Dimensionality reduction as a defense against evasion attacks on machine learning classifiers

Arjun Nitin Bhagoji and Prateek Mittal

Princeton University

DC-Area Anonymity, Privacy, and Security Seminar, Fall 2016

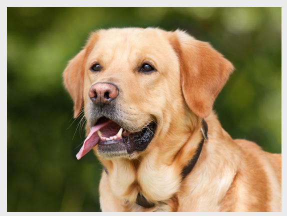
The Sixfold Path

1. Motivation
2. Machine learning, briefly
3. Adversaries and attacks
4. Defenses
5. Results
6. Ongoing Work and Extensions

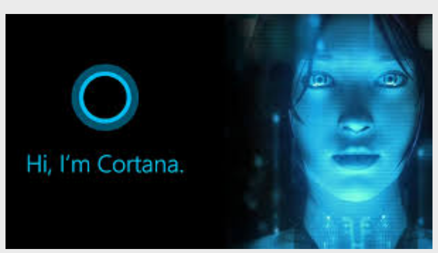
Motivation

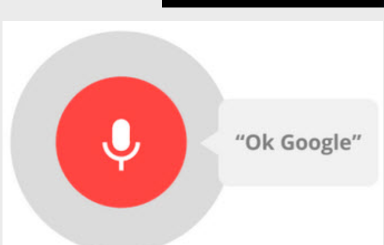
The Ubiquity of Machine Learning

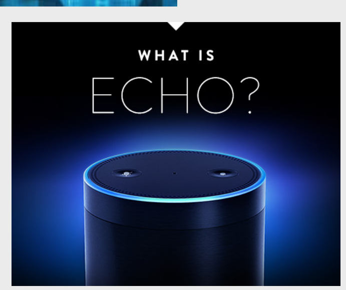
Images

Sound



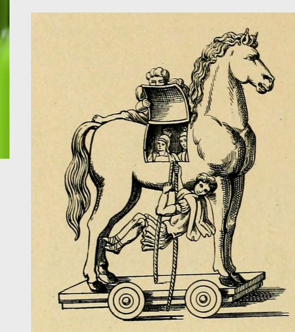


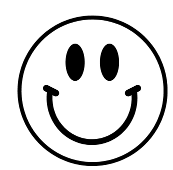


Machine Learning Systems

SVMs, Neural Networks, Random Forests,…

Malware

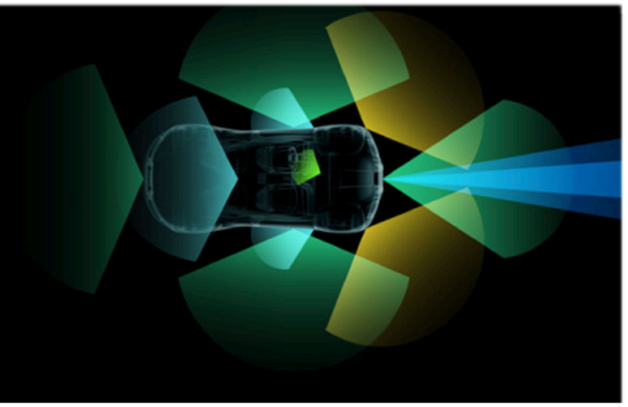





Cat Malicious
Dog Benign

“Who are you?”
“Open mail…”

Critical Applications of ML



SENSOR FUSION

DRIVE PX can fuse data from 12 cameras, as well as lidar, radar, and ultrasonic sensors. This allows algorithms to accurately understand the full 360 degree environment around the car to produce a robust representation, including static and dynamic objects. Use of Deep Neural Networks (DNN) for the detection and classification of objects dramatically increases the accuracy of the resulting fused sensor data.

Click here for a list of sensor partners.

COMPUTER VISION AND DEEP NEURAL NETWORK PIPELINE

DRIVE PX platforms are built around deep learning and include a powerful framework (Caffe) to run DNN models designed and trained on NVIDIA DIGITS™. DRIVE PX also includes an advanced computer vision (CV) library and primitives. Together, these technologies deliver an impressive combination of detection and tracking.

See the NVIDIA research paper End to End Learning for Self-Driving Cars that details how a convolutional neural network (CNN) was deployed on DRIVE PX enabling a self-driving car. Read the research paper.

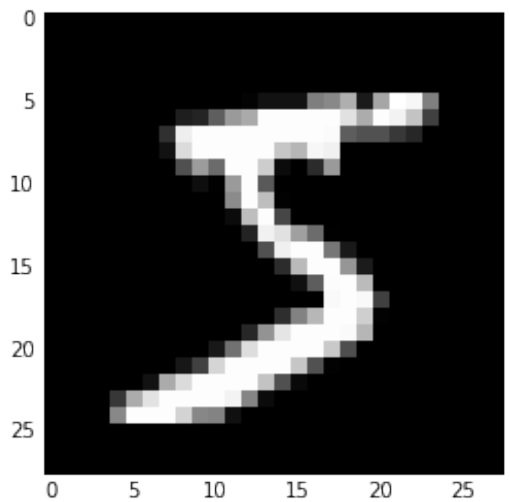
How PayPal beats the bad guys with machine learning

Credit: Shutterstock

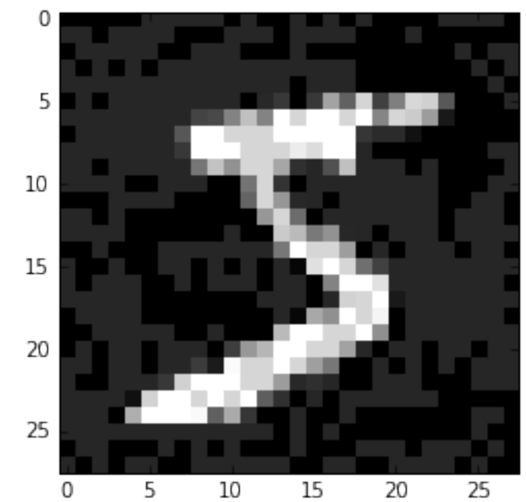
As big cloud players roll out machine learning tools to developers, Dr. Hui Wang of PayPal offers a peek at some of the most advanced work in the field

InfoWorld | Apr 13, 2015

Vulnerability of ML



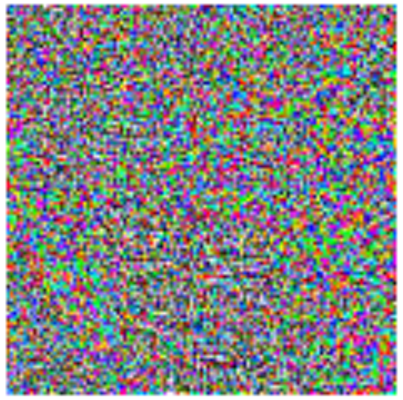
Modified by adversary →



Classified as 5

Classified as 0

$+ .007 \times$



$=$

$\boldsymbol{x}$	$\text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$	$\boldsymbol{x} + \epsilon\text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$
"panda"	"nematode"	"gibbon"
57.7% confidence	8.2% confidence	99.3 % confidence

Figure taken from 'Explaining and harnessing adversarial examples' by Goodfellow et. al.

Machine Learning, Briefly

Typical ML Pipeline

Training phase

Training Data Labels

Training data

Training algorithm

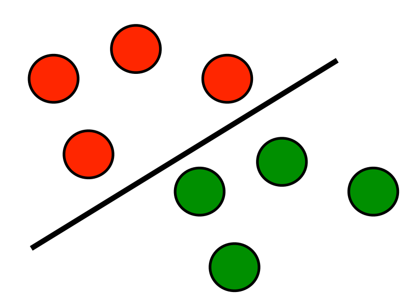
1. Starts with f_0 ,untrained classifier
2. Optimizes to find f , which labels most data correctly

