Machine Learning in Security: Attacks & Defenses

Yamuna Prasad
IIT Jammu

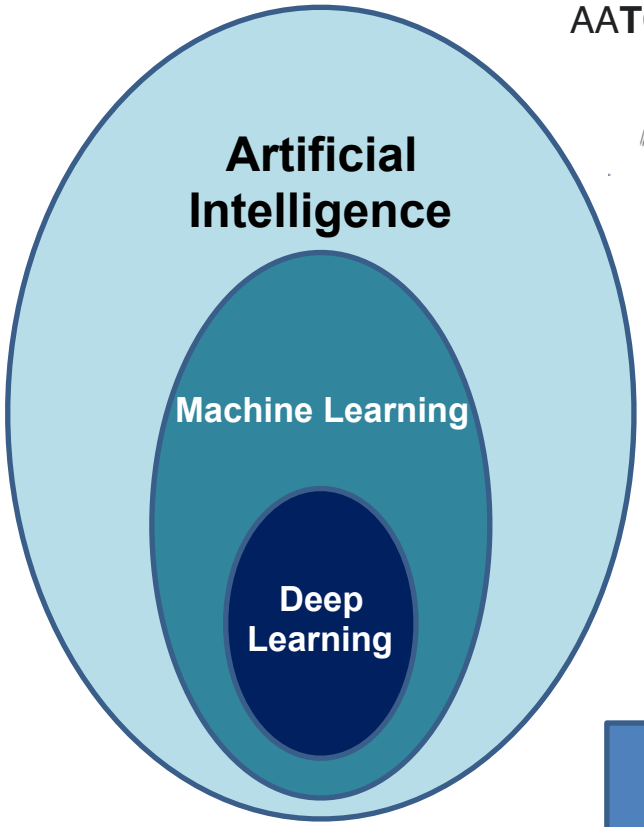
yamuna.prasad@iitjammu.ac.in

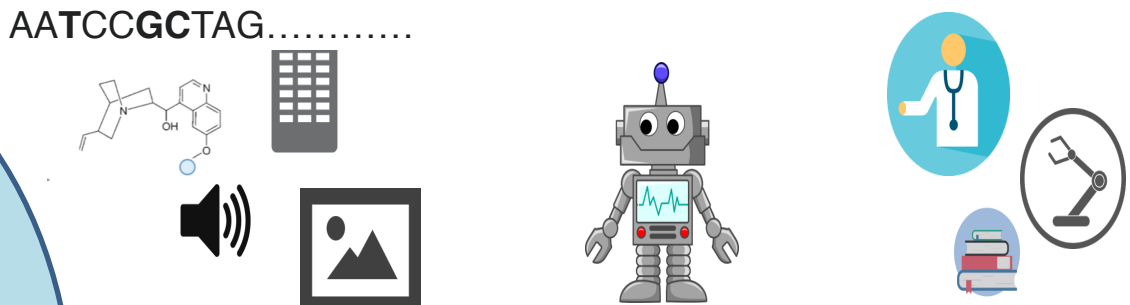
IIT JAMMU

Outline

- Introduction : Process Involved in ML (or AI Model)
- Sources of Data, Types and Challenges
- Problem Types
- Learning Process: Supervised & Unsupervised
- Approaches & Challenges in Learning Process
- ML in Security
- Attacks Spaces & Attack Types
- Attack Resistive Models
- Demo and Open Discussions

Introduction:Learning Process



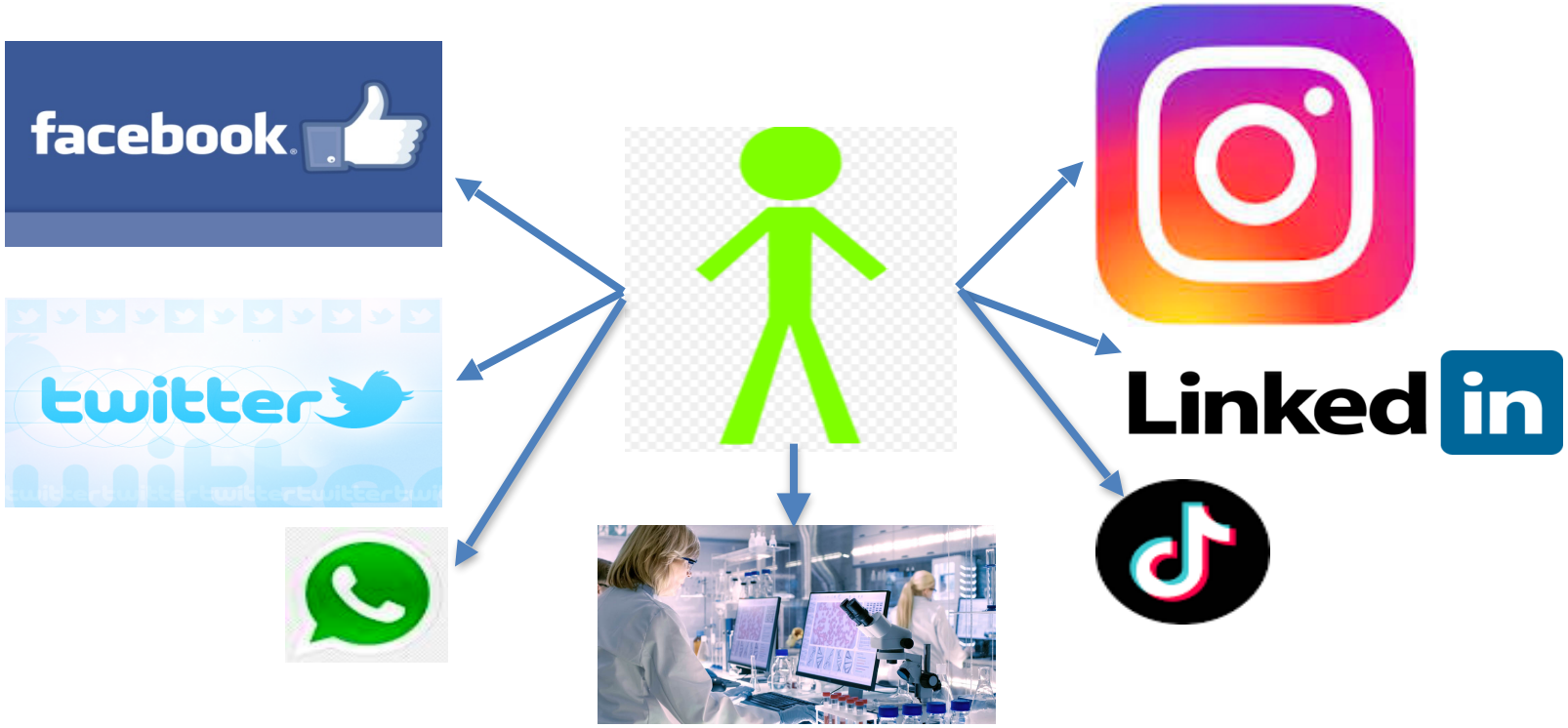


Img Ref: web

Introduction:Learning Process contd…

- Data Cleansing + Data Preparation + Data Analysis
- Data Cleansing: Noise Removal
- Data Preparation: *Missing Value Curation*, (Multi Dimensional) Vector Space Representation, Feature Identification, Feature Extraction, Feature Vector Normalisation, Data Augmentation without bias, Data Visualization
- Data Analysis: Statistical Inferencing, Machine Learning Models and Other Complex Predictive Models

Sources of Data

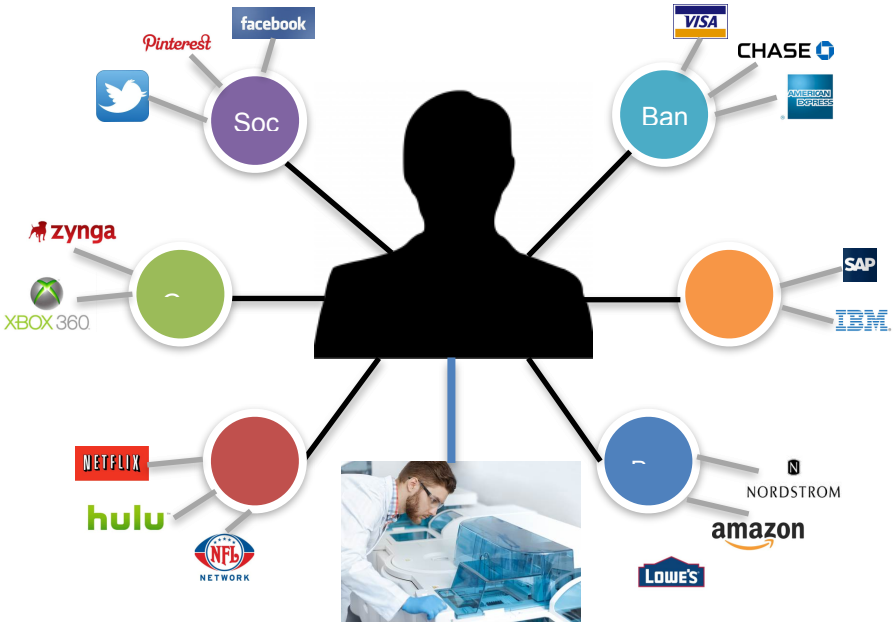


- User Location, Sentiment analysis, Personalisation, Product Recommendations, IoT and Security, Disease Diagnosis, Trend Prediction, Drug Discovery etc.
- Apps and Sensor

Img Ref: web sources & elsevier

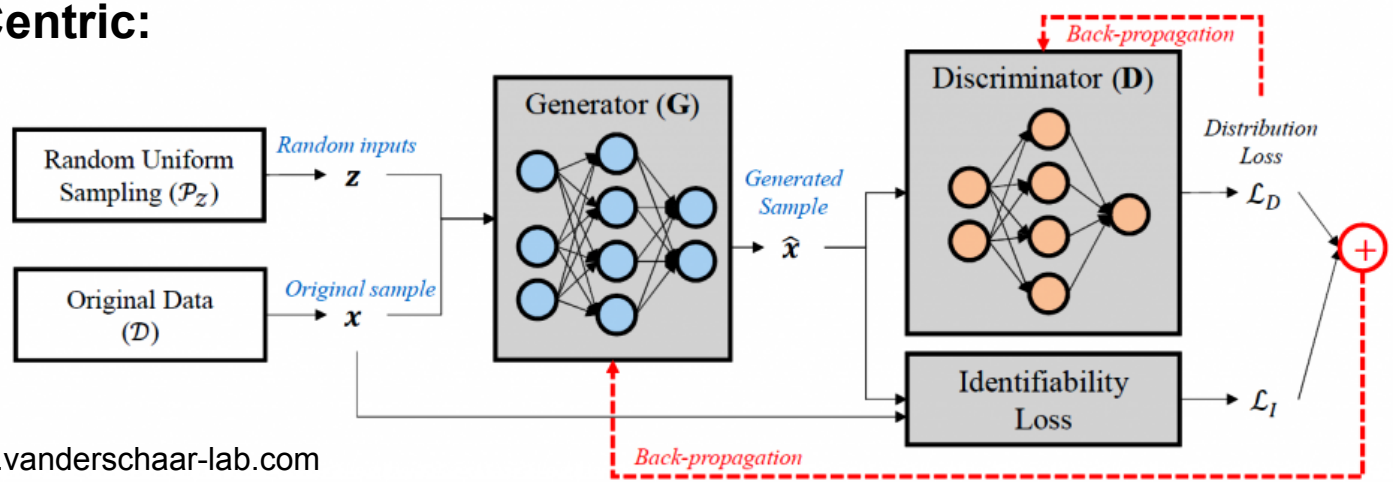
Data Generation: Detailed View

User Centric



Img Ref: web sources

Model Centric:

