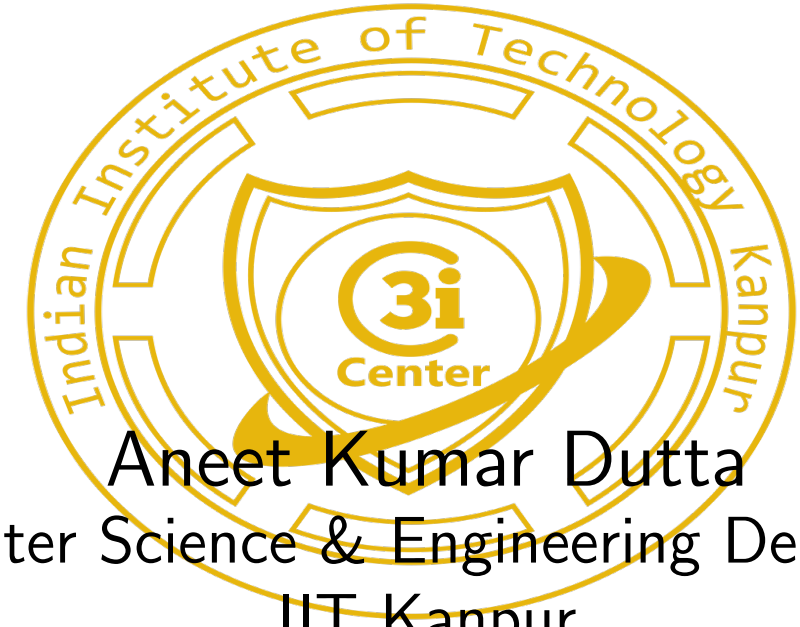

Aneet Kumar Dutta
Computer Science & Engineering Department
IIT Kanpur
08/03/2019

Introduction

- Machine Learning is widely used in security sensitive applications like Intrusion Detection Systems, Malware Analysis, Phishing Email Detection
- Success of this machine learning algorithms depends on their resistance to adversarial examples
- Adversarial agent can actively manipulate the samples in order to avoid detection unlike classical machine learning where underlying data distribution is stationary

Attack Types

- Poisoning Attack: Manipulating training data in different ways, the classifier is trained with wrong data so it mis-classifies attack points as legitimate points during test time
- Evasion Attack: Attempt to fool the classifier at test time by generating attack samples which will be classified as legitimate points.
  No manipulation in the training data or any weights in the training phase
- This presentation is on evasion attack based on the paper "Evasion Attacks against Machine Learning at Test Time" by Batista B Biggio.

Overview

- Evasion attack by exploiting gradient-based approach.
- Risk of the classifier is directly proportional to the attacker's knowledge about the machine learning model
- Degradation in classifier performance under adversarial examples is used as performance measuring index which will help in more informed model selection

I

Security Aspects in Machine Learning

- finding potential vulnerabilities of learning before they are exploited by adversary
- evaluating classifier security by the measure of degradation in classifier performance under adversarial examples
- devising appropriate countermeasures if an attack is found to significantly degrade classifier's performance

Approaches in Adversarial Learning

- Min-Max approach: The learner's and attacker's loss function are exactly opposite and yields to simple optimization problem.
- Game-Theoretic Approach: The learner's and attacker's loss function are not exactly opposite, like if an attacker wants to know if his generated samples are actually attack.
- However, in realistic constraints existing game theoretic approaches are too complex and multi-faceted

Optimal Evasion at Test Time(Problem Setting)

- Classification algorithm $f : X -> Y$ represents a function f which maps a sample in some feature space $x \epsilon X$ to a set of predefined classes $y \epsilon Y$. Here, $Y = \{-1, +1\}$, where -1 represents the normal class and $+1$ represents the malicious class.
- Training dataset $D = \{x_i, y_i\}_{i=1}^{n}$ sampled from the distribution $p(X, Y)$
- Label y^c refer to the label assigned by the classifier and y is the true label
- $g(x)$ is the discriminant function such that $f(x) = -1$ if $g(x) < 0$ and $+1$ otherwise.
  Eg:In linear case $g(x)$ can be $w^T x + b$