Test phase

Test data

Trained ML Classifier

$y = f(x)$



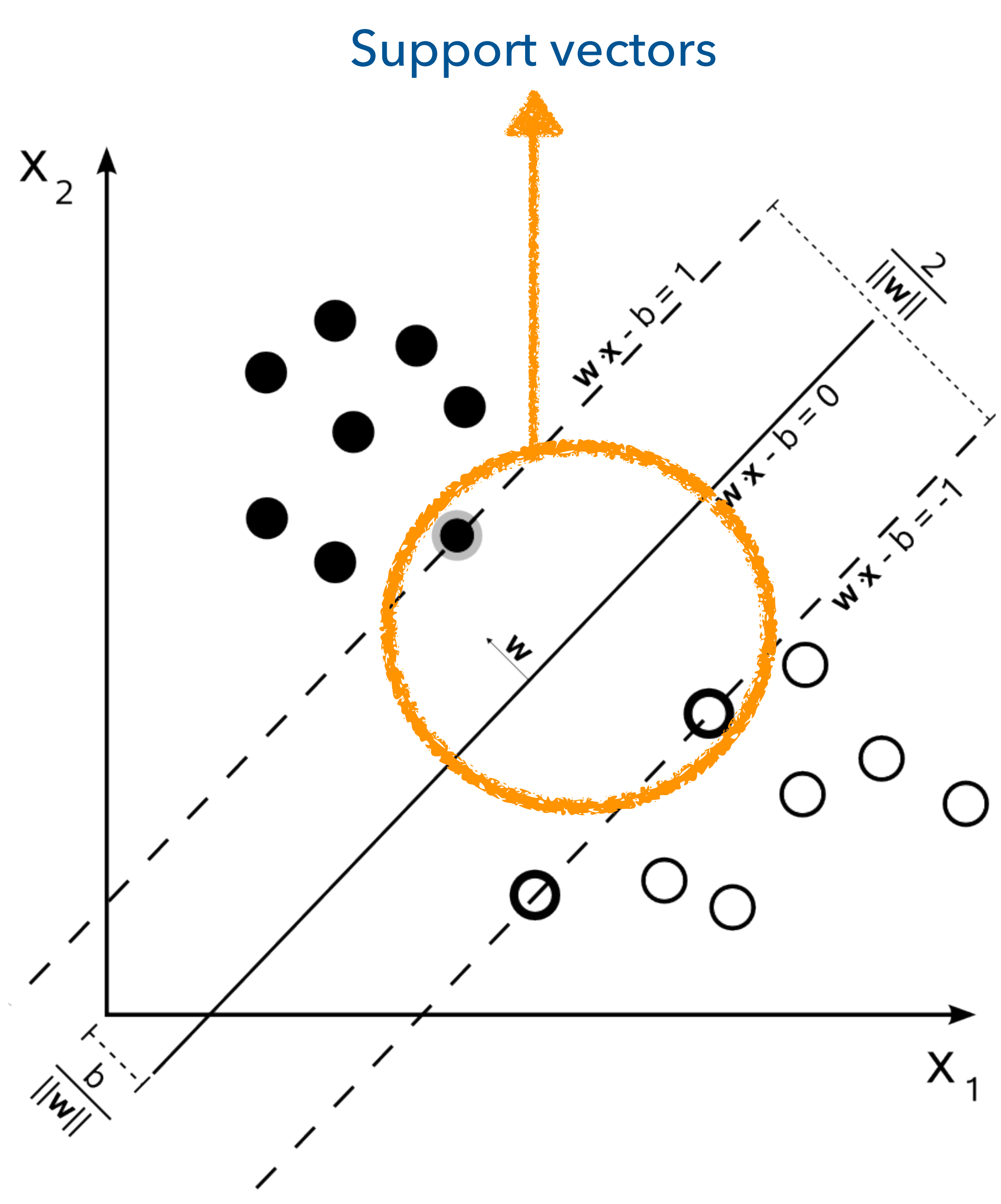
Test Data Labels

Verify

Predicted labels

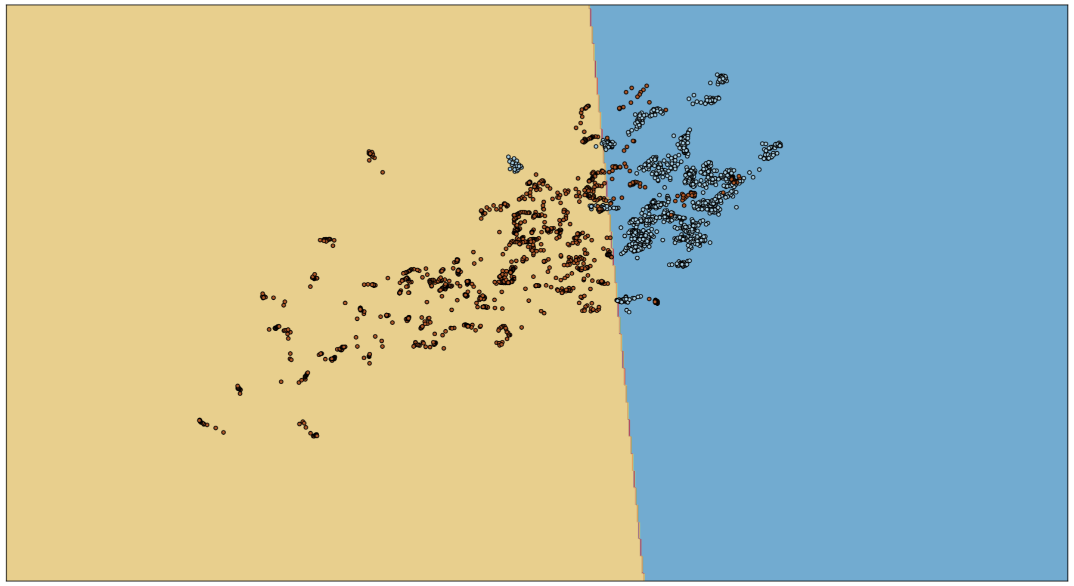
To find misclassification percentage

Support Vector Machines (SVMs)



Maximum margin separating hyperplane

Image courtesy: Wikimedia Foundation



Linear SVM on UCI Human Activity Recognition dataset

Sitting vs. Walking

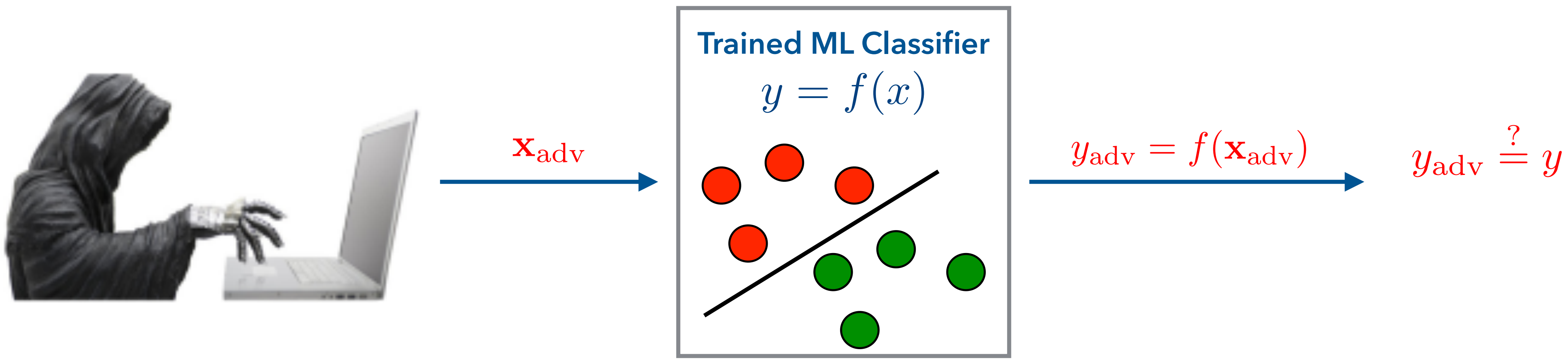
Margin: Distance between parallel hyperplanes separating data

Max. margin hyperplane: Halfway in between parallel hyperplanes

Adversaries and Attacks

Adversarial setup

During the test phase (or once deployed)…

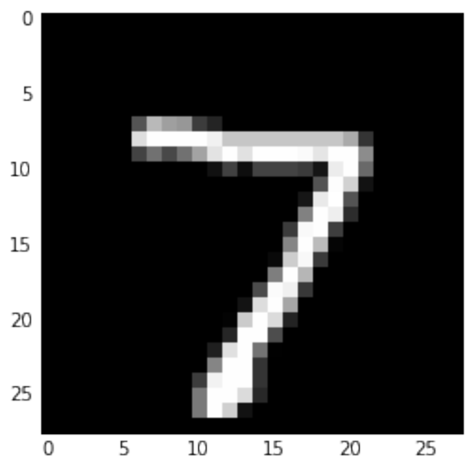


Minimally modifies legitimate inputs to induce misclassification at test time

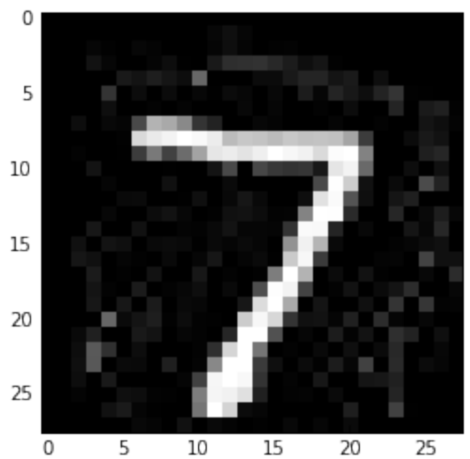
Assume powerful adversary has knowledge of trained classifier and input datasets

Previous work has shown black-box ML systems can be reverse engineered enough to carry out evasion attacks using queries