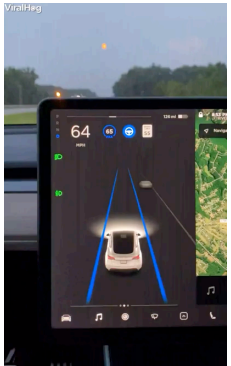


Img Ref: www.vanderschaar-lab.com

IIT JAMMU

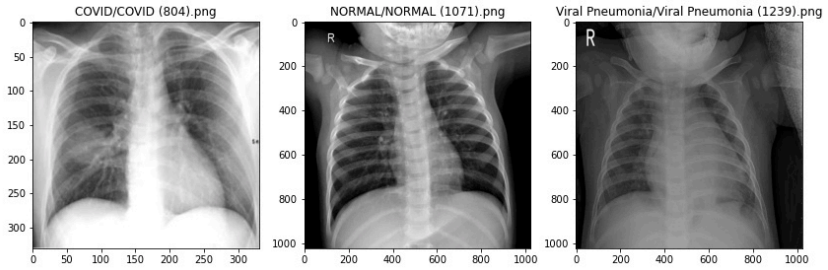
Challenges in Data

- Bioinformatics/Medical
  - Few samples, high dimensions/features etc.
  - Bias due to device noise, human error etc.
- Computer Vision:
  - Noises from sources, multimodality, human bias, IoTs
- Natural Language Processing
  - Noisy data, code-mixed language, discreteness, human bias

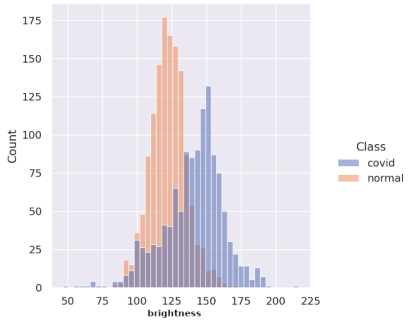


Challenges in Data: Bias Examples

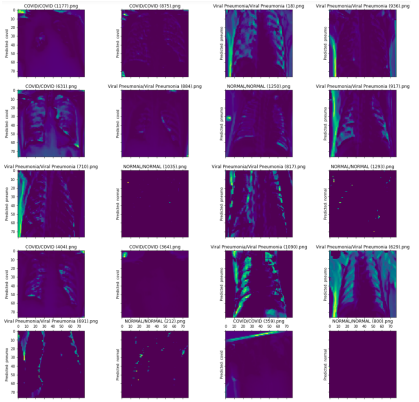
- COVID 19 Xray Images (COVID19 data at Kaggle)



Data



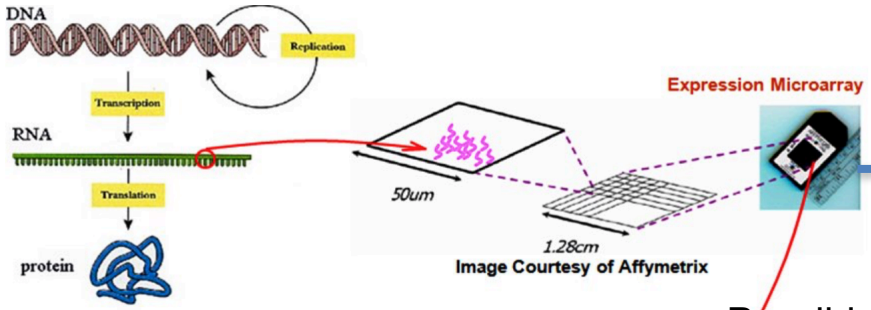
Histogram



Gradient Visualisation

IMg Ref: https://towardsdatascience.com

- Gene Expression Data



Genes Exp values

Sample \ Gene	M23197_at	U66497_at	M92287_at	...	Class
Sample 1	261	88	4778	...	ALL
Sample 2	101	74	2700	...	ALL
Sample 3	1450	34	498	...	AML
.	.	.	.	...	.
.	.	.	.	...	.
.	.	.	.	...	.

Expression Microarray Data Set

Possible Bias

Img Ref: web sources

IIT JAMMU

Types of Data

- Structured Data: Databases (sample and feature (attribute) value pairs
  - Most of the learning Models needs Structured Data for Modelling
- Unstructured Data: Text, Audio, Image and Video
- Semi-Structured: Partially Structured data
- Unimodal/Multimodal: Behavioural

GOAL for Data Analysis: (Un/Semi)-Structured => Structured

Types of Data: Examples

Unstructured data

The university has 5600 students.
John's ID is number 1, he is 18 years old and already holds a B.Sc. degree.
David's ID is number 2, he is 31 years old and holds a Ph.D. degree. Robert's ID is number 3, he is 51 years old and also holds the same degree as David, a Ph.D. degree.

AA**T**CC**GC**TAG……

Semi-structured data

```
<University>
 <Student ID="1">
  <Name>John</Name>
  <Age>18</Age>
  <Degree>B.Sc.</Degree>
 </Student>
 <Student ID="2">
  <Name>David</Name>
  <Age>31</Age>
  <Degree>Ph.D. </Degree>
 </Student>
....
</University>
```

Structured data

ID	Name	Age	Degree
1	John	18	B.Sc.
2	David	31	Ph.D.
3	Robert	51	Ph.D.
4	Rick	26	M.Sc.
5	Michael	19	B.Sc.

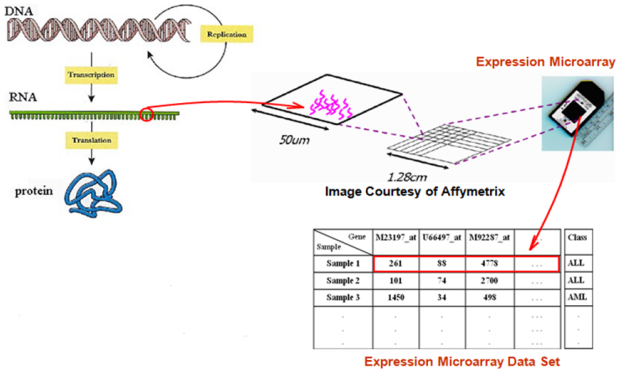
Unstructured Data ——> Feature Extraction ——> Vector Space Representation

Eg.

Structure (DNA/RNA/Protein) ===> Nucleotide Sequences (enumerate character rep)
Text Data ===> Bag of Words Representation/ One-hot Vector
Image Data ===> Pixel based/feature extraction
Speech/Audio/Signal ===> Feature Extraction

Problem Types

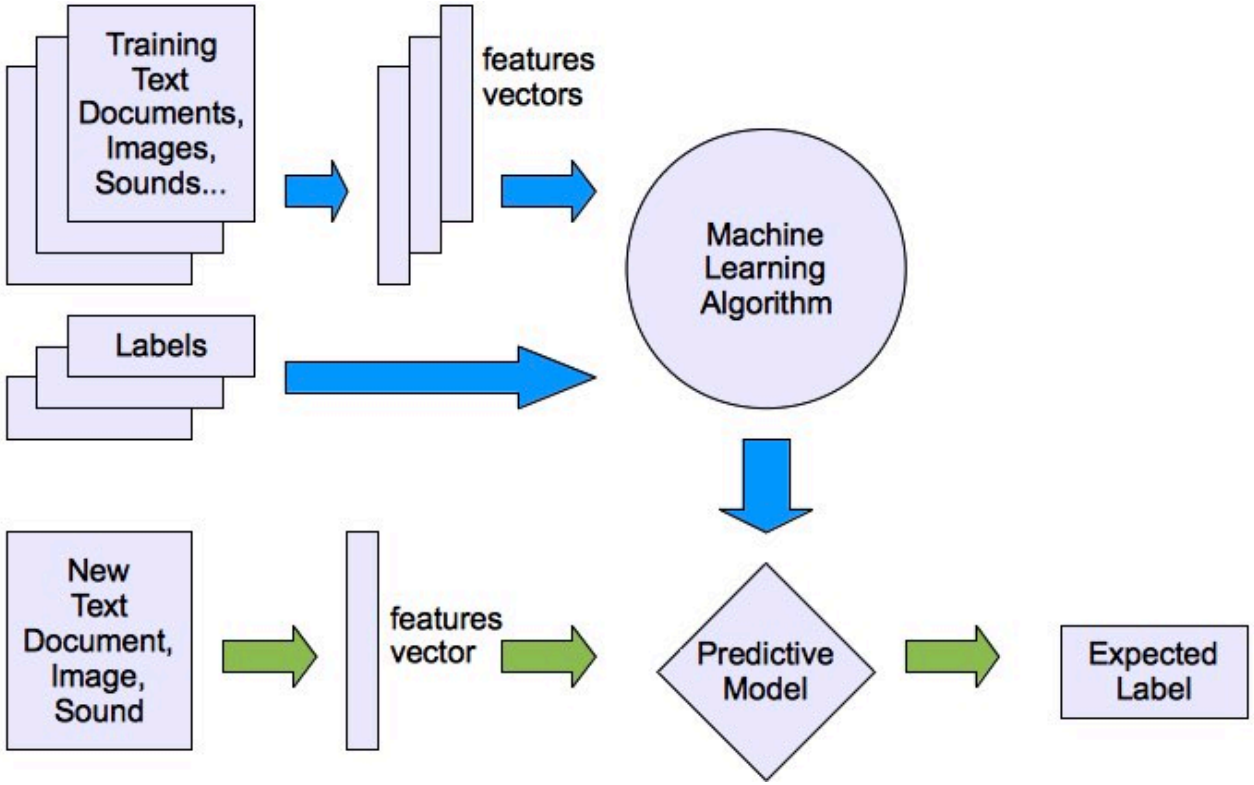
- Classification
  - Disease Diagnosis, Drug Target Identification, Drug Design and Discovery, Microbial Species Prediction,Environmental and Host Phenotypes Pred.,Interaction Pred
  - Intrusion Detection, Spam, Sentiment
  - Emotion, Product Reviews etc.
- Regression
  - Efficacy prediction
- Other problems in Various domains
  - Rankings
  - Translation
  - Summarisation etc.



Task: Classify novel samples into known disease type (disease diagnosis)
Challenge: Thousands of genes (columns), few samples
Solution: Dimensionality Reduction