Adversary Model

Adversary Model explains:

- Adversary's goal
- Adversary's Knowledge
- Adversary's capability

Adversary's Goal

- Adversary wants to find an attack sample x which will be declared as a legitimate point such that $g(x) < -\epsilon$ for any $\epsilon > 0$ which results in $f(x) = -1$.
  This attacks can be easily detected by shifting the threshold ϵ.
- So, the adversary's goal is defined in terms of utility loss function which the adversary wants to minimize.
- In this paper, the attacker focuses on minimizing classifier's discriminant function $g(x)$

Adversary's Knowledge

The attacker's knowledge may include:

- Training set or part of the training set
- feature space
- type of learning algorithms used
- trained classifier model, eg: knowing the weights and hyper-parameters of the model
- feedback from the classifier

Depending upon the knowledge of adversary there can be two attack scenarios: Perfect Knowledge and Limited Knowledge

Perfect Knowledge

The attacker's knowledge include:

- Training set or part of the training set
- feature space
- type of learning algorithms used
- trained classifier model, eg: knowing the weights and hyper-parameters of the model

Limited Knowledge

The attacker's knowledge include:

- feature space
- type of learning algorithms used

Attack Strategy with Limited knowledge

- The adversary do not have learned classifier f and training data D, so it can not compute $g(x)$
- In this setting, $g(x)$ is approximated with $\hat{g}(x)$ and a surrogate dataset $D^{'} = \{\hat{x}_i, \hat{y}_i\}_{i=1}^{n_q}$ is used where $n_q << n$ and n_q is used as an hyper parameter by the attacker.
  $D^{'}$ belongs to the same distribution $p(X, Y)$ from which D belongs
- Since, the adversary wants to approximate targeted classifier f the adversary learns from y_i^c the label predicted by the classifier instead of true label $\hat{y}_i$

Forming the Optimization Problem

The optimization problem is:
$argmin_x\ F(x) = \hat{g}(x)$ such that $d(x, x^0) \leq d_{max}$

- The attacker tries to minimize $\hat{g}(x)$ and tries to find sample x which is within a maximum distance from a original attack data point x^0
- This constraint in the optimization problem is solved using Lagrange's multiplier.
- $\hat{g}(x)$ can be convex or non-convex function. Solving this problem we might land up in a local optima where the data distribution $p(x) = 0$
- $g(x)$ is a posterior estimate of $p(y^c = -1|x)$

Now the updated optimization problem is:
$argmin_x\ F(x) = \hat{g}(x) - \frac{\lambda}{n}\sum_{i|y^c=-1} k(\frac{x-x_i}{h})$ such that $d(x, x^0) \leq d_{max}$

- The problem of local optima is solved by using the additional term in the optimization problem, it acts as a penalizer for x in low density region.
- This additional term is equivalent to $p(x|y^c = -1)$ which is the likelihood.

Optimization Problem in Words

We know, $posterior = prior * likelihood$
$p(y^c = -1|x) = p(x) * p(x|y^c = -1)$
Taking log on both sides,
$logp(y^c = -1|x) = logp(x) + logp(x|y^c = -1)$
$logp(x) = logp(y^c = -1|x) - logp(x|y^c = -1)$

- We can consider $\hat{g}(x)$ is the estimate of $logp(y^c = -1|x)$ and the additional term is the estimate of the log-likelihood $logp(x|y^c = -1)$
- Solving this will give us $logp(x)$, thus we are able to get the data distribution $p(x)$

- The samples drawn from the distribution $p(x)$ belongs to the legitimate class and within a maximum distance d_{max} from the original attack samples.
- Thus, the attack samples generated by the adversary will evade the machine learning classifier because it belongs from the legitimate class and will be within a certain distance from a original attack data point which will ensure that the sample is an attack which the classifier will not be able to detect

Gradient Descent Attack

- The mentioned optimization problem is solved by using Gradient Descent approach.
- The sample x is found which will evade the learned classifier

Things to do ahead

- We can make our designed IDS resistance to adversarial examples
- We can design adversarial model which will evade well known supervised machine learning techniques for IDS
- We can try to evade the IDS based on the prediction error(lots of research paper on LSTM and RNN models)