Evasion Attack on Linear SVM



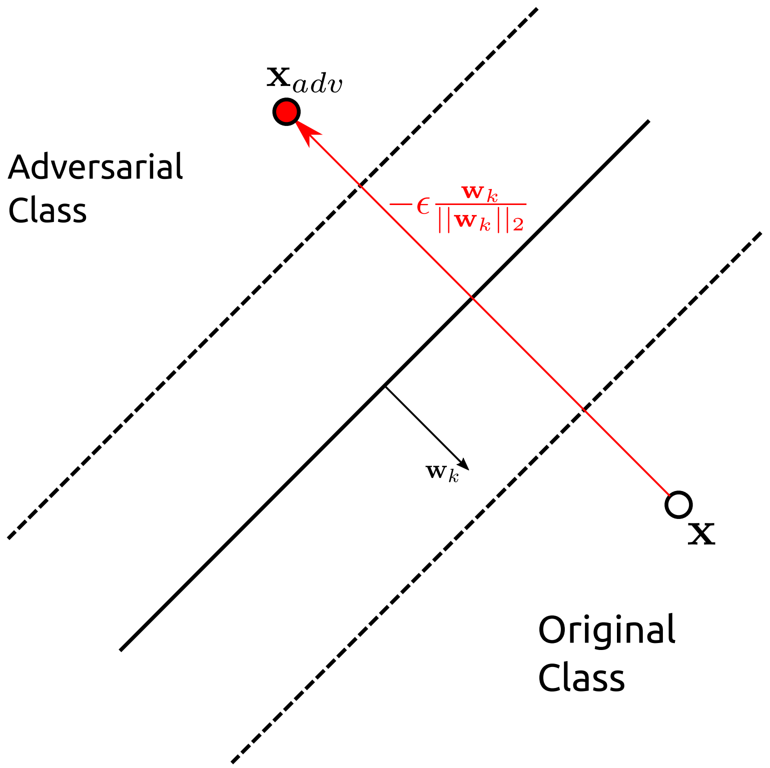
$\mathbf{x}$ Classified as 7



$\mathbf{x}_{adv}$ Classified as 3!

Adversarial image with ϵ =2.0.

Leads to 100% misclassification on test set.



$$\mathbf{x}_{adv} = \mathbf{x} - \epsilon \frac{\mathbf{w}_k}{\|\mathbf{w}_k\|_2}.$$

$$\epsilon \in [0, \infty)$$

Attack on Linear SVMs

ϵ controls the amount of perturbation added (typically small)

Not just Images…

Szegedy et. al. (2014),
Papernot et. al. (2015)

Xu et. al. (2016)

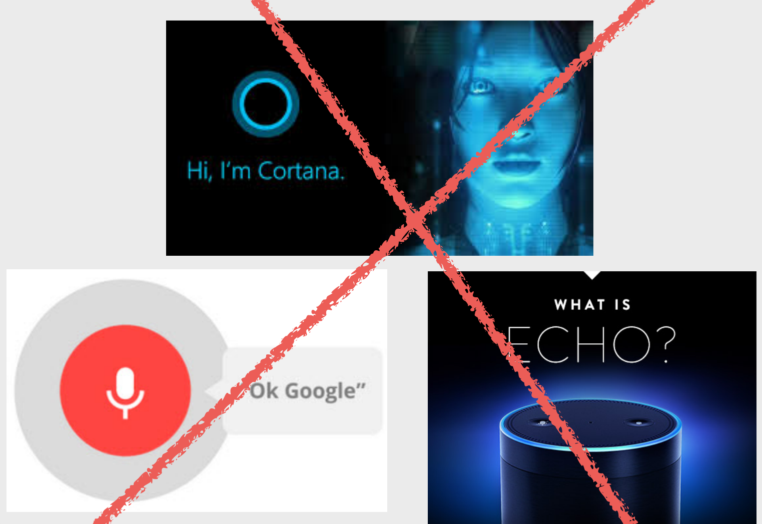
Images

Malware

Machine Learning Systems

SVMs, Neural Networks, Random Forests,…

Sound



Cat → Dog

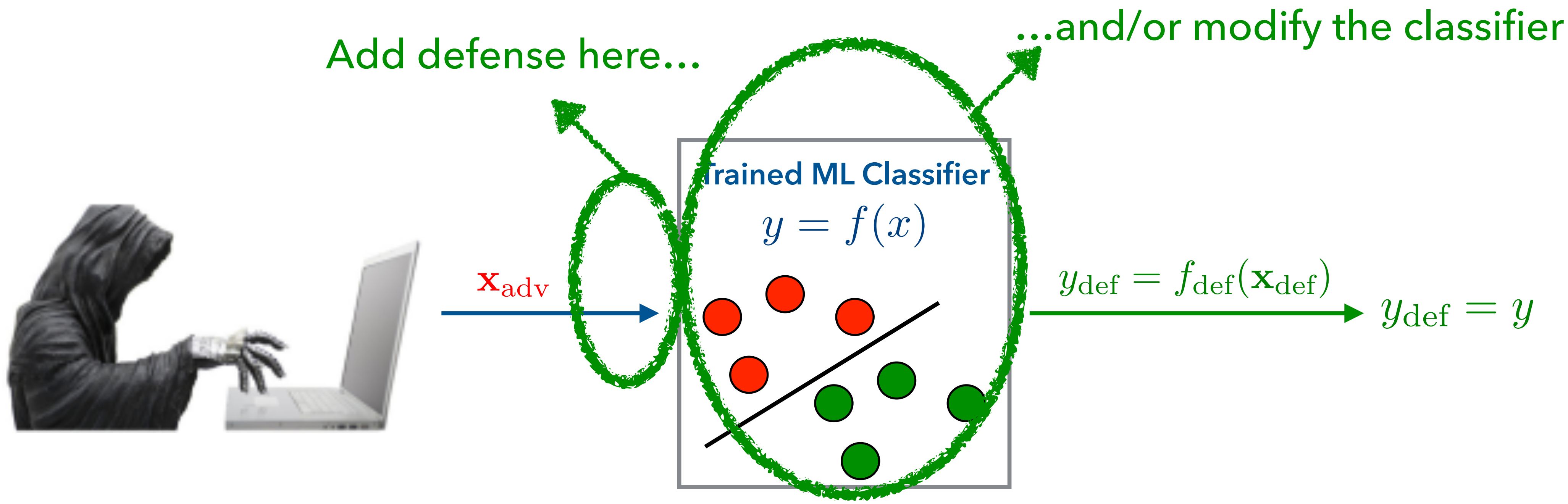
Malicious → Benign

"Who are you?" → "Open mail…"

Carlini et. al. (2016)

Defenses

Defense Desiderata



- Maintain classification accuracy (utility)
- Low efficiency overhead
- Improve security, i.e. resistance to adversarial samples
- Tunable, i.e. tradeoff utility, efficiency and security
- Effective in a range of settings

Limitations of Existing Defenses

- Focused on specific classifier families
- Resistance to adversary with knowledge of defense is unclear
- Only valid for specific attacks

Case in point

- Proposed defense for neural networks of Papernot et. al. (2015) broken by modified attack in Carlini et. al. (2016)