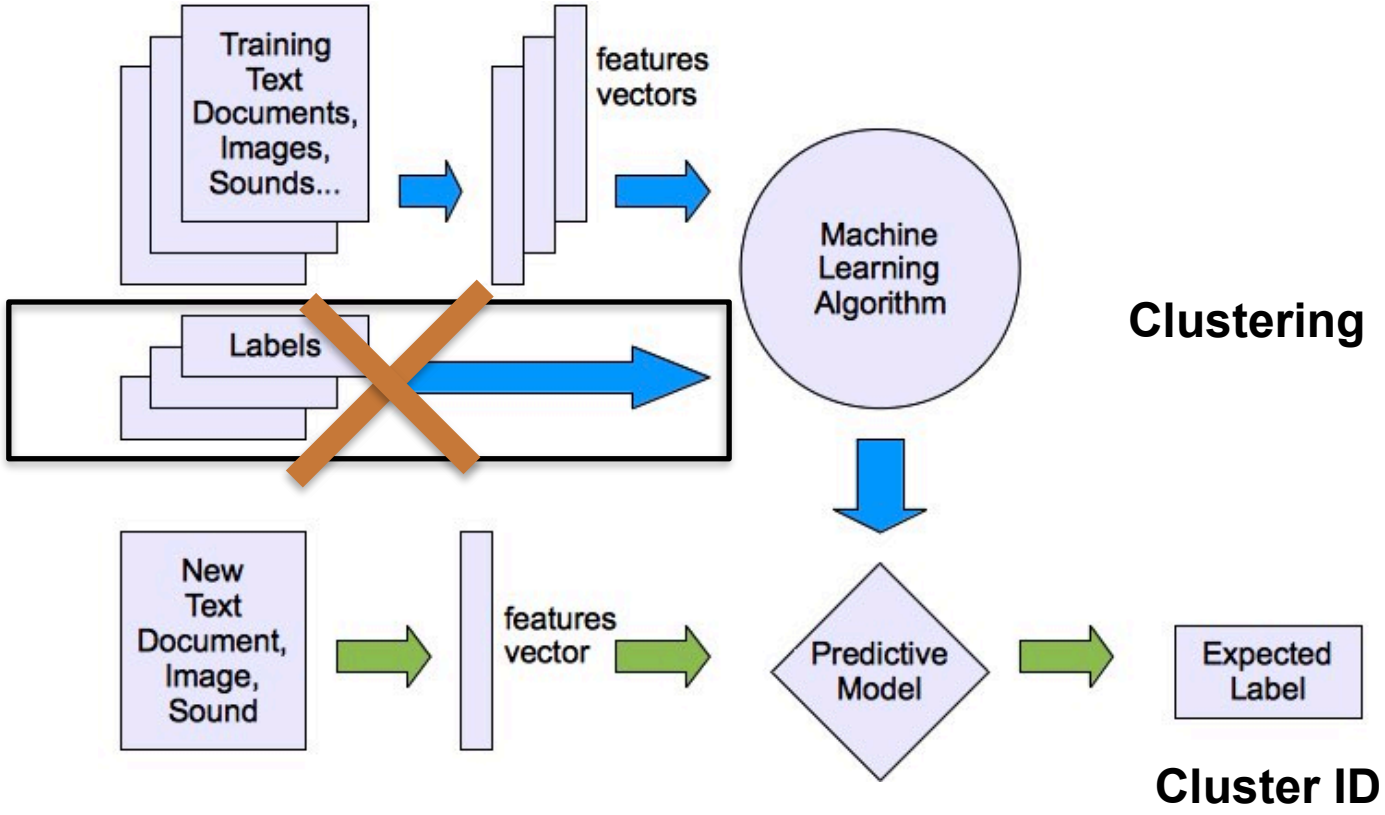
IIT JAMMU

Supervised Learning



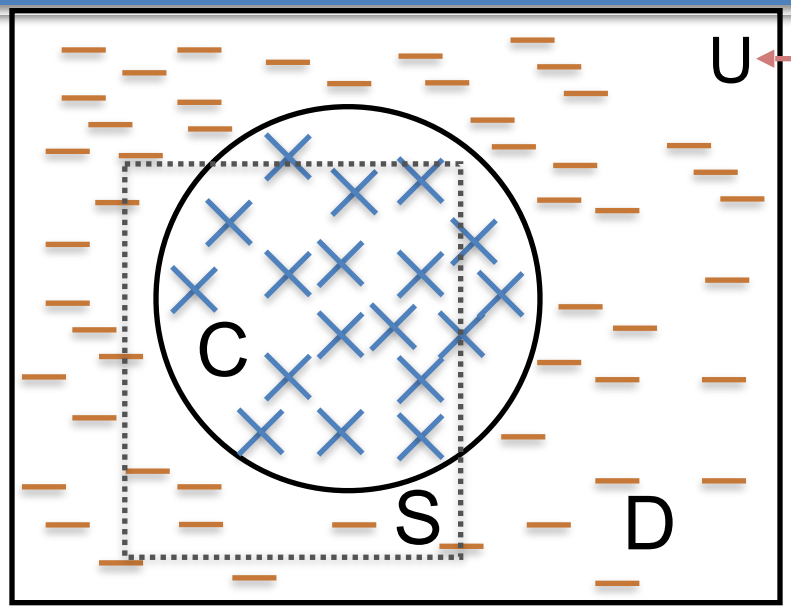
Img Ref: web

Un-Supervised Learning



Img Ref: web

Learning Theory



Universe data distribution

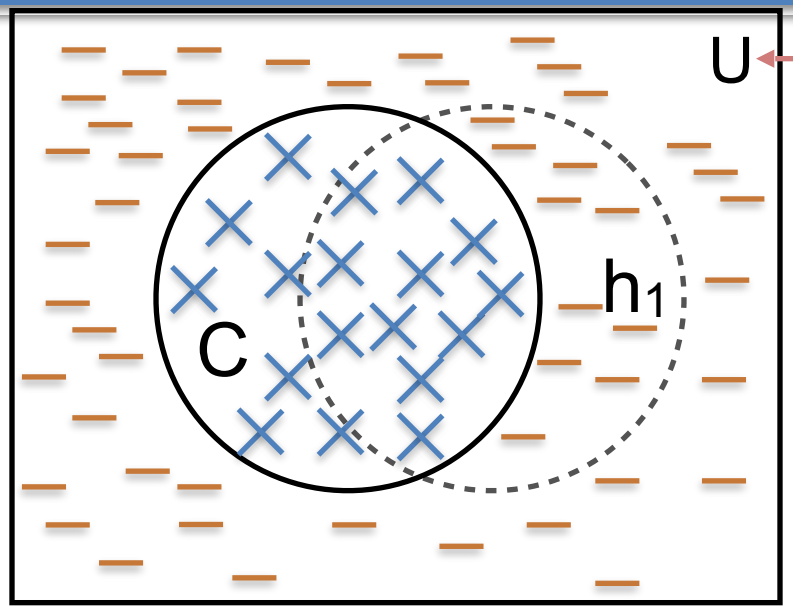
Let C be the concept (true function) which is capable of modelling the unknown data distribution **D**
And, the training sample **S** (i.i.d from **D**) is available to model the concept **C.**
We look for a hypothesis $h \in H$ which can model C with low generalization error (at the most $0 \leq \epsilon \leq 1$).
[Acceptable Hypothesis!]
Confidence?

Approximately correct:

$$P(C \oplus h) <= \epsilon$$

Prob. distribution Error

Learning Theory: Approximately correct hypothesis



Universe data distribution

$C \oplus h_1$ = Error region$_1$

$h_i \in H \quad \forall i$

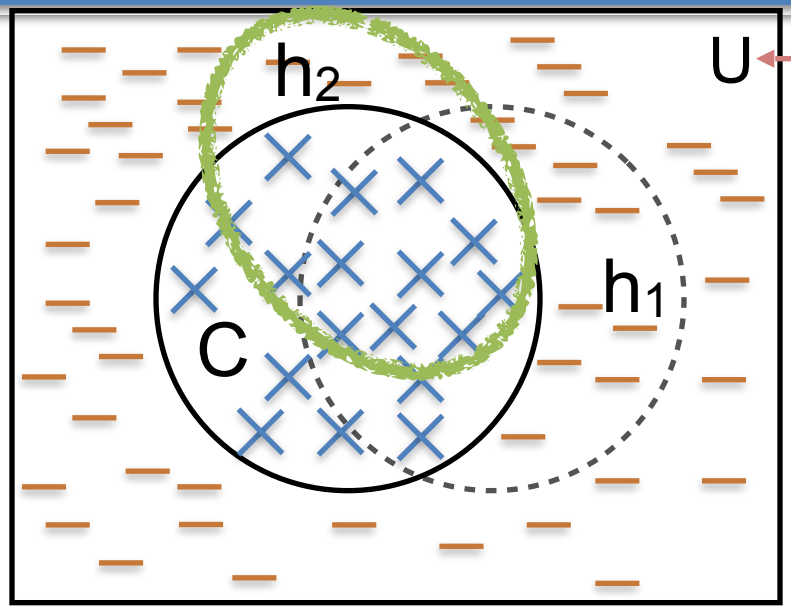
Approximately correct:

$$P(C \oplus h_1) <= \epsilon_1$$

Prob. distribution

Error

Learning Theory: PAC



Universe data distribution

$C \oplus h_1$ = Error region$_1$

$C \oplus h_2$ = Error region$_2$

$$h_i \in H \quad \forall i$$

Approximately correct:

$$P(C \oplus h_i) <= \epsilon_i$$

Prob. distribution

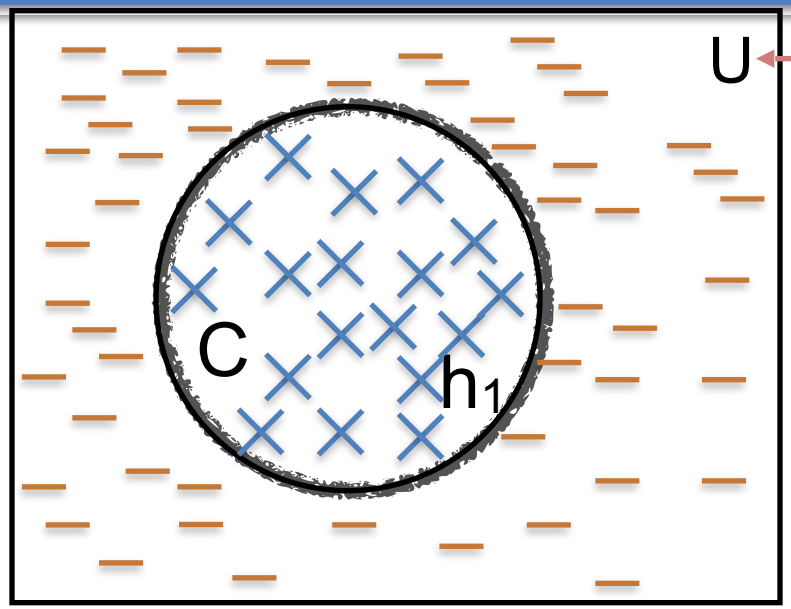
Error

Probably:

$$P(P(C \oplus h_1) > \epsilon_1)) < \delta$$

i.e. Probability that generalization error is less than ϵ_1 is at most $0 \leq \delta \leq 1$

Learning Theory: Consistent Hypothesis



Universe data distribution

$C \oplus h_1 = 0$

$h_i \in H \quad \forall i$

Approximately correct

$P(P(C \oplus h) <= \epsilon) >= (1 - \delta)$

Prob. distribution

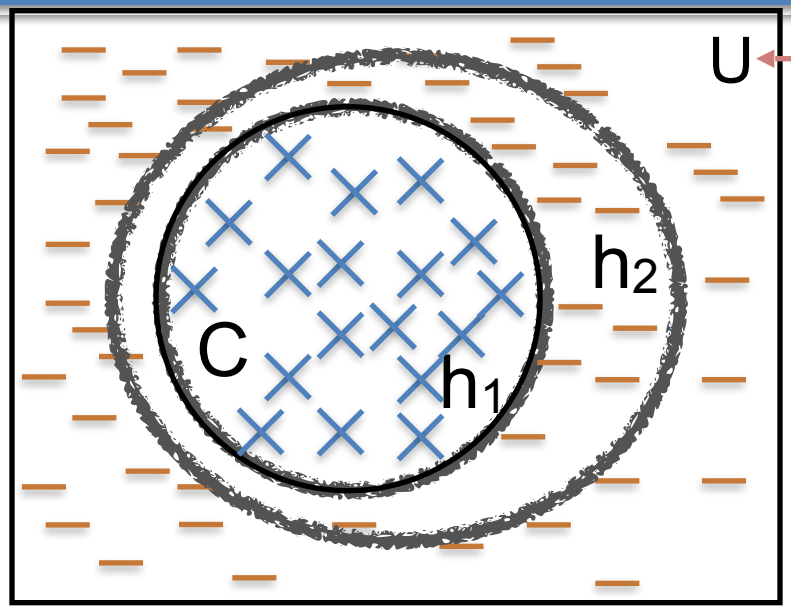
Confidence

OR

$P(P(C \oplus h) > \epsilon)) < \delta$

i.e. Probability that generalization error is less than ϵ is at most $0 \leq \delta \leq 1$

Learning Theory: Consistent Hypothesis



Universe data distribution

$C \oplus h_1 = 0$

$C \oplus h_2 = 0$

$h_i \in H \quad \forall i$

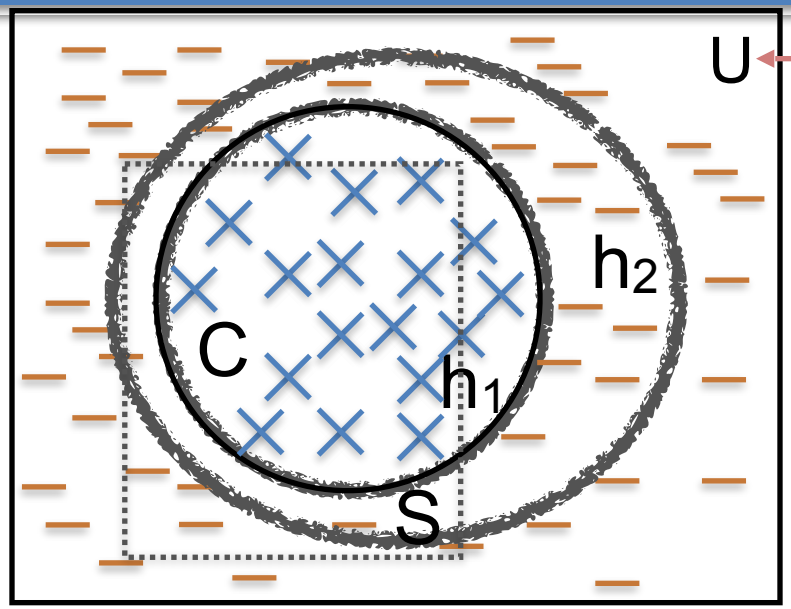
Approximately correct

$P(P(C \oplus h) <= \epsilon) >= (1 - \delta)$

Prob. distribution

Confidence

Learning Theory: Consistent Hypothesis



Universe data distribution

Consistent Hypothesis (i.e. $error_S(h) = 0$)

