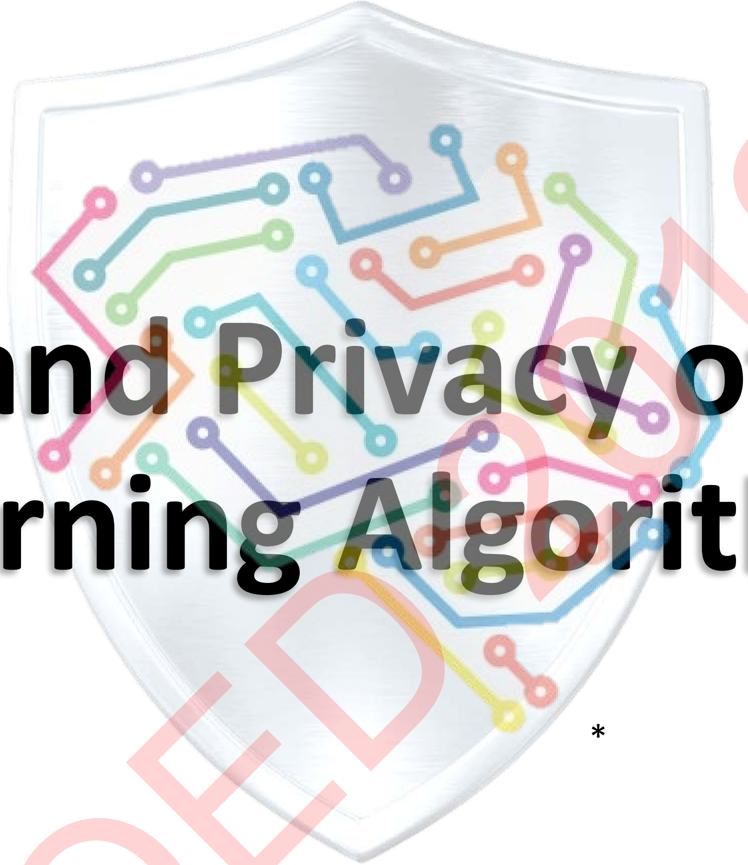
Security and Privacy of Machine Learning Algorithms


*

Sandip Kundu

National Science Foundation

on leave from

University of Massachusetts, Amherst

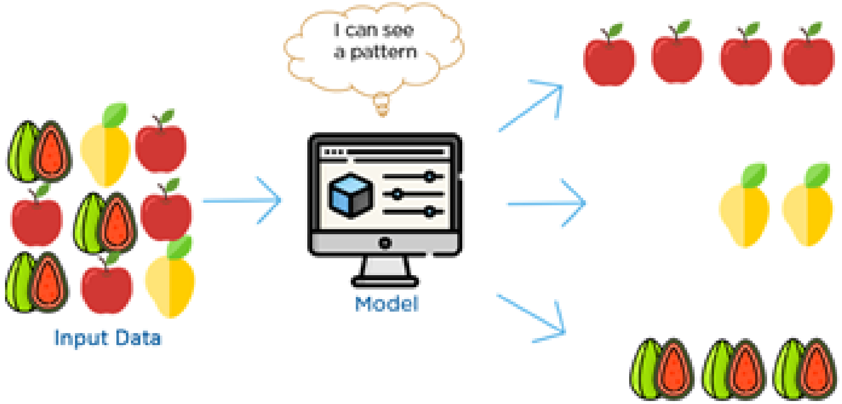
* Image - Created by Patrickss - Freepik.com

Machine Learning

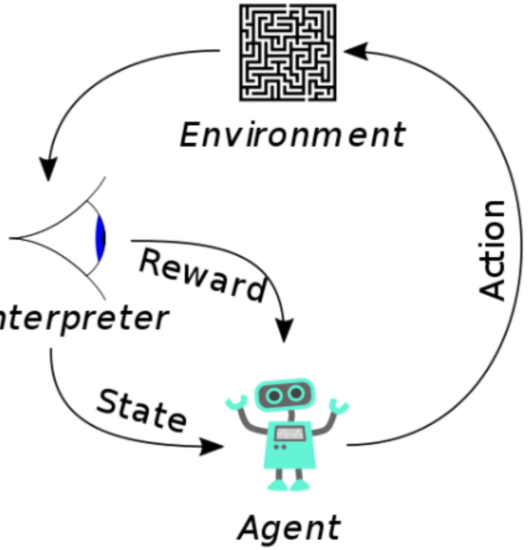
Training Data → Mathematical Model → Decision/ Prediction