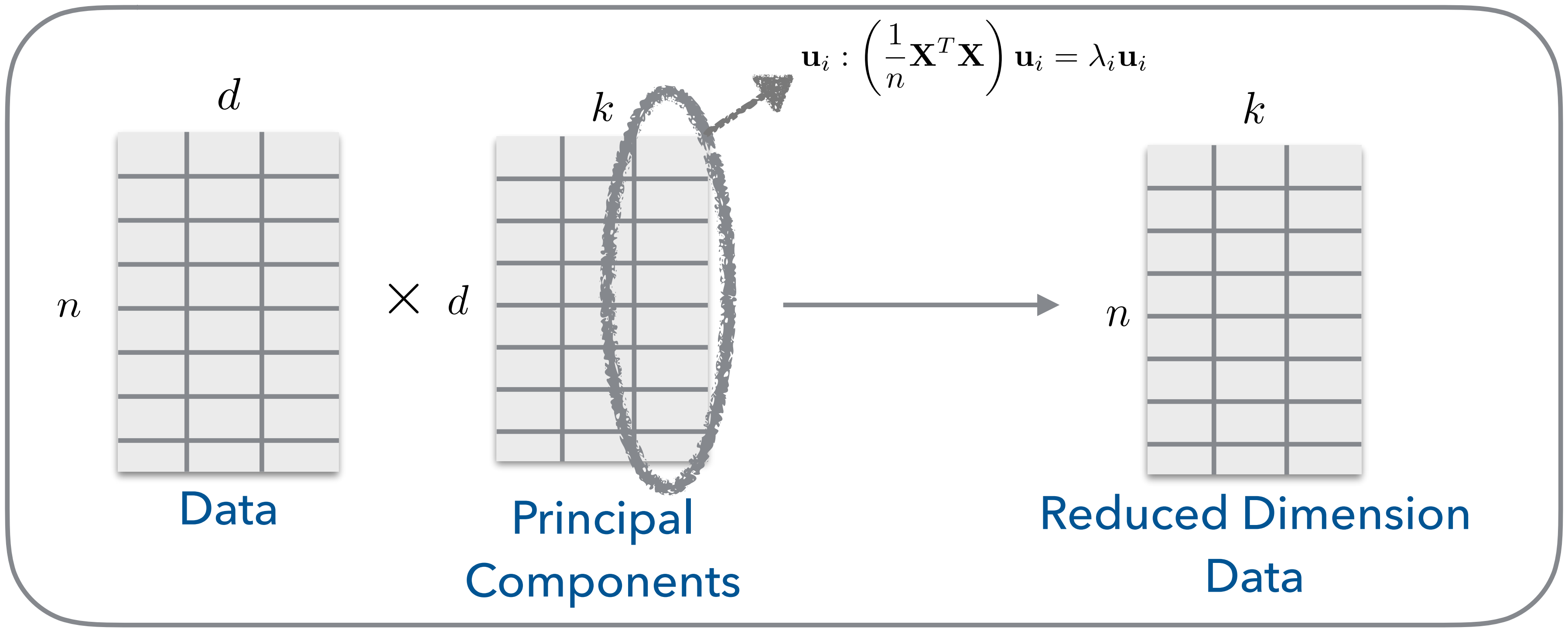
Dimensionality reduction

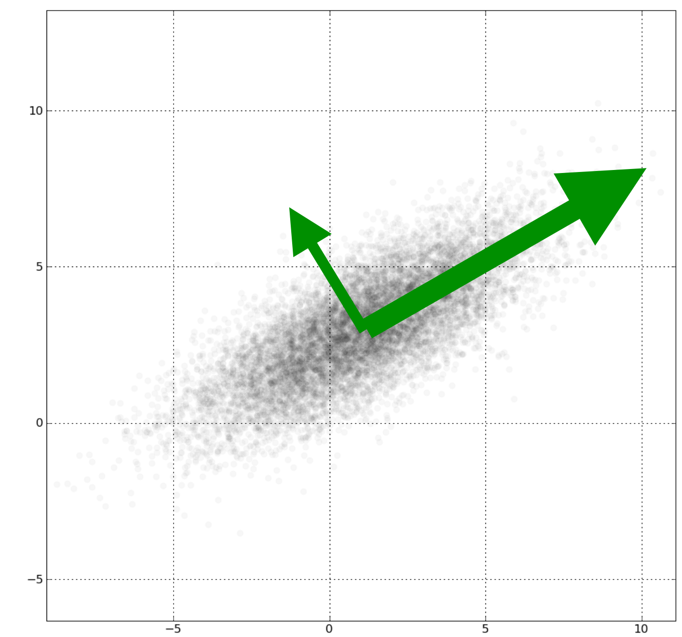
- Preprocessing step for high-dimensional data
- Novel use as a defense against evasion attacks

Various Algorithms tried…

- Principal Component Analysis (PCA)
- Random Projections
- Kernel PCA

Principal Component Analysis





Principal component

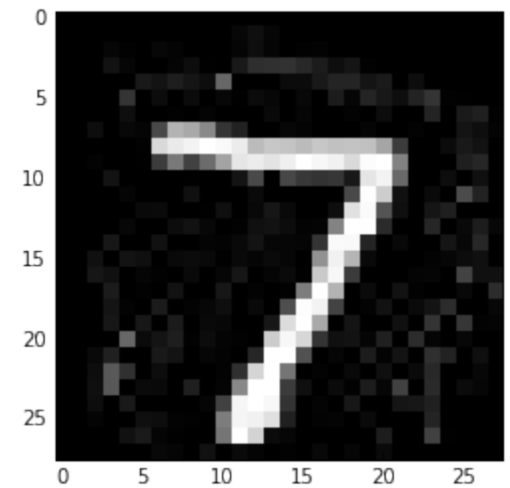
- Use Principal Component Analysis (PCA) to reduce dimension
- Identifies top *k* directions of highest variance
- Directions: eigenvectors of covariance matrix

Reconstruction-based defense

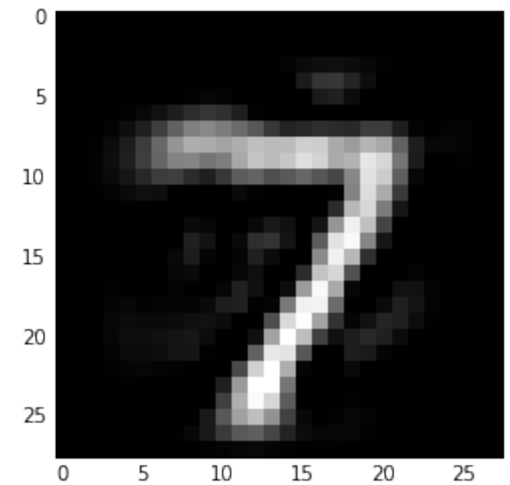
Step 1: Compute $\hat{\mathbf{x}} = \sum_{i=1}^{k} \langle \mathbf{x}, \mathbf{u}_i \rangle \mathbf{u}_i$, *reconstructed* input

(Input may be benign or adversarial)

Step 2: Find $f(\hat{\mathbf{x}})$, where $f(\cdot)$ is the original classifier



Initial adversarial example



After reconstruction

Intuition

- Perturbation added in existing attacks has low variance
- Reconstruction step removes perturbation

Re-training based defense

Step 1: Train new classifier f_k on $\mathbf{X}_k^{\text{train}}$ (red. dim. training data)

Step 2: Project all inputs to k dimensions

Step 3: Use f_k to classify subsequent inputs

Intuition

- For SVMs, margin increases for lower-dimensional classifiers

Results

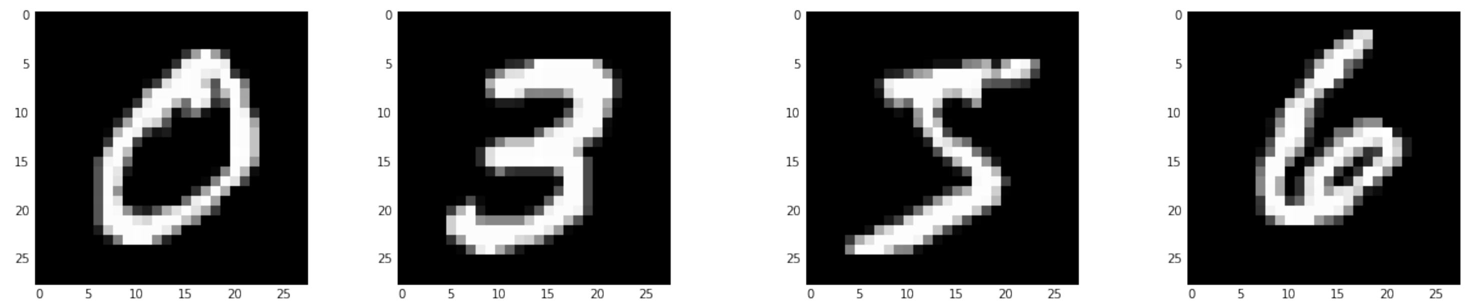
Validation of defenses

Do the defenses work for

1. different datasets?
2. various ML classifiers?
3. different attacks on the same classifier?
4. dimensionality reduction algorithms other than PCA?