The gap between training and true errors:

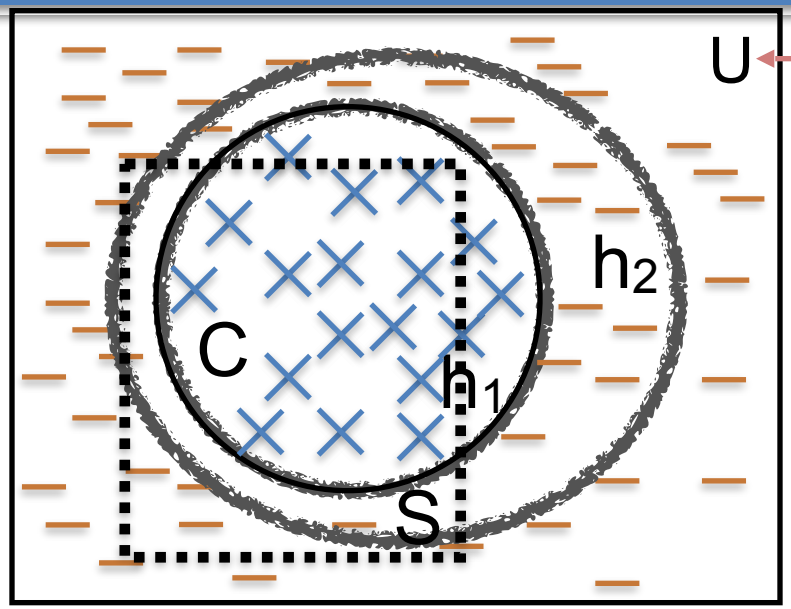
$$error_D(h) \leq error_S(h) + \epsilon$$

True error of a hypothesis h with respect to c ($error_D(h)$) is determined by how often h(x) and c(x) disagree (i.e. $h(x) \neq c(x)$) over future instances drawn at random (i.i.d) from D:

$$error_D(h) = P_{x \in D}[c(x) \neq h(x)]$$

$$error_D(h) = E_{x,c(x)}[L(c(x), h(x)] = \int_X \int_{c(X)} L(c(x), h(x)) P(x, c(x))\, dxdy$$

Learning Theory: Consistent Hypothesis



Universe data distribution

Consistent Hypothesis (i.e. $error_S(h) = 0$)

The gap between training and true errors:

$$error_D(h) \leq error_S(h) + \epsilon$$

Training error (empirical error) of a hypothesis h with respect to c ($error_S(h)$) is determined by how often h(x) and c(x) disagree (i.e. $h(x) \neq c(x)$) over training instances $x \in S$ ($\subseteq D$):

$$error_S(h) = P_{x \in S}[c(x) \neq h(x)] \equiv \frac{\sum_{x \in S} I(c(x) \neq h(x))}{|S|}$$

Learning Theory: Consistent Hypothesis (i.e. $error_S(h) = 0$)

The gap between training and true errors: $error_D(h) \leq error_S(h) + \epsilon$

$$P_r[(\exists h \in H)\ s.t.\ (error_S(h) = 0) \wedge (error_D(h) > \epsilon)] \leq |H| \exp^{-\epsilon m} \leq \delta$$

where, **m** is the number of samples in training data S (|S|)

Then:

$$m \geq \frac{1}{\epsilon}(\ln|H| + \ln(\frac{1}{\delta})$$

And, with probability at least (1- δ), the true error ($error_D(h)$) will be bounded as follows:

$$error_D(h) \leq \frac{1}{m}(\ln|H| + \ln(\frac{1}{\delta})$$

Learning Theory: Agnostic Learning (i.e. $error_S(h) \neq 0$)

The gap between training and true errors: $error_D(h) \leq error_S(h) + \epsilon$

$$P_r[(\exists h \in H)\ s.t.\ (error_D(h) > error_S(h) + \epsilon)] \leq |H| \exp^{-2\epsilon^2 m} \leq \delta$$

where, **m** is the number of samples in training data S (|S|)

Then:

$$m \geq \frac{1}{2\epsilon^2}(\ln|H| + \ln(\frac{1}{\delta})$$

And, with probability at least (1- δ), the true error ($error_D(h)$) will be bounded as follows:

$$error_D(h) \leq error_S(h) + \sqrt{\frac{1}{2m}(\ln|H| + \ln(\frac{1}{\delta}))}$$

Learning Theory: Infinite Hypothesis space (|H| = ∞)

Expressiveness of an infinite hypothesis space: ***Vapnik Chervonenkis DImension***

The gap between training and true errors: $error_D(h) \leq error_S(h) + \epsilon$

where, **m** is the number of samples in training data S (|S|)

Then:

$$m \geq \frac{1}{\epsilon}\left(8\ VC(H)\ \log_2\left(\frac{13}{\epsilon}\right) + 4\log_2\left(\frac{2}{\delta}\right)\right)$$

And, with probability at least (1- δ), the true error ($error_D(h)$) will be bounded as follows:

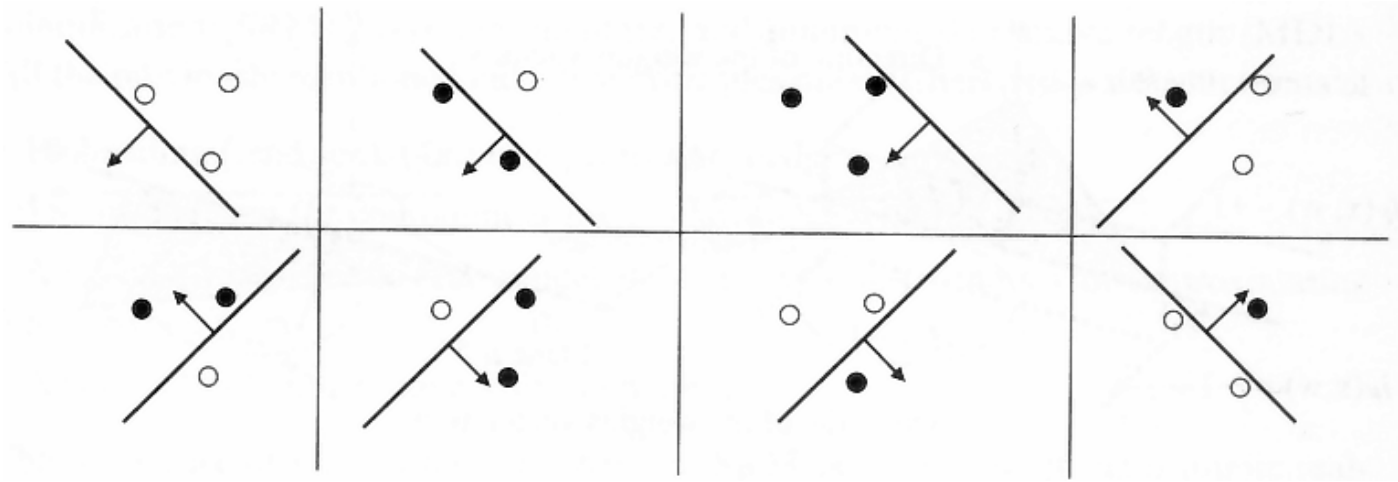
$$error_D(h) \leq error_S(h) + \sqrt{\frac{1}{m}\left(VC(H)\left(\ln\frac{2m}{VC(H)} + 1\right) + \ln(\frac{4}{\delta})\right)}$$

Learning Theory: Infinite Hypothesis space (|H| = ∞)

High VC dimension => better chance of approximating **h s.t.** **($error_S(h) = 0$)**

Low VC dimension => better chance of generalizing out of sample **($error_S(h) \approx error_D(h)$)**

The gap between training and true errors: $error_D(h) \leq error_S(h) + \Omega(VC(H))$



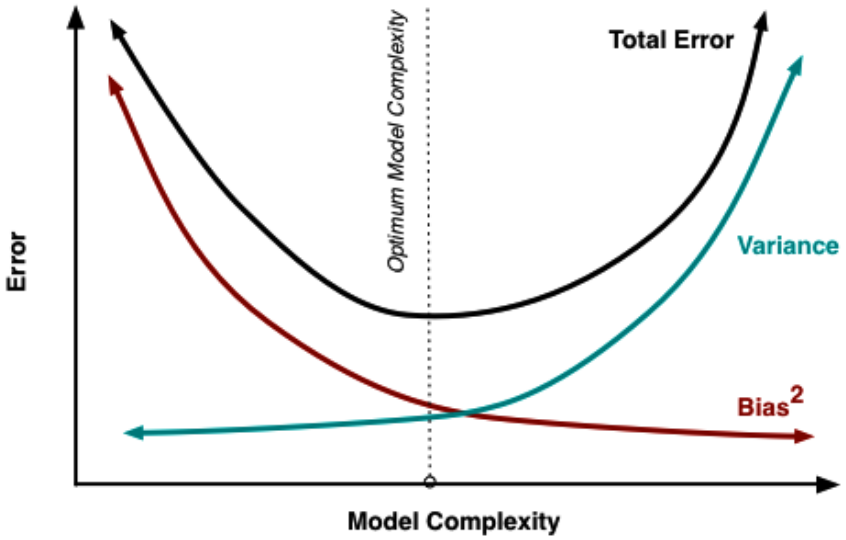
Learning Theory: Infinite Hypothesis space (|H| = ∞)

High VC dimension => better chance of approximating **h s.t. ($error_S(h) = 0$)**

Low VC dimension => better chance of generalizing out of sample **($error_S(h) \approx error_D(h)$)**

The gap between training and true errors:

$$error_D(h) \leq error_S(h) + \Omega(VC(H))$$



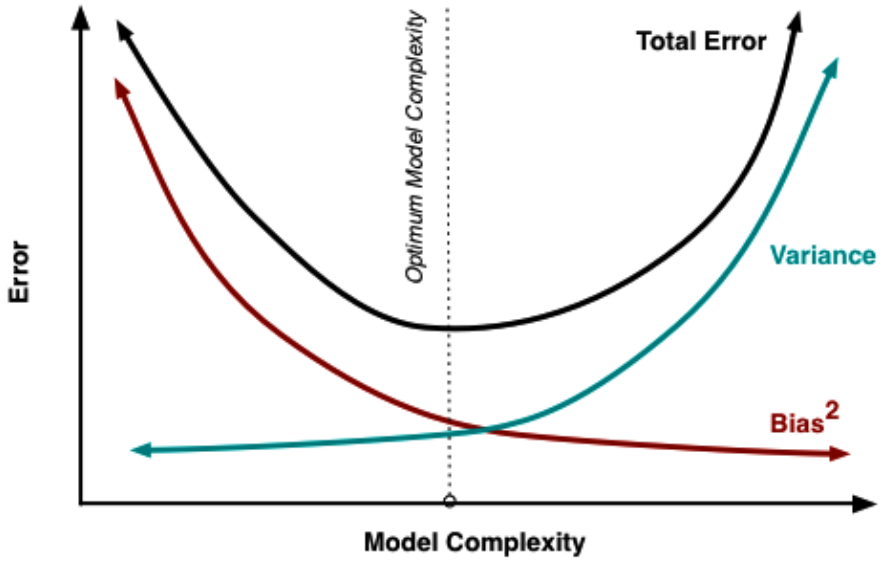
Img Ref: http://scott.fortmann-roe.com/docs/BiasVariance.html

Challenges in Learning

- Bias-Variance
  - Overfitting (Variance)
    - Regularization, Cross-validation, Data Augmentation
    - Reduce Model Complexity
  - Underfitting (Bias)
    - Increase the Model Complexity, Data curation, Domain Information

- Loss (Empirical and Structural)
- VC-dimensions

Explainable AI !



