situations, as networks need to be robust to noise in their inputs to fully incorporate all uncertainty present in the decision making process. To demonstrate this, we consider training and evaluating using a weighted cross-entropy loss

$$\phi(f_\theta(X'), Y) = -w(Y) \log f_\theta(X')_Y. \qquad (15)$$

By taking $w(Y^*) >> 1$ for a particular problem class Y^* and $w(Y) = 1$ for others, the classifier will be heavily penalized if it fails to correctly identify with high-confidence all occurrences of Y^*. In turn, this heavy penalty can increase the sensitivity to perturbations in the inputs: the classifier should not confidently predict that $Y \neq Y^*$ if our input is close to points for which $Y = Y^*$, as this risks incurring the penalty if our inputs are noisy or our classifier is imperfect.

To make comparisons, we trained on MNIST using the same network as in Experiment 6.1, but with this weighted CE loss where $w(8) = 100$ and $w(Y) = 1$ otherwise, i.e. penalizing classifiers which fail to confidently identify images of the number 8. We also use a Gaussian perturbation distribution, taking $p(X'|X) \sim \mathcal{N}(X, \sigma^2 I)$ with $\sigma = 0.3$.

The results, shown in Figure 2, exhibit several interesting traits. Firstly, we see that natural training is extremely vulnerable to noisy input perturbations (bottom left), producing SRR values at both train and test time that are multiple orders of magnitude worse than those achieved when corruption training (bottom right). This highlights both the importance of considering noisy inputs at test-time, and also the ability of SRRs to provide effective robust training.

Secondly, we see that while training with natural risk quickly overfits (top left), corruption training with the SRR provides

far better generalization (bottom right). In fact, the final test SRR of the corruption trained network is lower than the final test natural risk on the natural risk trained network, a powerful result given that the former evaluation metric is a corrupted version of the latter. Thus even when the inputs are not corrupted, we can achieve a better loss by artificially adding noise to both the training procedure **and** at test-time. This indicates that for the weighted loss, the SRR can provide robustness not only by accounting for potential input noise, but also by better accounting for the imperfect nature of the network to avoid overconfidently dismissing the potential for a test-time datapoint to belong to the problem class.

7 CONCLUSIONS

Motivated by applications where test-time corruptions are not generated adversarially but probabilistically, we introduced a statistically robust risk (SRR) framework, providing a class of metrics for evaluating robust performance under probabilistic input perturbations that are amenable to efficient training. We showed that SRRs can differ significantly from both natural and adversarial risk, and that networks with low test-time SRRs can be achieved through training with corrupted inputs. Unlike adversarial risk, our results suggest that SRRs generalize from the training data similarly to, and potentially even better than, natural risks, meaning that they have more general practical applicability to high-dimensional datasets and complex architectures. Thus, for probabilistic corruption threat settings, robust neural networks may be within reach for a wide range of applications.

Acknowledgements

TR gratefully acknowledges funding from Tencent AI Labs and a Junior Research Fellowship supported by Christ Church, Oxford.

References

Guozhong An. The effects of adding noise during backpropagation training on a generalization performance. *Neural computation*, 8(3):643–674, 1996.

Peter L. Bartlett, Dylan J. Foster, and Matus Telgarsky. Spectrally-normalized margin bounds for neural networks. In *Proceedings of 31st International Conference on Neural Information Processing Systems (NeurIPS)*, pages 6241–6250, 2017.

A. Ben-Tal, L. El Ghaoui, and A.S. Nemirovski. *Robust Optimization*. Princeton Series in Applied Mathematics. Princeton University Press, October 2009.

Olivier Chapelle, Jason Weston, Léon Bottou, and Vladimir Vapnik. Vicinal risk minimization. In *Proceedings of 13th International Conference on Neural Information Processing Systems (NeurIPS)*, pages 416–422, 2001.

Jeremy Cohen, Elan Rosenfeld, and Zico Kolter. Certified adversarial robustness via randomized smoothing. In *Proceedings of the 36th International Conference on Machine Learning*, pages 1310–1320, 2019.

Dimitrios I. Diochnos, Saeed Mahloujifar, and Mohammad Mahmoody. Adversarial risk and robustness: General definitions and implications for the uniform distribution. In *Proceedings of 32nd International Conference on Neural Information Processing Systems (NeurIPS)*, pages 10380–10389, 2018.

John Duchi and Hongseok Namkoong. Learning Models with Uniform Performance via Distributionally Robust Optimization. *arXiv e-prints*, art. arXiv:1810.08750, Oct 2018.

Jeffrey L Elman and David Zipser. Learning the hidden structure of speech. *The Journal of the Acoustical Society of America*, 83(4):1615–1626, 1988.

Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. Analysis of classifiers' robustness to adversarial perturbations. *Machine Learning*, 107(3):481–508, Mar 2018.

Timon Gehr, Matthew Mirman, Dana Drachsler-Cohen, Petar Tsankov, Swarat Chaudhuri, and Martin Vechev. AI^2: Safety and robustness certification of neural networks with abstract interpretation. In *Security and Privacy (SP), 2018 IEEE Symposium on*, 2018.