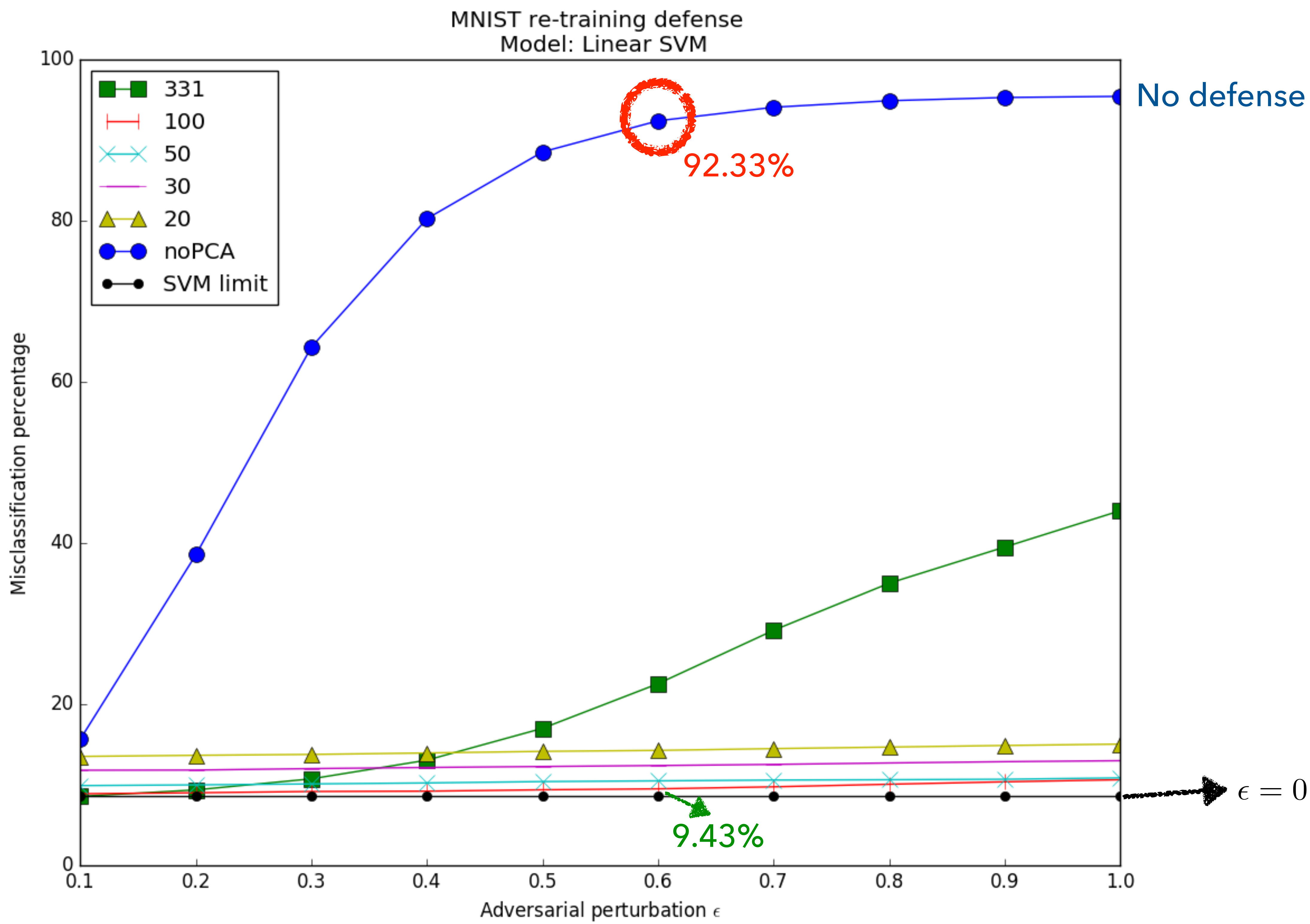
Datasets used

- MNIST: Handwritten digits from 0 to 9. Extensively studied from the attack perspective. Enables visual evaluation of defenses.

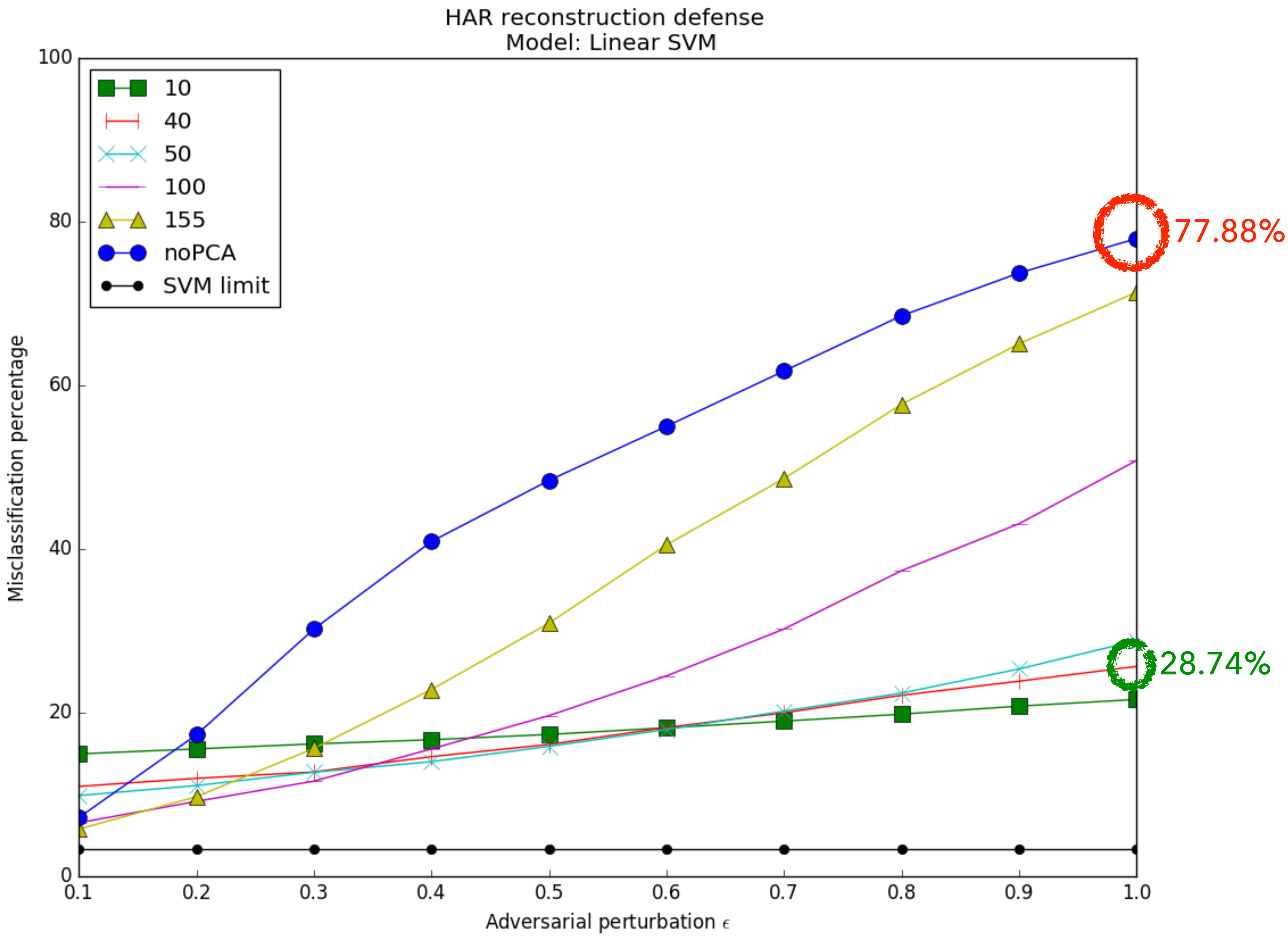


- UCI HAR: Measurements obtained from a smartphone's accelerometer and gyroscope. Six activities: Walking, Walking Upstairs, Walking Downstairs, Sitting, Standing and Laying.