Img Ref: http://scott.fortmann-roe.com/docs/BiasVariance.html

A Toy Adversarial Example

Let h(x) be $P(y=1|<x;w,b>) = \sigma(w^T x+b)$, where $\sigma(z)=1/(1+e^{-z})$

This simple logistic regression (or perceptron model) decides that the **class of the input x is 1 if h(x) > 0.5 and 0 otherwise**

x = [2, -1, 3, -2, 2, 2, 1, -4, 5, 1] // input
w = [-1, -1, 1, -1, 1, -1, 1, 1, -1, 1] // weight vector
b = 0

Before adversarial:

Probability of class 1 is computed as $1/(1+e^{(-(-3))})$ = **0.0474**

// xad = x + 0.5w produces:
xad = [1.5, -1.5, 3.5, -2.5, 2.5, 1.5, 1.5, -3.5, 4.5, 1.5]

After Adversarial:

Probability of class 1 becomes $1/(1+e^{(-2)})$ = **0.88**

A Toy Adversarial Example Contd…

Let h(x) be $P(y=1|<x;w,b>) = \sigma(w^T x+b)$, where $\sigma(z)=1/(1+e^{-z})$

This simple logistic regression (or perceptron model) decides that the **class of the input x is 1 if h(x) > 0.5 and 0 otherwise**

x = [2, -1, 3, -2, 2, 2, 1, -4, 5, 1] // input
w = [-1, -1, 1, -1, 1, -1, 1, 1, -1, 1] // weight vector
b = 0

Before adversarial:

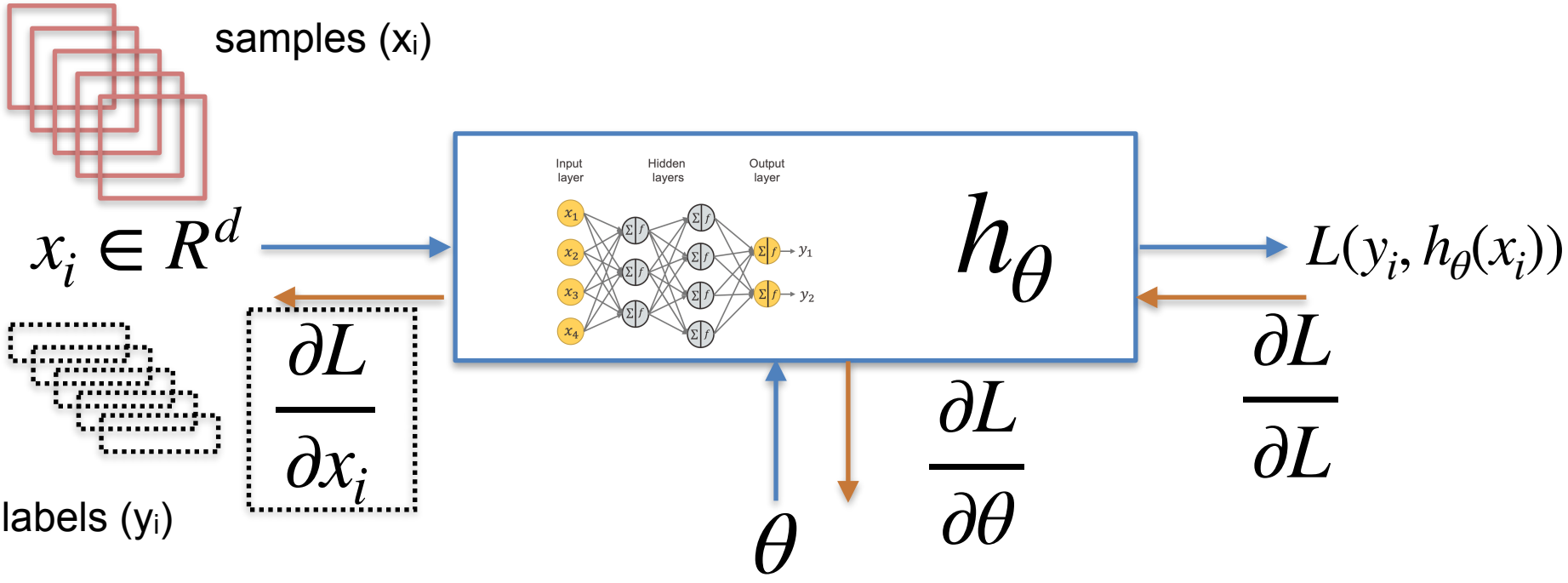
Probability of class 1 is computed as $1/(1+e^{(-(-3))})$ = **0.0474**

// xad = x + 0.5w produces:
xad = [1.5, -1.5, 3.5, -2.5, 2.5, 1.5, 1.5, -3.5, 4.5, 1.5]

After Adversarial:

Probability of class 1 becomes $1/(1+e^{(-2)})$ = **0.88**

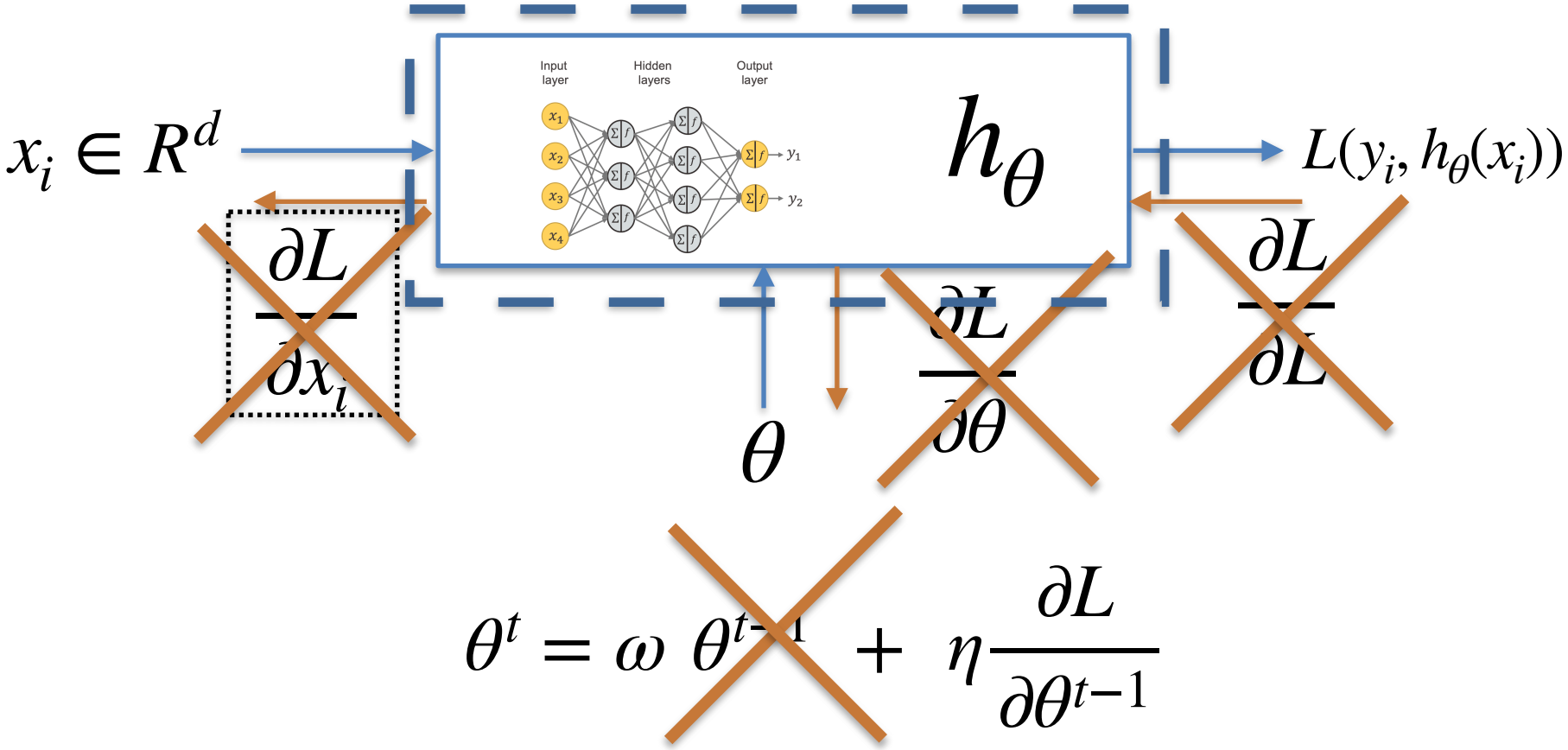
Attack Spaces: White-Box



$$\theta^t = \omega\ \theta^{t-1} + \eta \frac{\partial L}{\partial \theta^{t-1}}$$