Types of Machine Learning

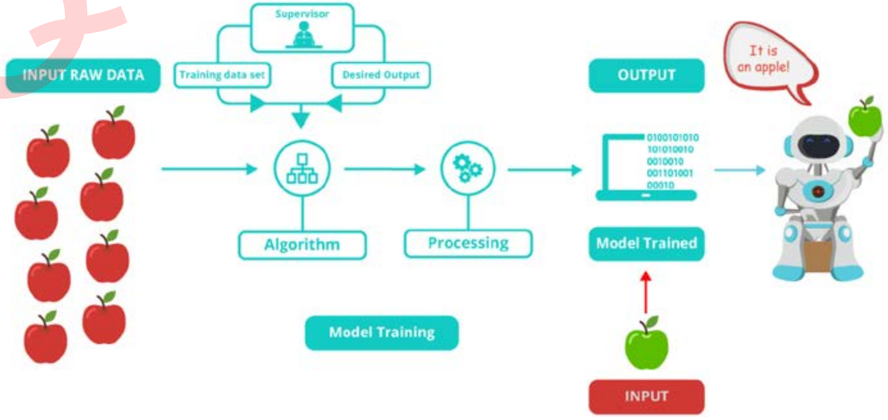
Unsupervised



Reinforcement

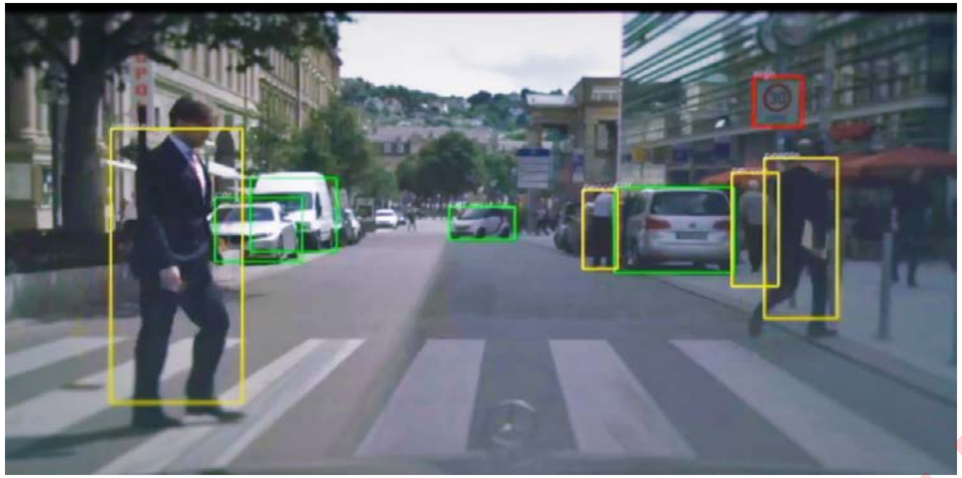


Supervised



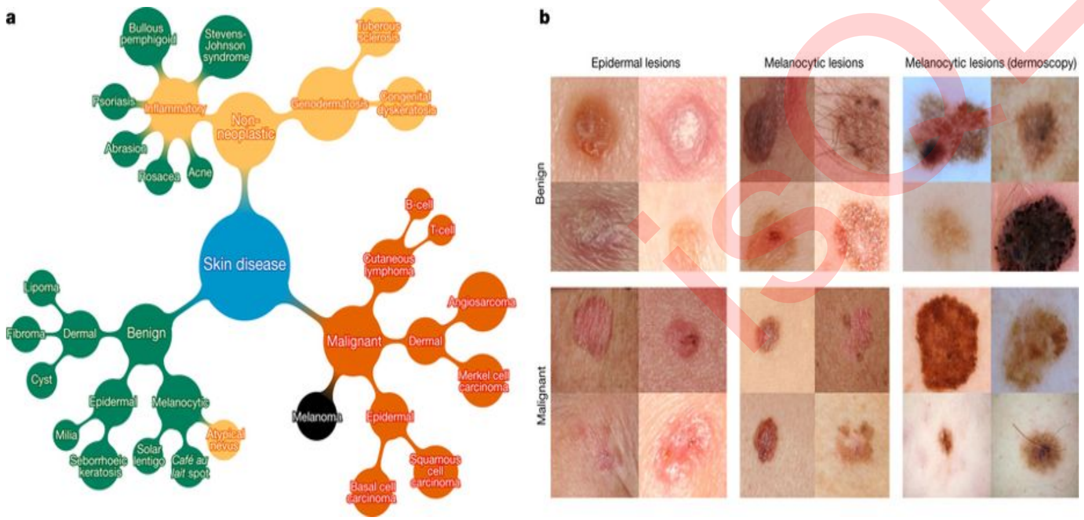
Major applications

Self-driving Cars

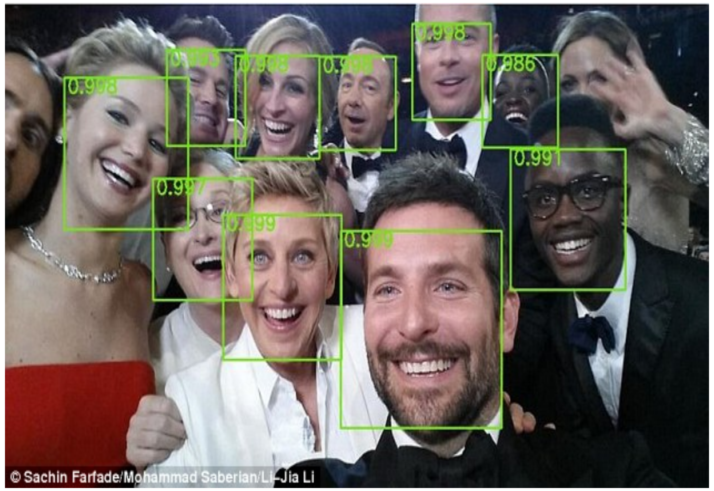