Linear SVM: Re-training Defense for MNIST

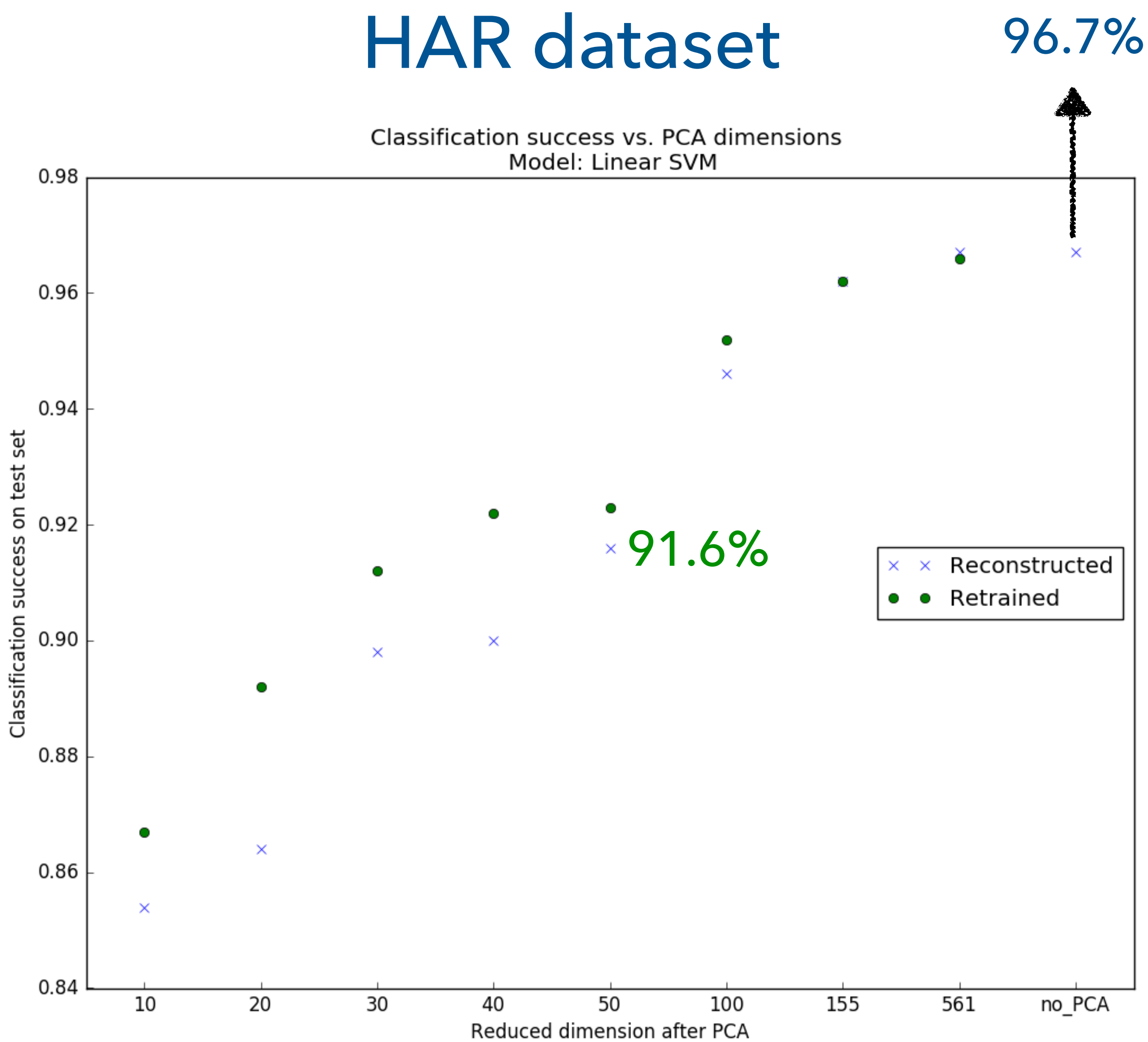


Linear SVM: Reconstruction Defense for HAR

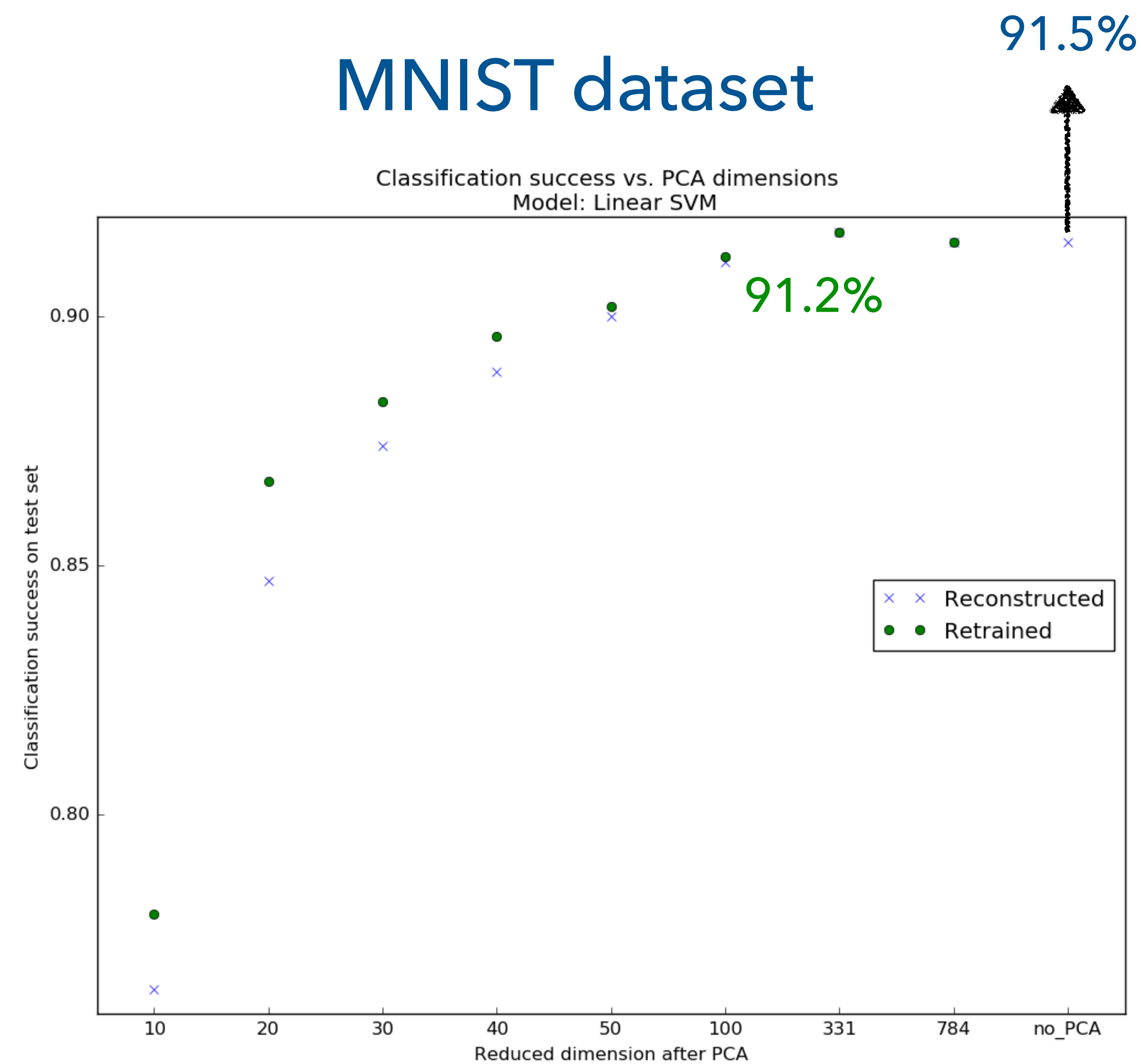


Classification accuracy

HAR dataset

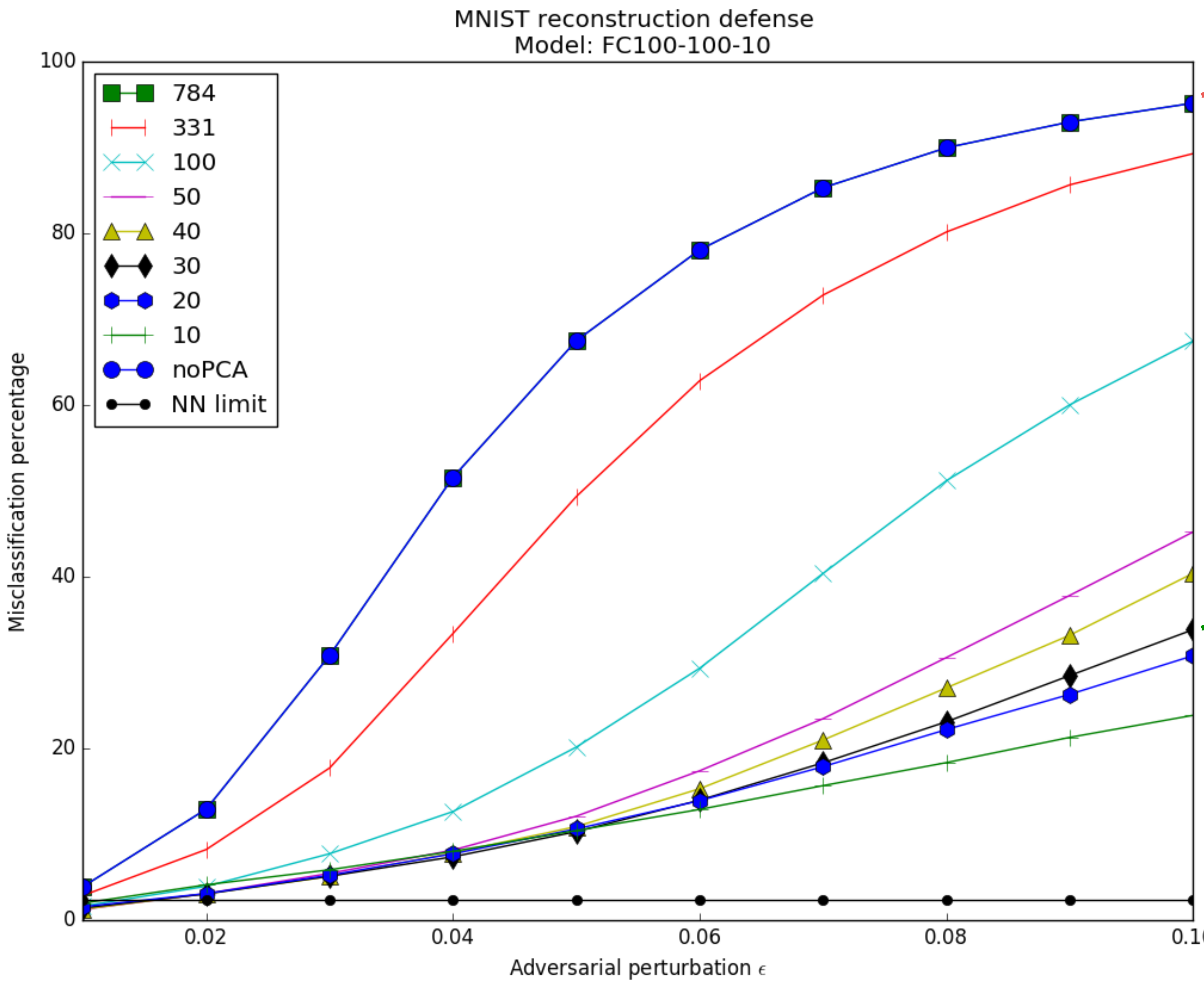


MNIST dataset



Takeaway: Defenses work for two different datasets with minimal utility loss

Neural Network: Reconstruction Defense for MNIST



96.86%,
Utility: 97.71%

33.77%,
Utility: 96.43%

Re-training gives 7.17% misclassification at utility of 97.19%!