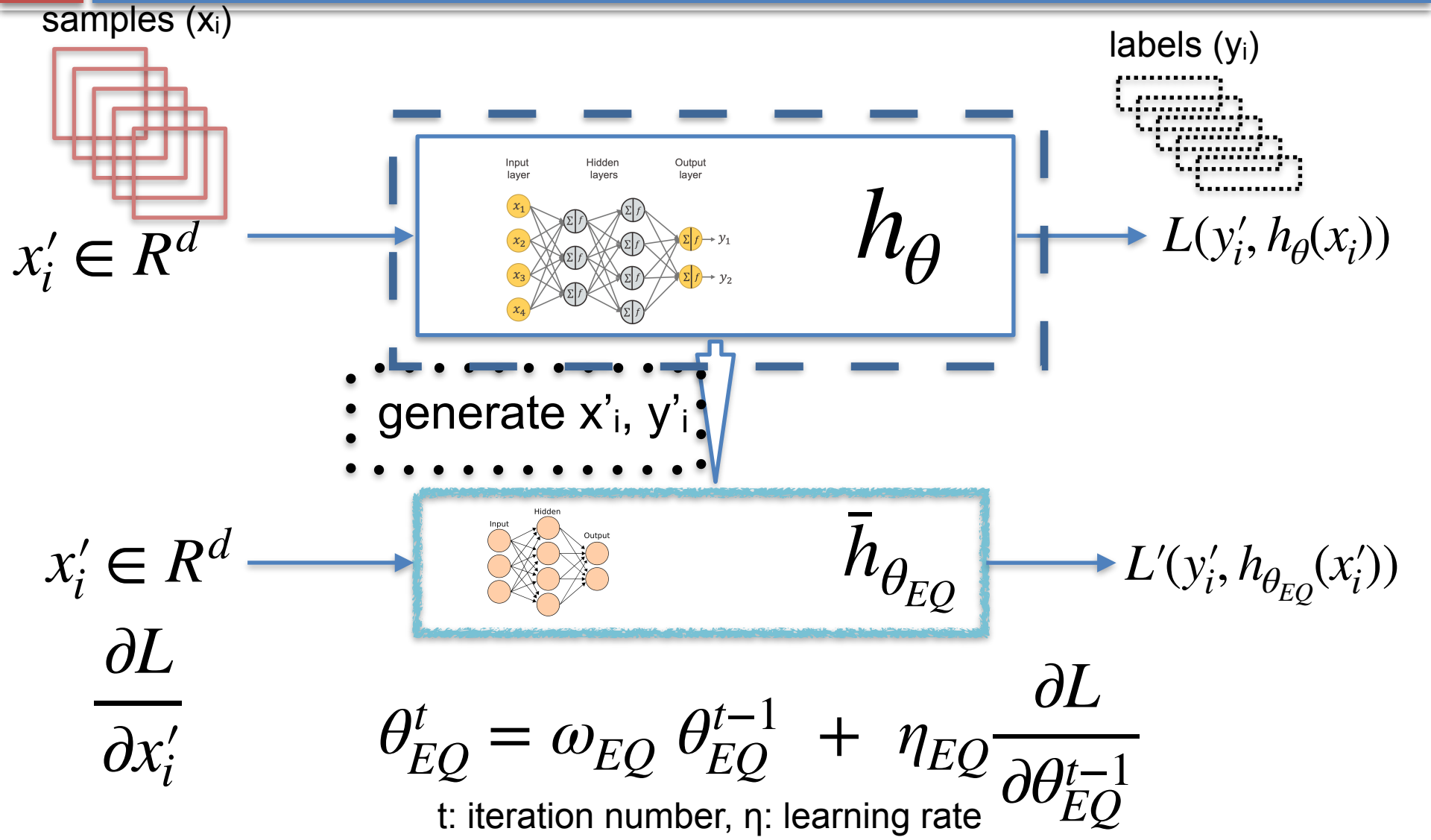
t: iteration number, η: learning rate

Attack Spaces: Black-Box

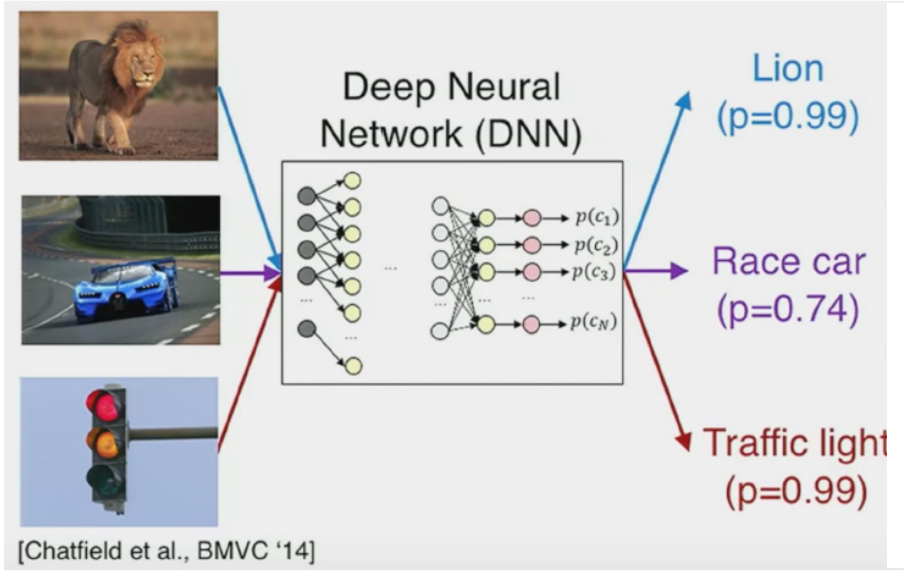


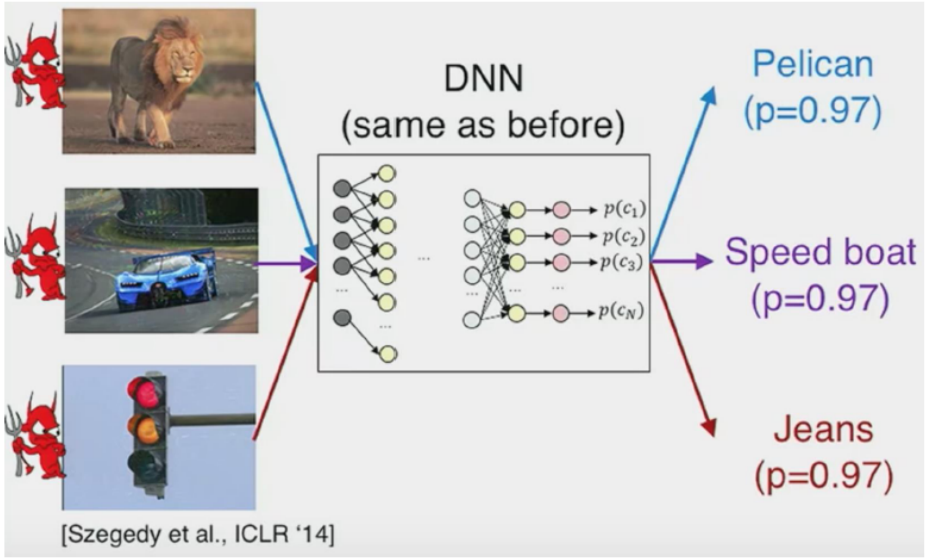
t: iteration number, η: learning rate

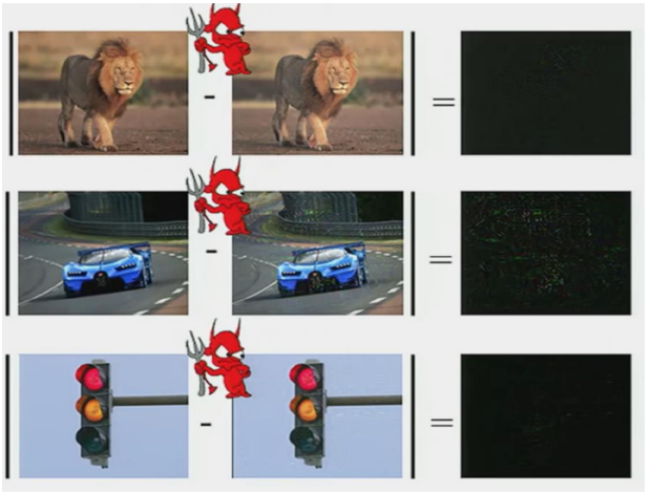
Attack Spaces: Black-Box



Adversarial Examples

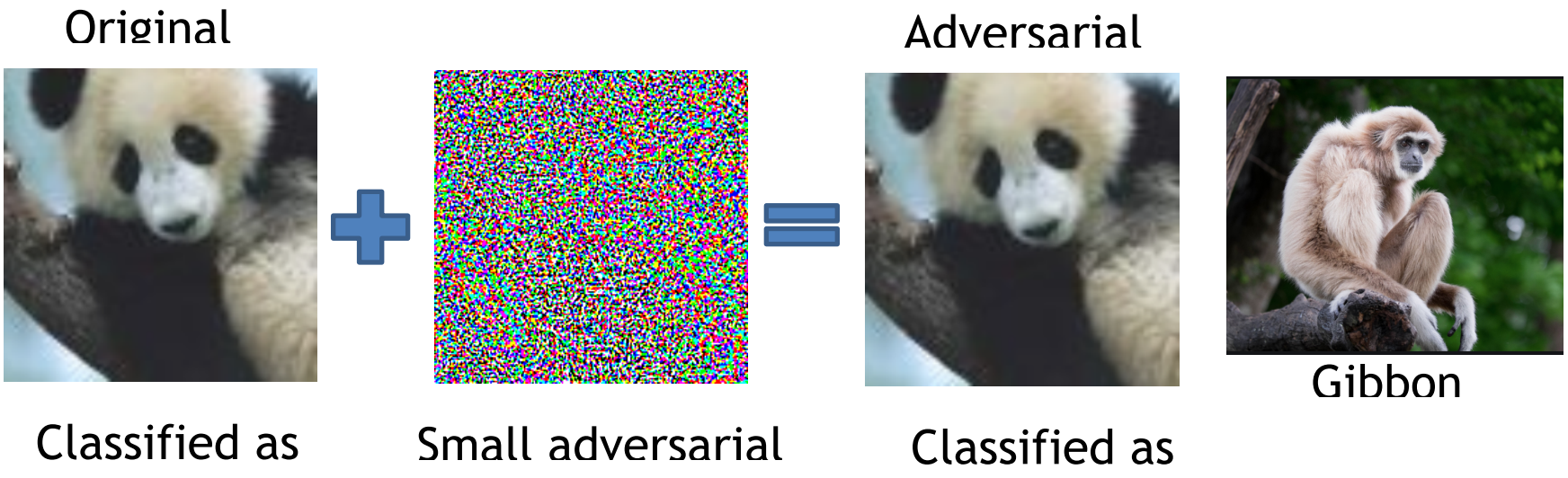






Adversarial Examples

- ***Adversarial examples*** **are inputs to ML models that an attacker intentionally designed to cause the model to make mistakes**

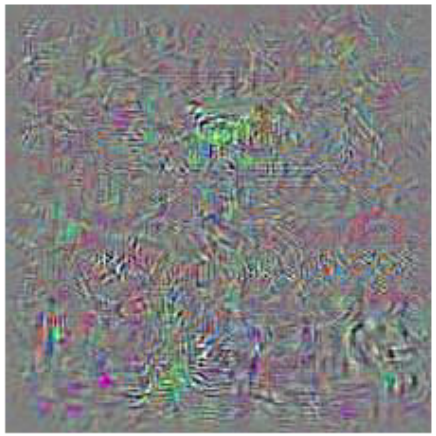


Picture from: https://blog.floydhub.com/introduction-to-adversarial-machine-learning/

Adversarial Examples

Schoolbus

+



Perturbation

(rescaled for visualization)

=

Ostrich

Picture from: https://blog.floydhub.com/introduction-to-adversarial-machine-learning/

Adversarial Examples

- Fast gradient sign method (FGSM) attack:

Treat a model as linear model, then take a step to the direction of the gradient. In reality, most models are not linear, therefore models are not robust.

- Input vector x and label y; Loss function L(x, y)

$$x_{adv} = x + \epsilon \, . \, sign(\nabla_x L(x, y))$$

Picture from: https://blog.floydhub.com/introduction-to-adversarial-machine-learning/

Adversarial Examples

- Iterative Fast gradient sign method (IFGSM) or Basic Iterative Method (BIM) attack:
- Input vector x and label y

$$x_{adv}^{0} = x$$

for t = 1 to T do

$$x_{adv}^{t} = x_{adv}^{t-1} + \epsilon \,.\, sign(\nabla_{(} x_{adv}^{t-1}) L(x_{adv}^{t-1}, y))$$

Ref: https://medium.com/swlh/gradient-based-adversarial-attacks-an-introduction-526238660dc9

Picture from: https://blog.floydhub.com/introduction-to-adversarial-machine-learning/

IIT JAMMU

Adversarial Examples

- Projected gradient descent (PGD) attacks
  - Taking the number of iterations steps n instead of a single step to generate the attack images.
  - Norm: and Input vector x, label y ; Constraint:($L_\infty \leq \epsilon$)

$$x^0_{adv} = x$$

for t = 1 to T do

$$x^t_{adv} = clip_\epsilon(x^{t-1}_{adv} + \delta \,.\, sign(\nabla_(x^{t-1}_{adv}) L(x^{t-1}_{adv}, y)))$$

Better results, but slower comparing to FGSM

Ref: https://towardsdatascience.com/know-your-enemy-7f7c5038bdf3

Picture from: https://blog.floydhub.com/introduction-to-adversarial-machine-learning/

IIT JAMMU

Attack Taxonomy

- A ***taxonomy of adversarial attacks*** is typically derived based on an assumed ***threat model*** regarding the goal, knowledge, and target strategy of the adversary
- Adversary's goal
  - *Poisoning attack, evasion attack*: cause the ML model to perform incorrectly
  - *Privacy attack*: acquire knowledge about the training data or the model
  - *Availability attack*: cause the ML model to be become unavailable
- Adversary's knowledge
  - *White-box attack*: the adversary has full knowledge of the ML model
  - *Black-box attack*: has no knowledge of the ML model
  - *Gray-box attack*: has some knowledge of the ML model
- Adversary's target strategy
  - *Targeted attack*: cause the ML model to output a target label for an input
  - *Non-targeted attack*: cause the ML to output any incorrect label for an input

Few Approaches

- Supervised
  - Bayes Prediction (Gold Accuracy)
  - Linear Regression, Logistic Regression
  - Support Vector Machine
  - Decision Trees, Random Forest
  - Neural Network (Universal Approximation) and its variants, Deep Neural Networks
- Unsupervised:
  - K-means, Hierarchical, DBSCAN, Auto-Encoders

Applications of ML in Security

- Cyber Threat Identification:
  - Using machine learning to detect malicious activity and stop attacks
  - Using machine learning to enhance human analysis
- Fighting AI Threats:
  - Using machine learning to automate repetitive security tasks
  - Using machine learning to close zero-day vulnerabilities
  - Email Monitoring etc.
- AI-based Antivirus Software
- User Behaviour Modelling

IIT JAMMU

Attack Types on ML Model

- White Box : Model & Parameters Known
  - Perturb or add noise (adversarial) to sample for Model Fooling
  - Applications: Spam/sentiment/emotion detections, Classifications
- Black Box: Model & Parameters Un-Known
  - Generating the train samples from Model
  - Applications: Sensitive information leaking (authentications biometric, access to sensitive zones)

IIT JAMMU

Attack Types on ML Model

- Poisoning Attack: Perturb train data, mis-label data
  - Application use case: Re-training models such as spam detection, sentiments and emotion detections (Text Attacks)
- Evasion Attacks: Manipulate data during deployment
  - Obfuscating the content of malware or spam emails, product reviews etc.
- Model Extraction: Probing a black box Model (reconstruct the model or extract the data it was trained on)
  - Applications: Training data or the model itself is sensitive and confidential (eg: authentication, stock market trend prediction etc.)

IIT JAMMU

Adversarial Sample Generation

- Limited-memory BFGS (L-BFGS)
  - Advantages: Effective at generating adversarial examples.
  - Disadvantages: Very computationally intensive, as it is an optimized method with box constraints.
- FastGradient Sign method (FGSM)
  - Advantages: Comparably efficient computing times.
  - Disadvantages: Perturbations are added to every feature.
- Jacobian-based Saliency Map Attack (JSMA)
  - Advantages: Very few features are perturbed.
  - Disadvantages: More computationally intensive than FGSM.
- Deepfool Attack
  - Advantages: Effective at producing adversarial examples, with fewer perturbations and higher misclassification rates.
  - Disadvantages: More computationally intensive than FGSM and JSMA. Also, adversarial examples are likely not optimal.