Justin Gilmer, Nicolas Ford, Nicholas Carlini, and Ekin Cubuk. Adversarial examples are a natural consequence of test error in noise. In *Proceedings of the 36th International Conference on Machine Learning*, pages 2280–2289, 2019.

Ian Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *Proceedings of 3rd International Conference on Learning Representations (ICML)*, 2015.

Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. Speech recognition with deep recurrent neural networks. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 6645–6649. IEEE, 2013.

Shixiang Gu and Luca Rigazio. Towards deep neural network architectures robust to adversarial examples. In *Proceedings of 3rd International Conference on Learning Representations (ICLR)*, 2015.

Dan Hendrycks and Thomas G. Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *Proceedings of the 7th International Conference on Learning Representations (ICLR)*, 2019.

Alex Kurakin, Dan Boneh, Florian Tramèr, Ian Goodfellow, Nicolas Papernot, and Patrick McDaniel. Ensemble adversarial training: Attacks and defenses. In *Proceedings of 6th International Conference on Learning Representations (ICML)*, 2018.

Michel Ledoux and Michel Talagrand. *Probability in Banach spaces: isoperimetry and processes*. Springer Science & Business Media, 2013.

Bai Li, Changyou Chen, Wenlin Wang, and Lawrence Carin. Certified adversarial robustness with additive noise. In *Proceedings of 32nd Conference on Neural Information Processing Systems (NeurIPS)*, pages 9459–9469, 2019.

Zachary Lipton, Yu-Xiang Wang, and Alexander Smola. Detecting and correcting for label shift with black box predictors. In *Proceedings of the 35th International Conference on Machine Learning*, pages 3122–3130, 2018.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *Proceedings of 6th International Conference on Learning Representations (ICLR)*, 2018.

Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of Machine Learning*. The MIT Press, 2012. ISBN 026201825X.

Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. DeepFool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2574–2582, 2016.

Arvind Neelakantan, Luke Vilnis, Quoc V Le, Ilya Sutskever, Lukasz Kaiser, Karol Kurach, and James Martens. Adding gradient noise improves learning for very deep networks. *arXiv preprint arXiv:1511.06807*, 2015.

Nicolas Papernot, Patrick McDaniel, Somesh Jha, Matt Fredrikson, Z Berkay Celik, and Ananthram Swami. The limitations of deep learning in adversarial settings. In *Proceedings of the IEEE European Symposium on Security and Privacy (EuroS&P)*, pages 372–387. IEEE, 2016.

Joaquin Quionero-Candela, Masashi Sugiyama, Anton Schwaighofer, and Neil D. Lawrence. *Dataset Shift in Machine Learning*. MIT Press, 2009.

Ludwig Schmidt, Shibani Santurkar, Dimitris Tsipras, Kunal Talwar, and Aleksander Madry. Adversarially robust generalization requires more data. In *Proceedings of 32nd International Conference on Neural Information Processing Systems (NeurIPS)*, pages 5019–5031, 2018.

Lukas Schott, Jonas Rauber, Matthias Bethge, and Wieland Brendel. Towards the first adversarially robust neural network model on MNIST. In *Proceedings of the 7th International Conference on Learning Representations (ICLR)*, 2019.

Shai Shalev-Shwartz and Shai Ben-David. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, New York, NY, USA, 2014.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *Proceedings of 2nd International Conference on Learning Representations (ICLR)*, 2014.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of 31st Conference on Neural Information Processing Systems (NeurIPS)*, pages 5998–6008, 2017.

Shiqi Wang, Kexin Pei, Justin Whitehouse, Junfeng Yang, and Suman Jana. Efficient formal safety analysis of neural networks. In *Proceedings of 32nd International Conference on Neural Information Processing Systems (NeurIPS)*, pages 6369–6379, 2018.

Stefan Webb, Tom Rainforth, Yee Whye Teh, and M. Pawan Kumar. A Statistical Approach to Assessing Neural Network Robustness. In *Proceedings of 7th International Conference on Learning Representations (ICLR)*, 2019.

Lily Weng, Pin-Yu Chen, Lam Nguyen, Mark Squillante, Akhilan Boopathy, Ivan Oseledets, and Luca Daniel. PROVEN: Verifying robustness of neural networks with a probabilistic approach. In *Proceedings of the 36th International Conference on Machine Learning*, pages 6727–6736, 2019.

Eric Wong and J. Zico Kolter. Provable defenses against adversarial examples via the convex outer adversarial polytope. In *Proceedings of 35th International Conference on Machine Learning (ICML)*, pages 5283–5292, 2018.

Dong Yin, Ramchandran Kannan, and Peter Bartlett. Rademacher complexity for adversarially robust generalization. In *Proceedings of 36th International Conference*

on Machine Learning (ICML), volume 97, pages 7085–7094, 2019.

Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2016.

Valentina Zantedeschi, Maria-Irina Nicolae, and Ambrish Rawat. Efficient defenses against adversarial attacks. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, page 39–49, 2017.