Ongoing Work and Extensions

Strategic attacks

- What if the adversary is aware of the defenses?
- For PCA defense, heuristically, adversary adds perturbation in directions with large projection along principal components

- Ongoing evaluations suggest defenses are effective even for strategic adversary

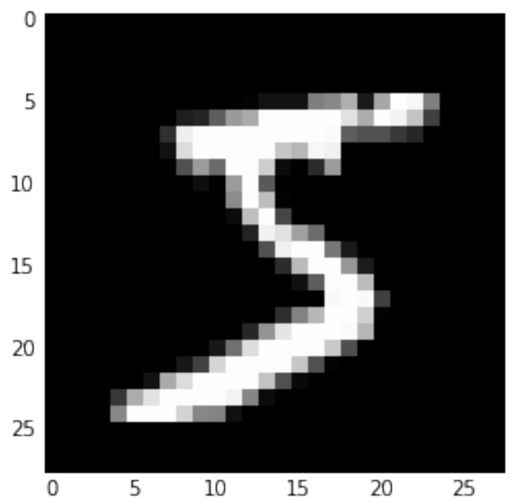
Extensions

- Formal definitions of classifier security
- Proofs for the effectiveness of dimensionality reduction
- Optimal attacks against various defenses and classifiers

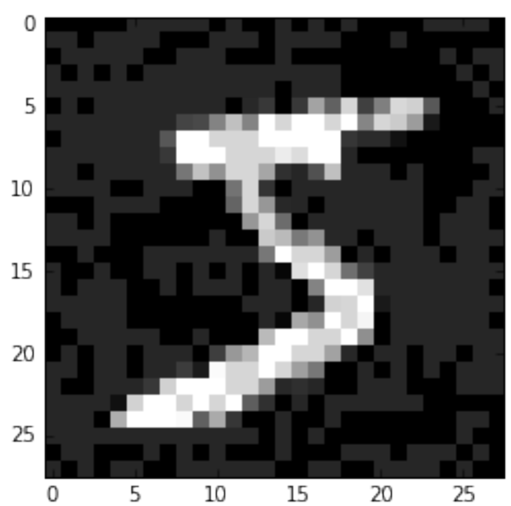
That's all folks!
Questions?

Backup slides

Evasion Attack on Neural Networks



Classified as 5



Classified as 0!

Adversarial image with ϵ=0.15

Leads to 99% misclassification on test set.

Fast Sign Gradient attack

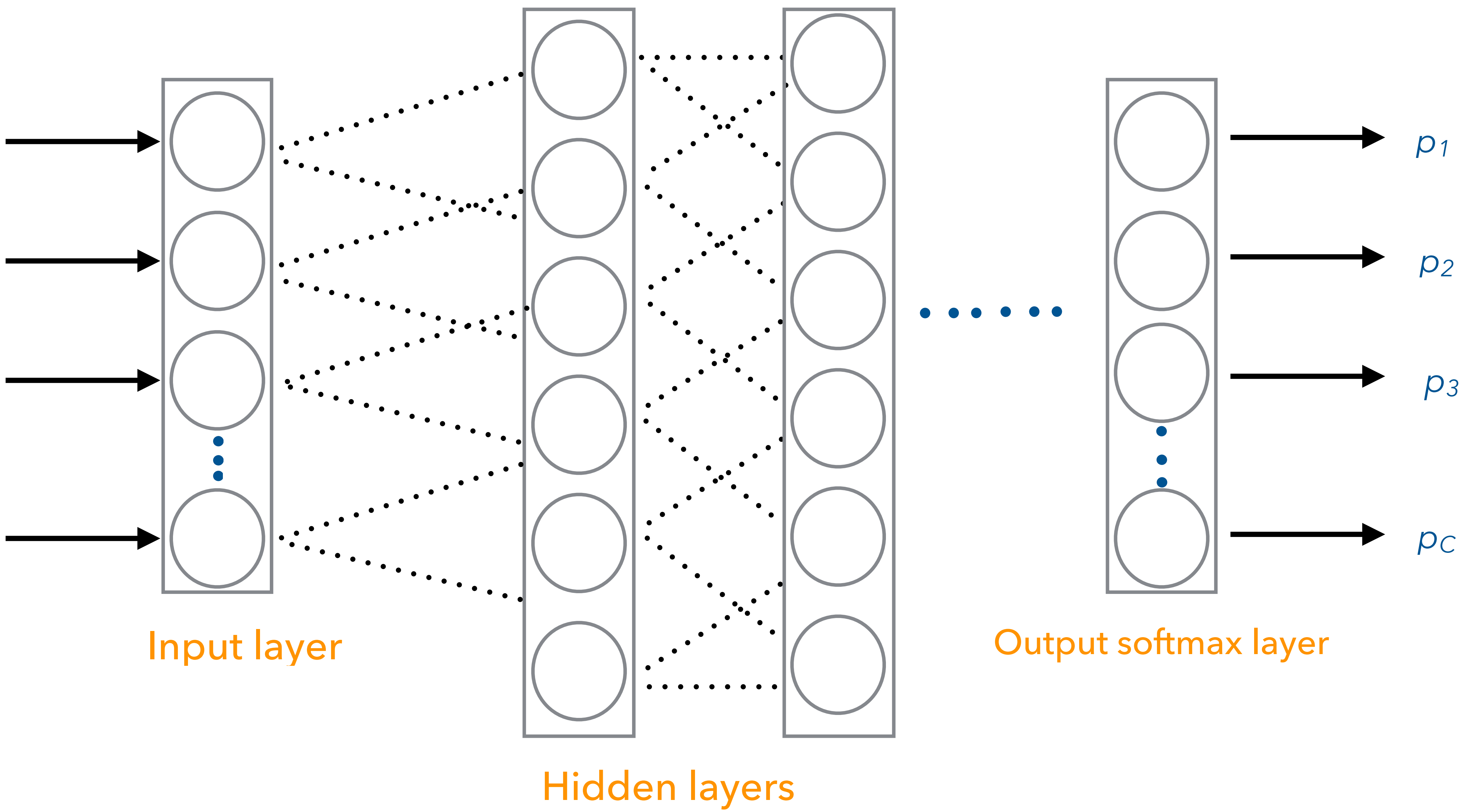
$$\mathbf{x}_{adv} = \mathbf{x} + \epsilon \ \mathrm{sign}(\nabla J_f(\mathbf{x}, y, \theta))$$

$$\epsilon \in [0, 1]$$

where $J_f(\cdot)$ is the loss function

of the neural network

Neural Networks



Function that takes an input **x** and outputs a vector of probabilities **y**, giving the probability of each class

Motivation

Machine Learning systems are ubiquitous

BUT

Vulnerable to adversarially modified inputs

SO

'Good' defenses are needed

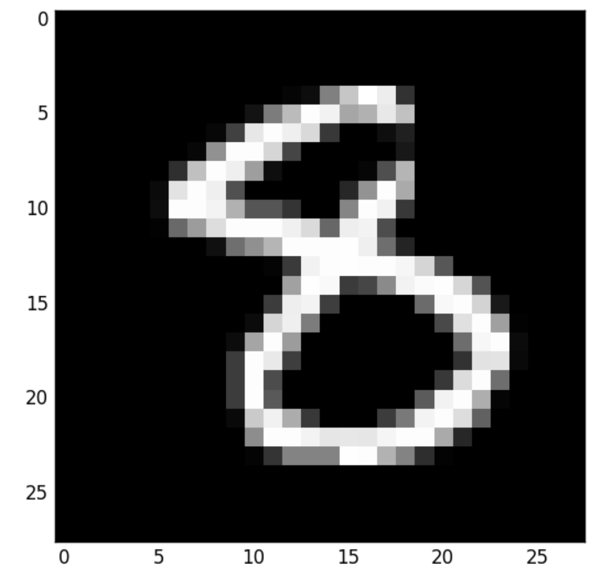
Dimensionality reduction as a defense against evasion attacks on machine learning classifiers

$$\min_{\mathbf{r}} \|\mathbf{r}\|_2$$
$$\text{subject to } f(\mathbf{x}+\mathbf{r}) = l,$$
$$\mathbf{x}+\mathbf{r} \in [0,1]^d.$$

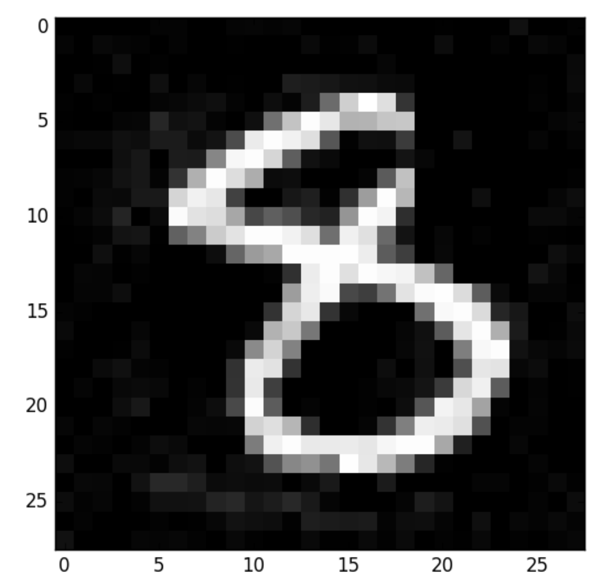
ere $\mathbf{x}$ is the input,

perturbation $\mathbf{r}$, and

e neural network f.