Cybersecurity

Healthcare



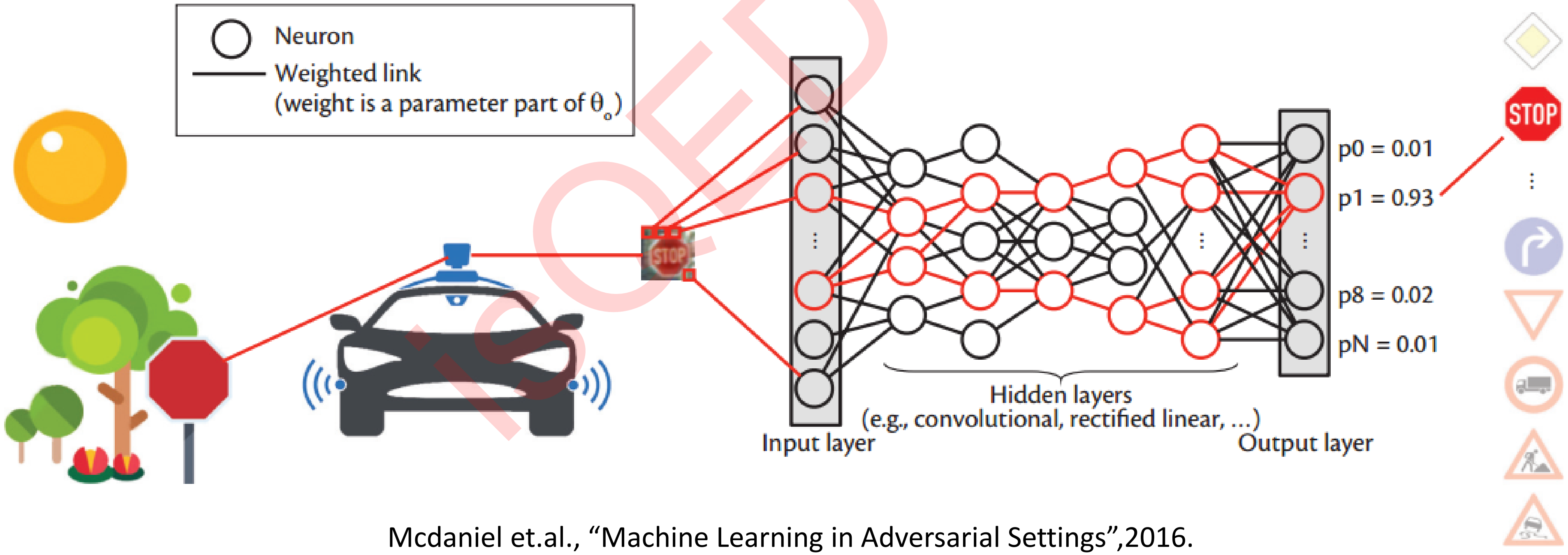
Facial Recognition



Speech Recognition

Self-driving Cars

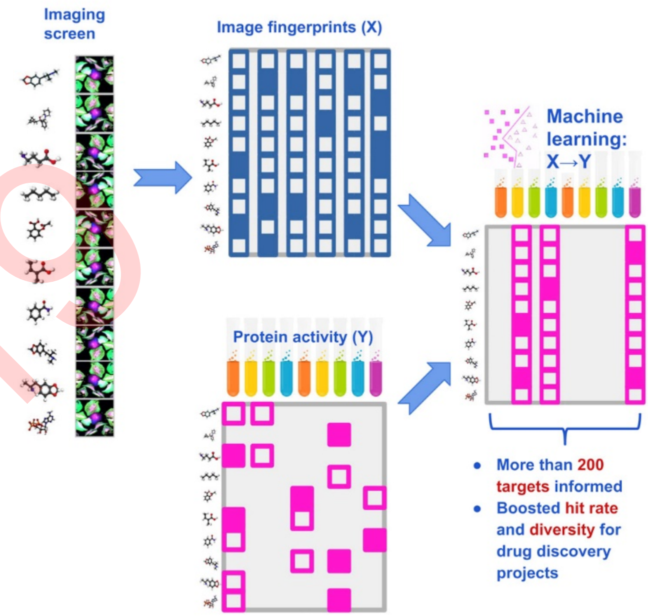
- Cars incorporating systems to assist or replace drivers