Ref: https://viso.ai/deep-learning/adversarial-machine-learning/

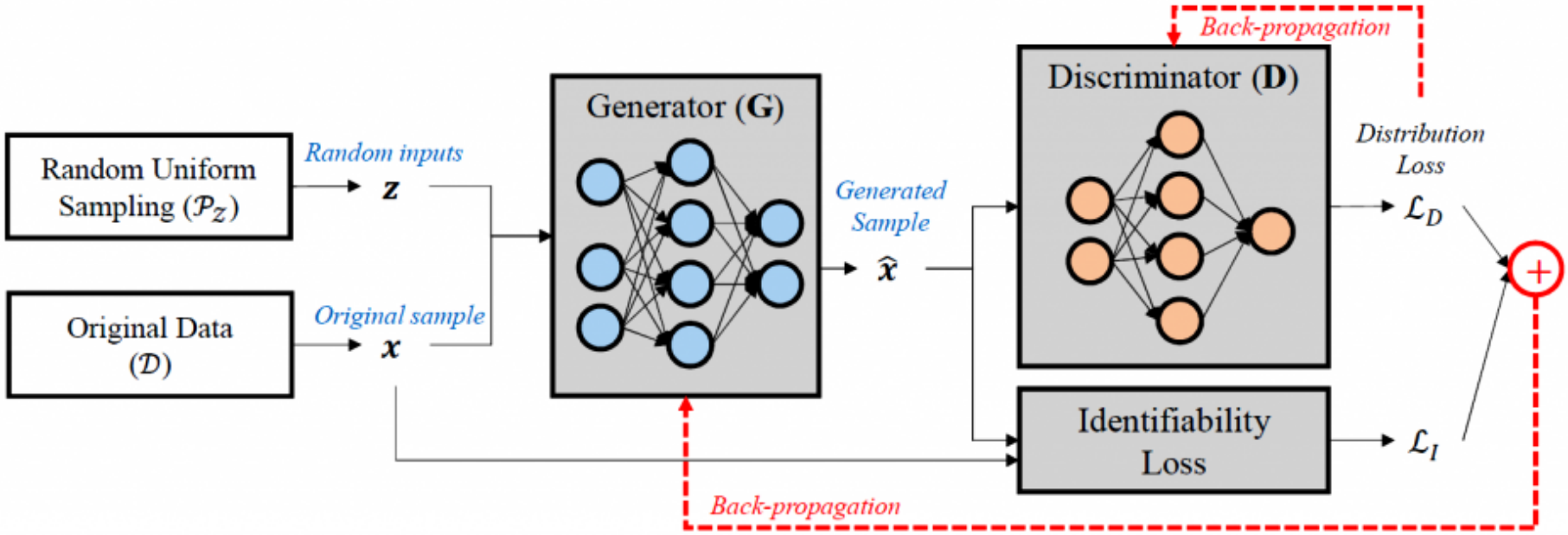
IIT JAMMU

Adversarial Sample Generation

- Carlini & Wagner Attack (C&W)
  - Advantages: Very effective at producing adversarial examples. Also, it can defeat some adversarial defenses.
  - Disadvantages: More computationally intensive than FGSM, JSMA, and Deepfool.
- Generative Adversarial Networks (GAN)
  - Advantages: Generation of samples different from the ones used in training.
  - Disadvantages: Training a Generate Adversarial Network is very computationally intensive and can be highly unstable.
- Zeroth-order optimization attack (ZOO)
  - Advantages: Similar performance to the C&W attack. No training of substitute models or information on the classifier is required.
  - Disadvantages: Requires a large number of queries to the target classifier.

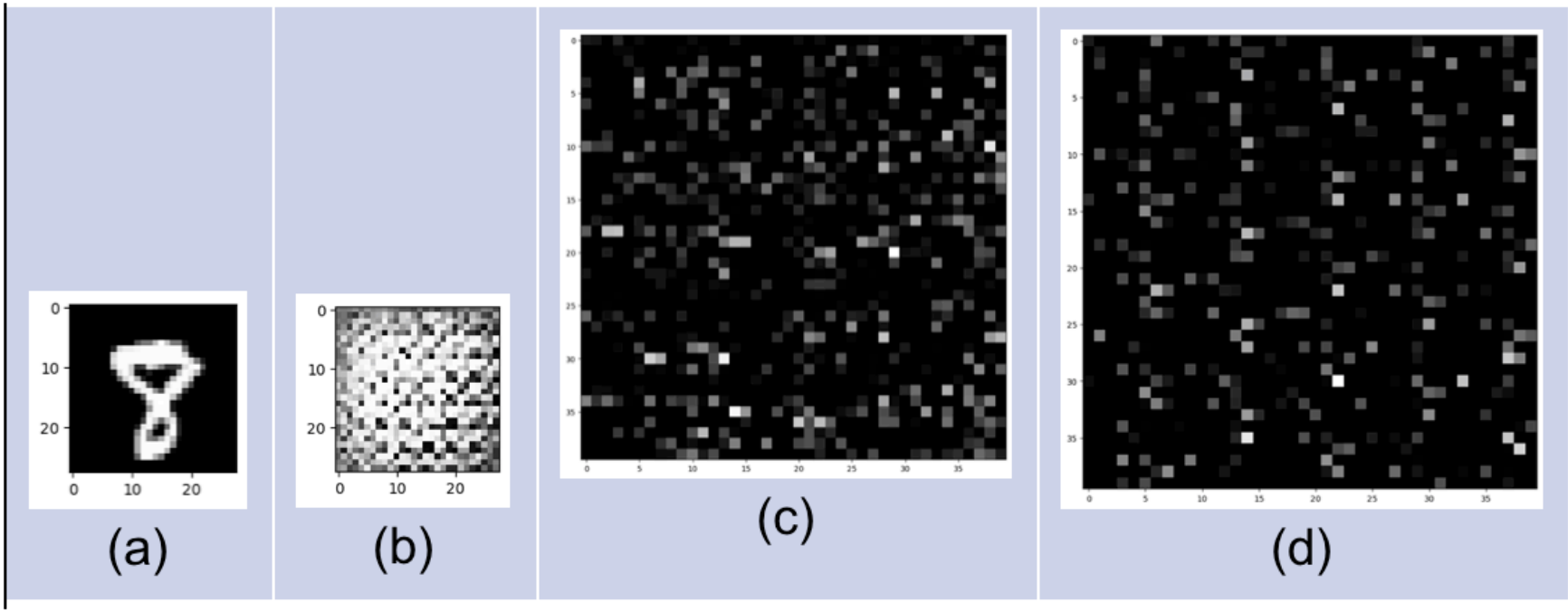
Ref: https://viso.ai/deep-learning/adversarial-machine-learning/

Adversarial Sample Generation: GAN



Img Ref: www.vanderschaar-lab.com

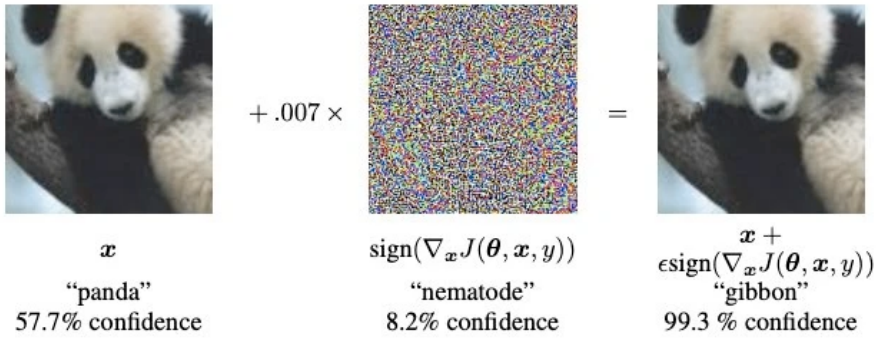
Adversarial Sample Generation: GAN contd…



Figures(a) and (b) are original and adversarial sample (generated by GAN) respectively. Figures (c) and (d) resent the visualisation of 2nd last layer of Deep Neural Network model for original and adversarial samples respectively.

Experiments: **0.350** cosine similarity using t-SNE in feature vector space.

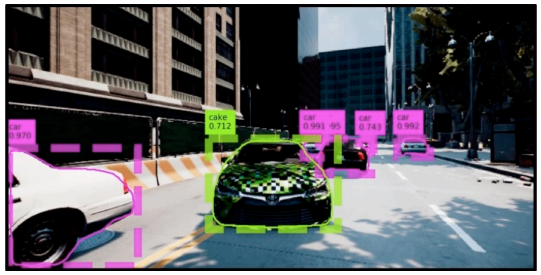
Attack in Realtime: Adversarial Samples



Goodfellow et al, 2015

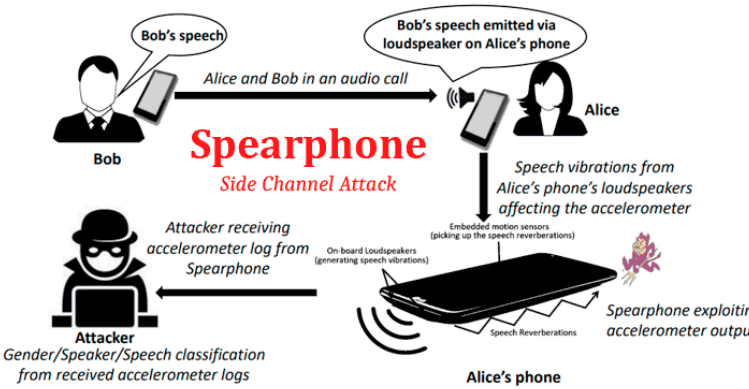
Prediction	Confidence	Texts
Positive	99.7%	This is a **unique** masterpiece made by the best director ever lived in the ussr. He knows the art of film making and can use it very well. If you find this movie, buy or copy it!
Negative	86.2%	This is a **sole** masterpiece made by the best director ever lived in the ussr. He knows the art of film making and can use it very well. If you find this movie, buy or copy it!