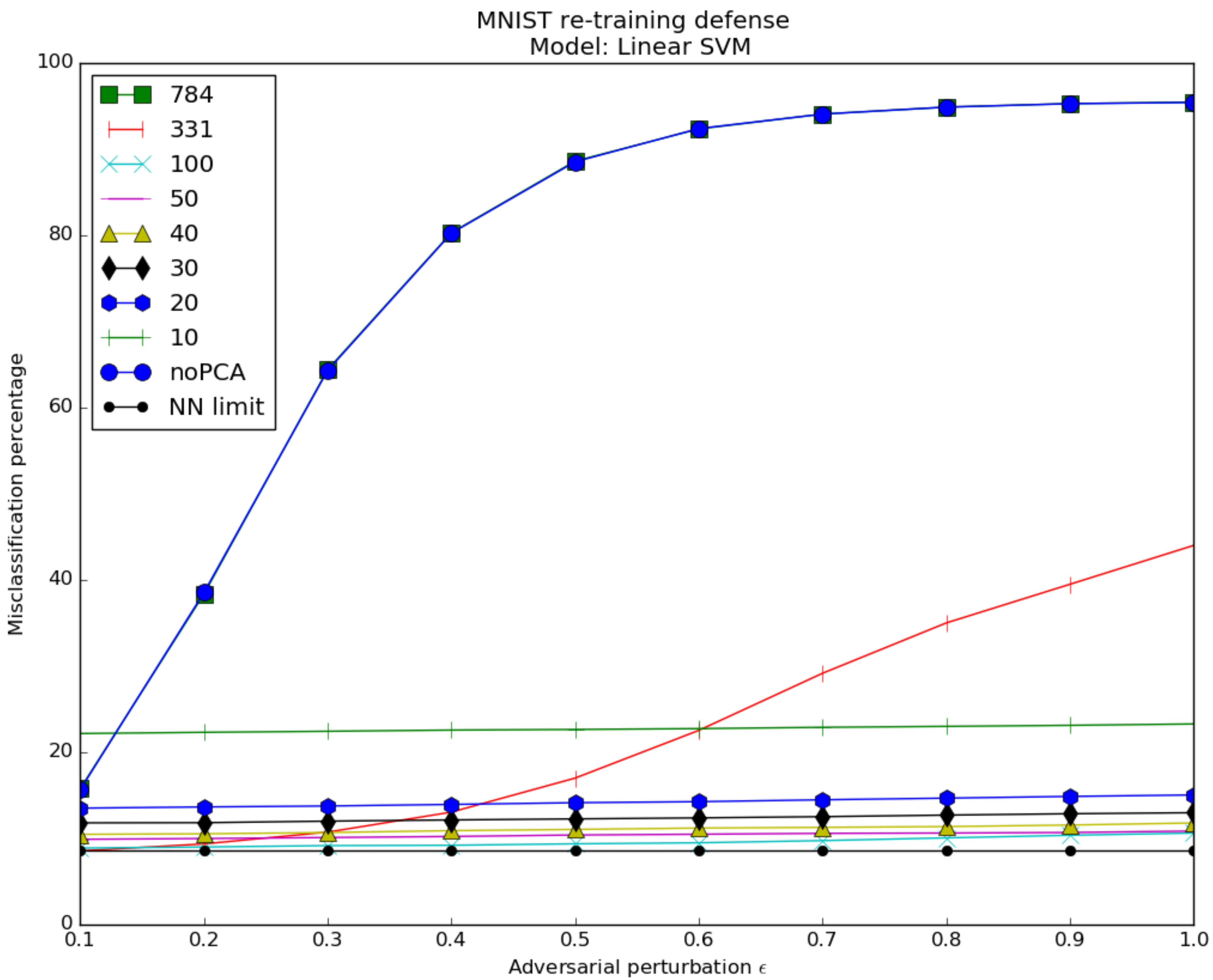


Classified as 8

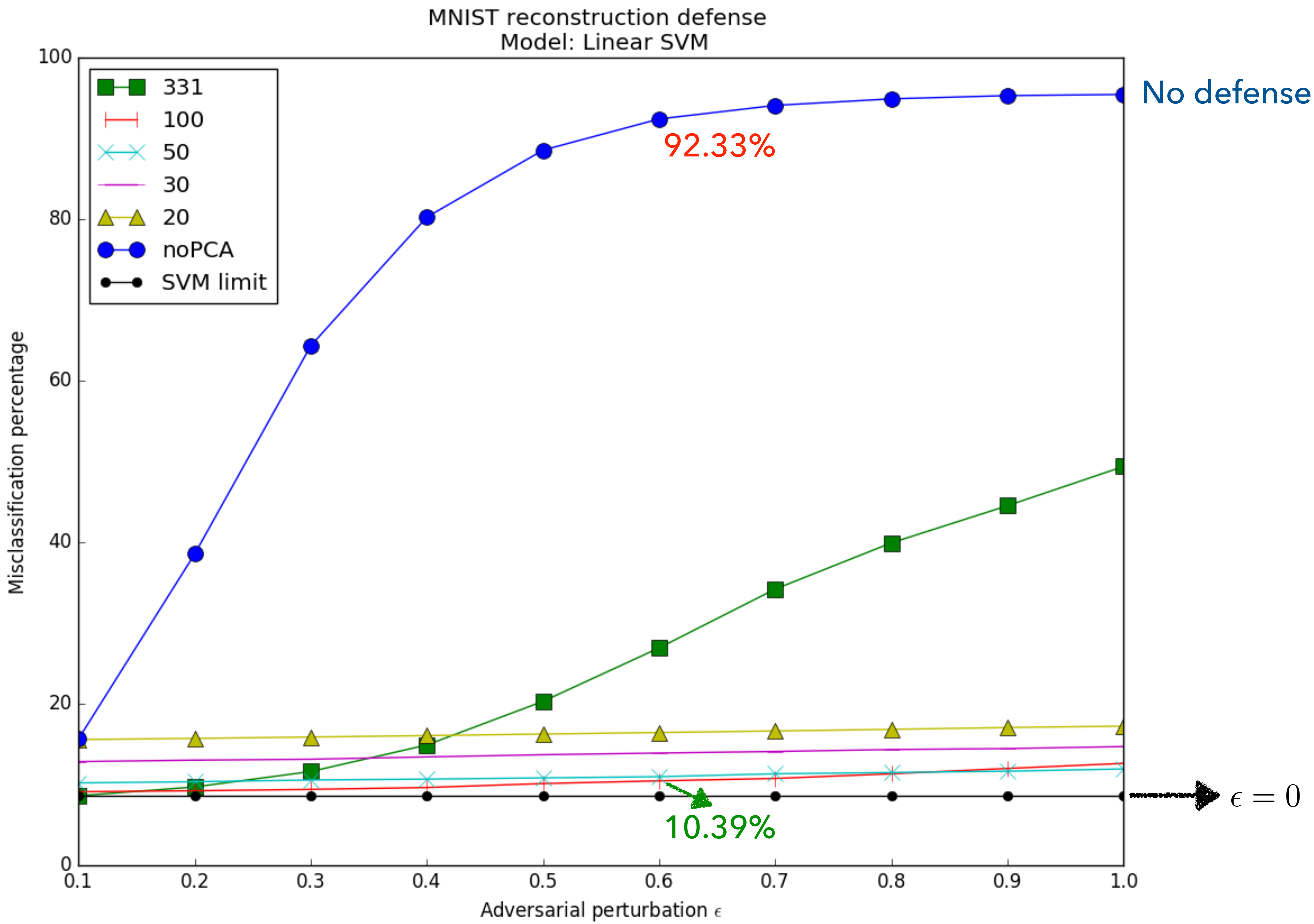


Classified as 3

Linear SVM: Re-training Defense for MNIST



Linear SVM: Reconstruction Defense for MNIST



Linear SVM: Re-training defense for HAR