  - Ex. automatic parking, Waymo
- Self-driving cars with ML infrastructure will become commonplace
  - Ex. NVIDIA DRIVETM PX 2 – open AI car computing system



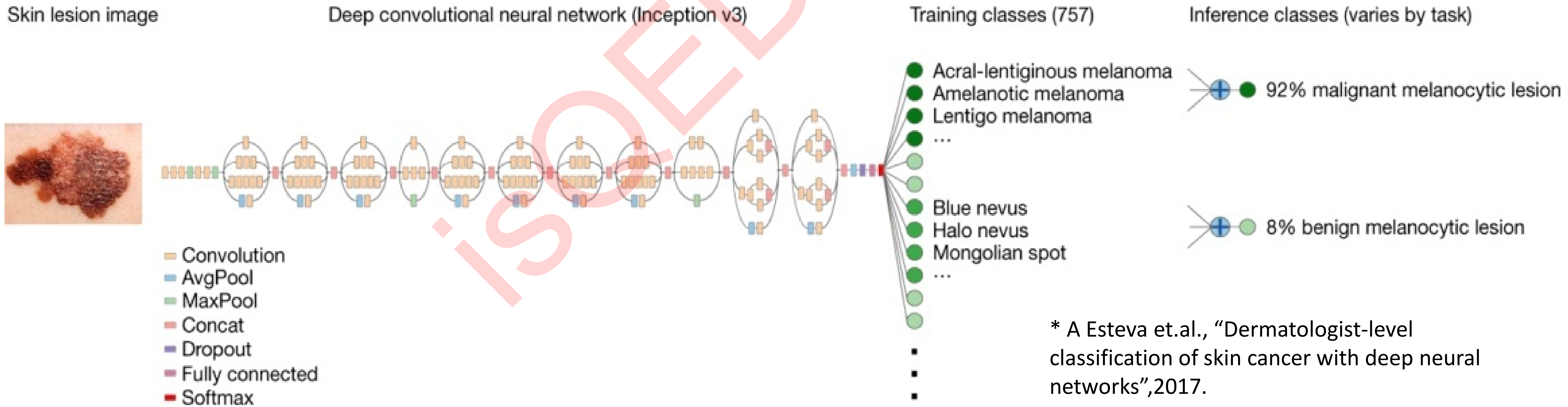
Mcdaniel et.al., “Machine Learning in Adversarial Settings”,2016.