Wang et al, UAI2021



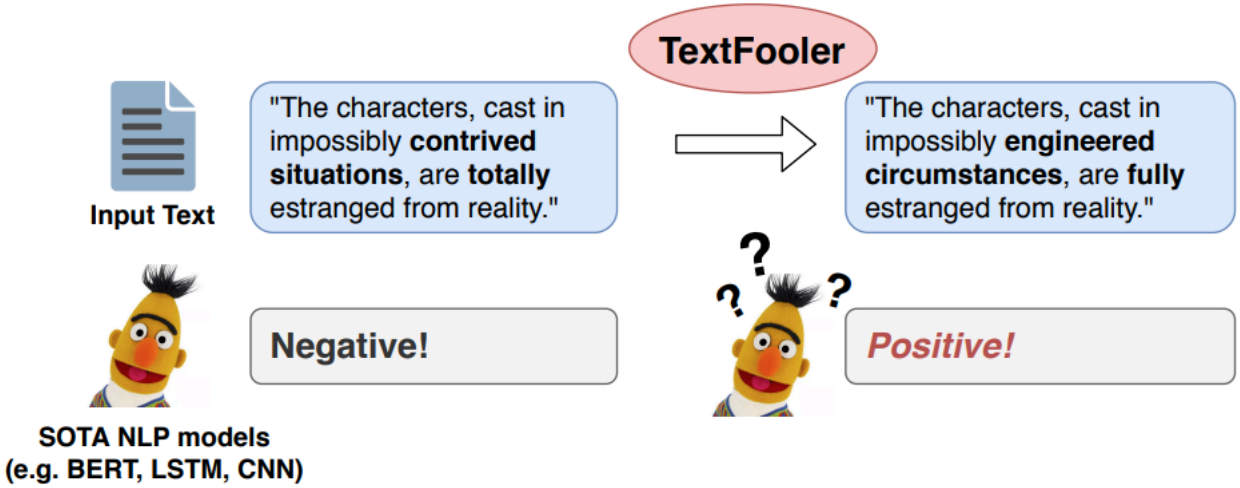
Zhang et al, ICLR 2019

Biggio et al, PR 2018



https://thehackernews.com/2019/07/android-side-channel-attacks.html

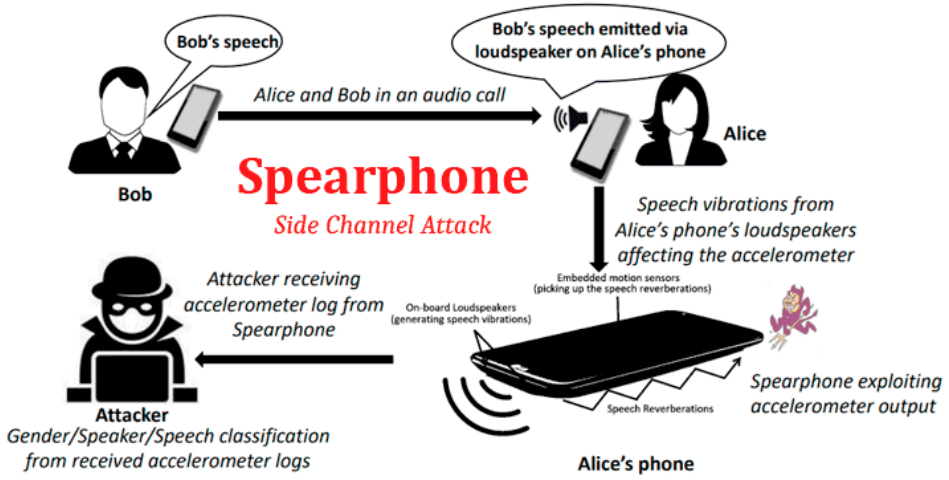
Attack in Realtime: Adversarial Samples



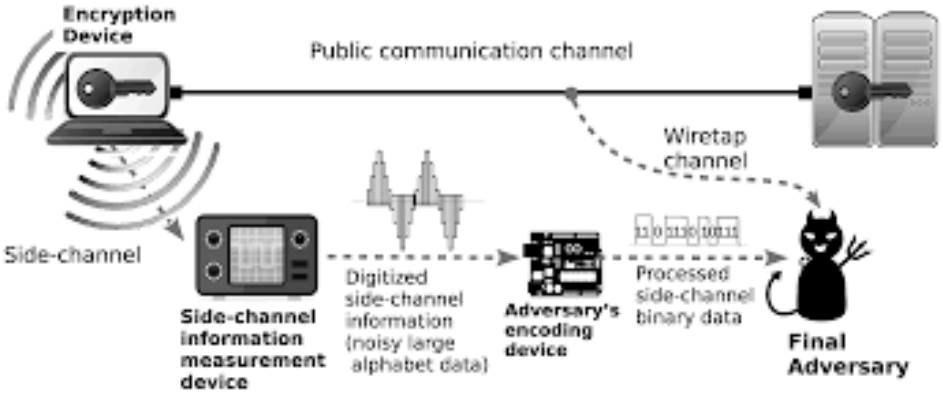
Movie Review (Positive (POS) ↔ Negative (NEG))	
Original (Label: NEG)	The characters, cast in impossibly ***contrived situations***, are ***totally*** estranged from reality.
Attack (Label: POS)	The characters, cast in impossibly ***engineered circumstances***, are ***fully*** estranged from reality.
Original (Label: POS)	It cuts to the ***knot*** of what it actually means to face your ***scares***, and to ride the ***overwhelming*** **metaphorical wave** that life wherever it takes you.
Attack (Label: NEG)	It cuts to the ***core*** of what it actually means to face your ***fears***, and to ride the ***big*** **metaphorical wave** that life wherever it takes you.

Jin et al 2020: https://arxiv.org/pdf/1907.11932.pdf

Attack in Realtime: Hardware



https://thehackernews.com/2019/07/android-side-channel-attacks.html



https://www.mdpi.com/

IIT JAMMU

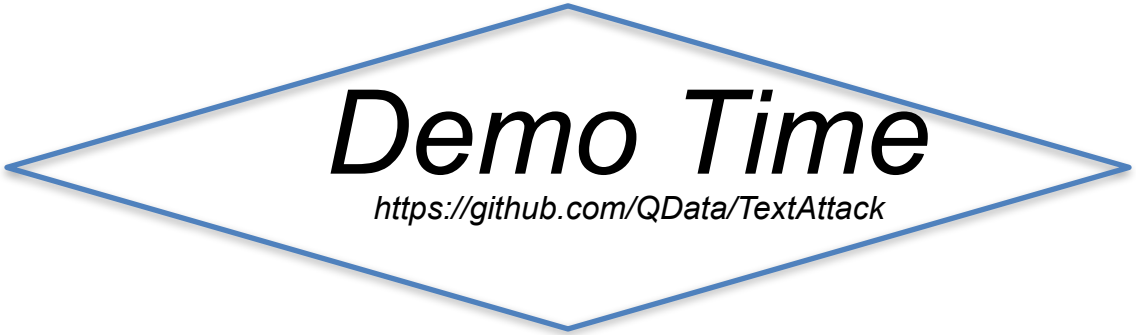
Research

Attack	Publication	Similarity	Attacking Capability	Algorithm	Apply Domain
L-BFGS	(Szegedy et al., 2013)	l_2	White-Box	Iterative	Image Classification
FGSM	(Goodfellow et al., 2014b)	l_∞,l_2	White-Box	Single-Step	Image Classification
Deepfool	(Moosavi-Dezfooli et al., 2016)	l_2	White-Box	Iterative	Image Classification
JSMA	(Papernot et al., 2016a)	l_2	White-Box	Iterative	Image Classification
BIM	(Kurakin et al., 2016a)	l_∞	White-Box	Iterative	Image Classification
C & W	(Carlini & Wagner, 2017b)	l_2	White-Box	Iterative	Image Classification
Ground Truth	(Carlini et al., 2017)	l_0	White-Box	SMT solver	Image Classification
Spatial	(Xiao et al., 2018b)	Total Variation	White-Box	Iterative	Image Classification
Universal	(Metzen et al., 2017b)	l_∞, l_2	White-Box	Iterative	Image Classification
One-Pixel	(Su et al., 2019)	l_0	White-Box	Iterative	Image Classification
EAD	(Chen et al., 2018)	$l_1 + l_2$, l_2	White-Box	Iterative	Image Classification
Substitute	(Papernot et al., 2017)	l_p	Black-Box	Iterative	Image Classification
ZOO	(Chen et al., 2017)	l_p	Black-Box	Iterative	Image Classification
Biggio	(Biggio et al., 2012)	l_2	Poisoning	Iterative	Image Classification
Explanation	(Koh & Liang, 2017)	l_p	Poisoning	Iterative	Image Classification
Zugner's	(Zügner et al., 2018)	Degree Distribution, Coocurrence	Poisoning	Greedy	Node Classification
Dai's	(Dai et al., 2018)	Edges	Black-Box	RL	Node & Graph Classification
Meta	(Zügner & Günnemann, 2019)	Edges	Black-Box	RL	Node Classification
C & W	(Carlini & Wagner, 2018)	max dB	White-Box	Iterative	Speech Recognition
Word Embedding	(Miyato et al., 2016)	l_p	White-Box	One-Step	Text Classification
HotFlip	(Ebrahimi et al., 2017)	letters	White-Box	Greedy	Text Classification
Jia & Liang	(Jia & Liang, 2017)	letters	Black-Box	Greedy	Reading Comprehension
Face Recognition	(Sharif et al., 2016)	physical	White-Box	Iterative	Face Recognition
RL attack	(Huang et al., 2017)	l_p	White-Box	RL	

Table from: Xu et al. (2019) - Adversarial Attacks and Defenses in Images, Graphs and Text: A Review

Attack Resistive Models

- Exploration of Learning Encrypted Data
  - Privacy Preserving (Few research)
- One way Learning
  - Training Data/Model Parameters can not be Estimated (Biometric Authentication) or with less probability
- Learning with Adversarial Samples
- Open Research Area



Good Resource: https://github.com/thunlp/TAADpapers

IIT JAMMU

References

[1] A theory of the learnable, Valiant, LG (1984), Communications of the ACM 27(11):1134 -1142.

[2] Learnability and the VC-dimension, A Blumer, A Ehrenfeucht, D Haussler, M Warmuth - Journal of the ACM, 1989.

[3] Tramèr, Florian, et al. "Stealing Machine Learning Models via Prediction APIs." USENIX security symposium. Vol. 16. 2016.

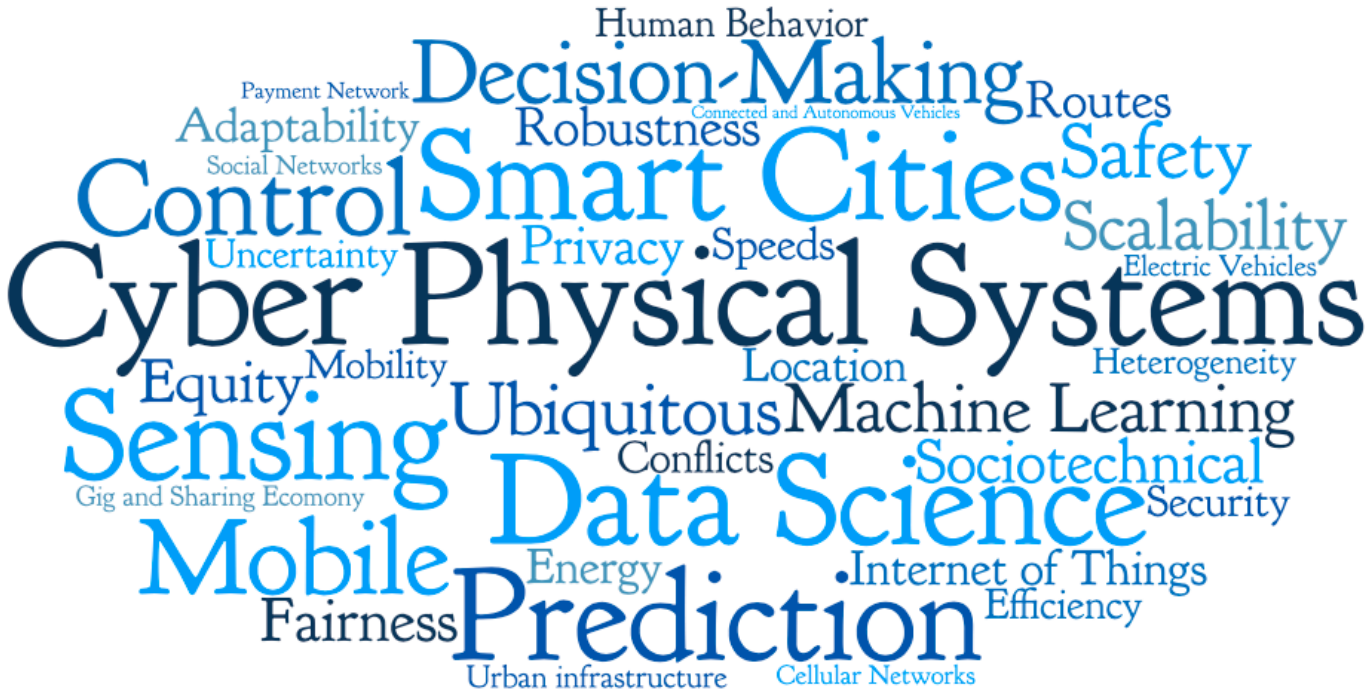
[4] T. Lee, B. Edwards, I. Molloy, and D. Su. Defending against neural network model stealing attacks using deceptive perturbations. In 2019 IEEE Security and Privacy Workshops (SPW), 2019.

[5] Manish Kesarwani, Bhaskar Mukhoty, Vijay Arya, and Sameep Mehta. Model extraction warning in MLaaS paradigm. In Proceedings of the 34th Annual Compu Security Applications Conference. ACM, 2018.

[6] Mika Juuti, Sebastian Szyller, Alexey Dmitrenko, Samuel Marchal, and N. Asokan. PRADA: Protecting against DNN model stealing attacks. In 2019 IEEE European Symposium on Security and Privacy (EuroS&P), 2019.

[7] Tribhuvanesh Orekondy, Bernt Schiele, and Mario Fritz. Knockoff Nets: Stealing functionality of black-box models. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 4954–4963, 2019b.

Thank You


Human Behavior
Payment Network
Decision-Making
Connected and Autonomous Vehicles
Routes
Adaptability
Robustness
Social Networks
Control
Smart Cities
Safety
Scalability
Uncertainty
Privacy
Speeds
Electric Vehicles
Cyber Physical Systems
Equity
Mobility
Location
Heterogeneity
Sensing
Ubiquitous
Machine Learning
Conflicts
Sociotechnical
Gig and Sharing Ecomony
Data Science
Security
Mobile
Energy
Internet of Things
Fairness
Prediction
Efficiency
Urban infrastructure
Cellular Networks