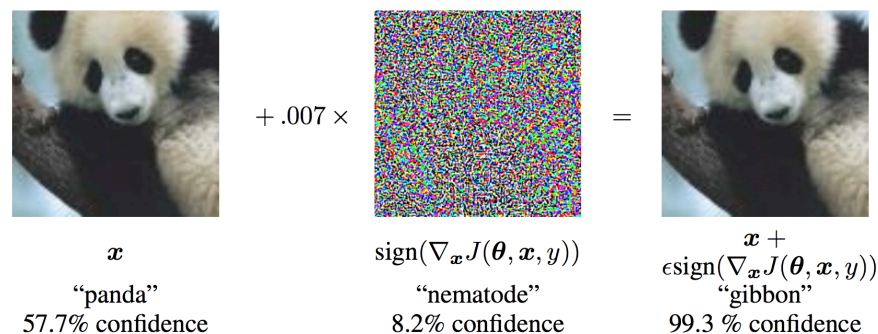


Efficient Defence-Dominant Adversarial Robustness using Moving Target Defence.

Ahmed Ghoor and Stephen Fowler

Adversarial Machine Learning



Adversarial Machine Learning is a field that attempts to get machine learning models to misclassify input data. This has been extensively studied within Computer Vision. A common approach is to perturb the pixels of an image using methods such as FGSM and PGD so that the changes are imperceptible to humans but cause the image to be misclassified by the targeted classification model. A canonical example is shown in the above figure, where, through perturbing the pixels of an image of a panda slightly, the resulting image is misclassified by the model to be a gibbon with very high confidence [Cleverhans].

Relevance to AI Safety

Adversarial ML can be applied to many other applications, such as evading spam email classifiers, network intrusion detectors, as well as classification tools used in self-driving cars and autonomous web agents. Hacking any of these systems that have been deployed in critical infrastructure could be catastrophic.

As AI systems become more powerful and dual-use automation penetration testing tools become increasingly accessible, the number and intensity of cyber attacks could likely increase drastically.

There could also be an increasing disparity between the computational power available to the smallest ML models used in consumer devices and on edge devices and the computational power available to an attacker, making the discovery of defense-favouring techniques crucial.

Adversarial Training

Existing adversarial training approaches often red team the various attacks that an attacker could use and then retrain their model to account for these attacks.

However, as the variety of attacks increase, accounting for these attacks can be computationally expensive which makes it infeasible to implement them all using an ensemble of different methods. For example, network intrusion detectors process so much internet traffic that needs to be classified in real time that classifiers have to be very efficient.

The tradeoff with only using one or two classifiers is that a different attack could easily bypass them.

Moving Target Defence

In moving target defense, the model used to classify incoming packets is constantly stochastically switched. This has the effect of altering the decision boundary in feature space for the classification problem, mitigating attacks which attempt to learn the boundary or optimise for a particular set of defenses.

It essentially preserves many of the benefits of the a defence-in-depth ensemble approach, but at a much lower computational cost. It also has the added benefit of preventing the decision boundary of the ensemble approach being learnt if the attack has temporary access to the target before the official attack.

Some use cases where this could be useful:

Phishing Emails: Phishing emails to employees in major companies are often the initial access point for an attacker to access a system before escalating their privileges to cause havoc. Monitoring all emails, however, can be computationally expensive for very large organisations and spam classifiers are often used to filter out emails. However, these models like others, can easily be evaded.

Network Intrusion Detection: A domain that particularly deals with a lot of data that could potentially be malicious is network intrusion detection systems (NIDS). Networks deal with huge amounts of data every second that NIDS need to monitor for a variety of different cyber attacks.

Autonomous Agents: Autonomous agents increasingly interact in the real world and draw on tools for classification, including the ones above, before making decisions about its actions. This could apply to an agent surfing the web classifying screenshots of a web page, a self-driving car classifying road signs, or an autonomous cyber defence system classifying network traffic to monitor for intrusions and taking autonomous actions from there.

Key Questions

Defense-dominance

By randomly selecting a single classification model at runtime, MTD does not impose additional time costs for the defender no matter how many additional models are added, aside from the cost of stochastically switching between models. For the attacker, they are forced to find adversarial attacks that work all models if they want to guarantee attack success, and this task increases in complexity as more models added by the defender. This is often very difficult as attacks are often constrained by the number of and scale of the perturbations made to evade the classifier while simultaneously preserving the core functionality, such as the similarity to the original image in computer vision and preserving the malicious behaviour of cyber attacks.

The task of finding an attack is also made more difficult by the stochasticity of switching between models, as it makes it harder for the attacker to learn and optimise against one decision boundary when creating adversarial examples.

Scale (Relevance)

As AI systems for attack get more powerful and new systems are created, they could often also be used for attack. However MTD has the added benefit of increasing the cost and complexity on the attacker at a significantly greater rate.

Scale (Financially)

Moving target defense is a general principle that can be applied to a number of different domains and can form a general tool for security applications, and is not limited to non-profit AI safety domains.

Prototype

We prototyped this concept using a data set of emails that are classified as being spam or not spam.

We applied a couple of attacks against the models, and then retrained them to improve their robustness.

We then compared this to both an MTD and Ensemble approach, comparing both the accuracy and time for computation.

Results

We show that attacks trained defend against one attack perform poorly on other attacks, the MTD approaches drastically improves this and that it is slightly less accurate than the Ensemble approach but significantly faster.

Future work could do more significantly more testing with other datasets and additional models.

Code

<https://colab.research.google.com/drive/1bKpjyrqQIQ-hz6aPbAmHANllyyMJMg8-H?usp=sharing>