Healthcare Applications

- ❖ Diagnosis in Medical Imaging
- ❖ Treatment Queries and Suggestions
- ❖ Drug Discovery
- ❖ Personalized Medicine



* Simm, Jaak, et al. "Repurposing high-throughput image assays enables biological activity prediction for drug discovery." *Cell chemical biology* (2018)



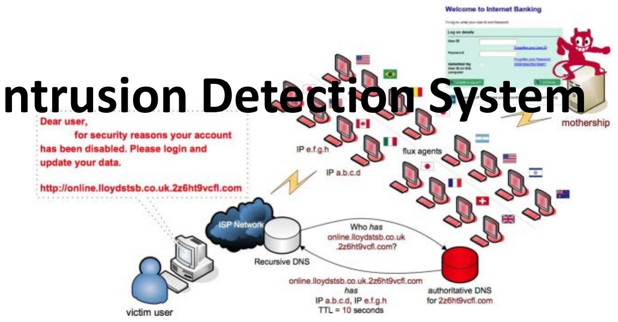
* A Esteva et.al., “Dermatologist-level classification of skin cancer with deep neural networks”,2017.

Cybersecurity

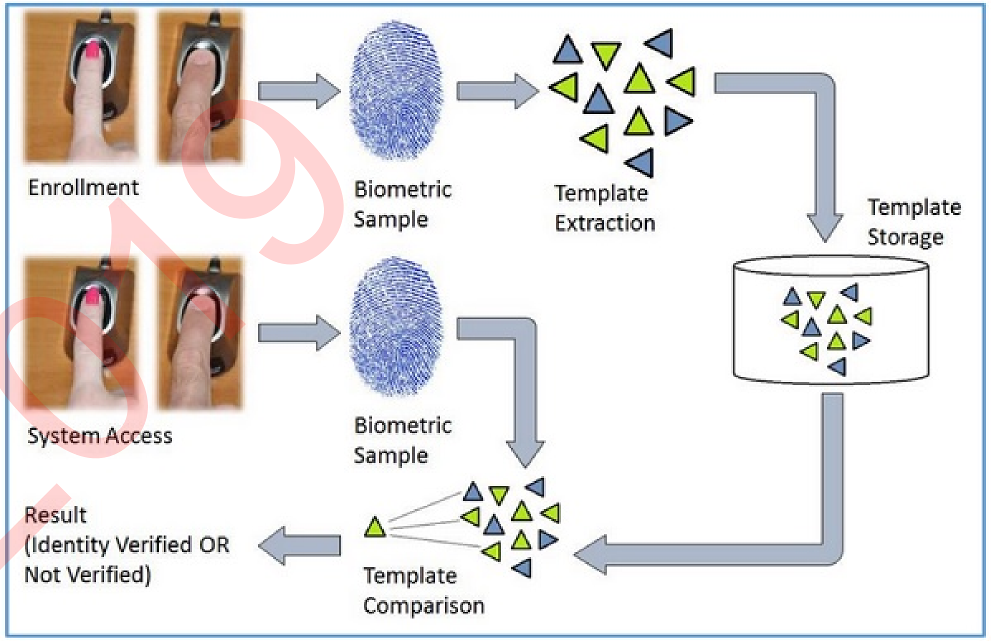
Spam Filtering

* http://www.thenonprofittimes.com/news-articles/rate-legit-emails-getting-caught-spam-filters-jumped/

Intrusion Detection System

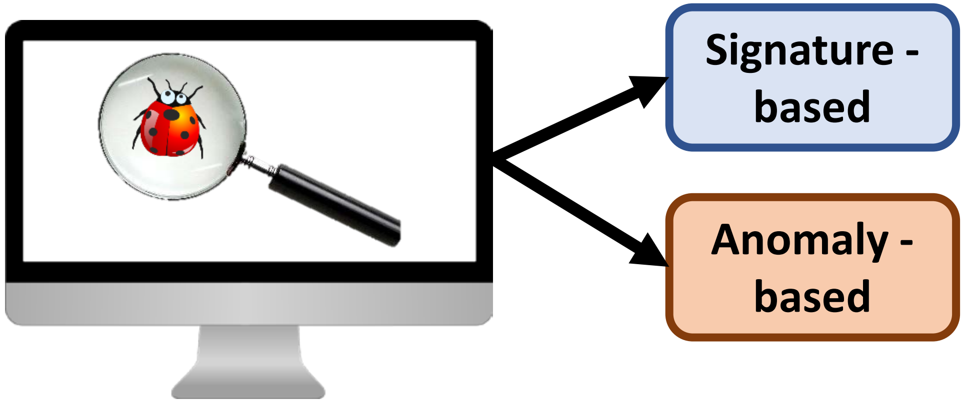


Biometrics ID



* https://www.tutorialspoint.com/biometrics/biometrics_overview.htm

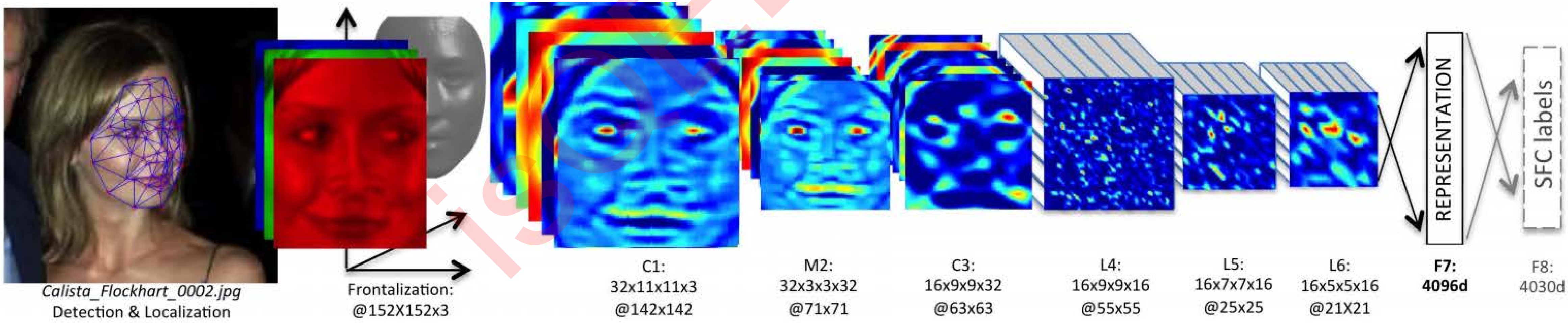
Malware Detection



Facial Recognition

- ❖ Secure Authentication and Identification
  - o Apple FaceID
  - o FBI database – criminal identification
- ❖ Customer Personalization
  - o Ad targeting
  - o Snapchat

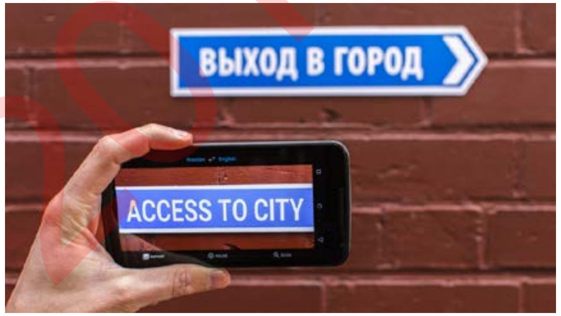
* Posterscope, Ouividi EYE Corp Media, Engage M1 – GMC Arcadia



Taigman et.al.,"DeepFace: Closing the Gap to Human-Level Performance in Face Verification",2014

Other Machine Vision Applications

- **Digital annotation** of real-world
  - Text, language recognition – E.g. Billboards, auto-translation
  - Geo-tagging Landmarks
  - Integration with other services – E.g. ratings for restaurant, directions

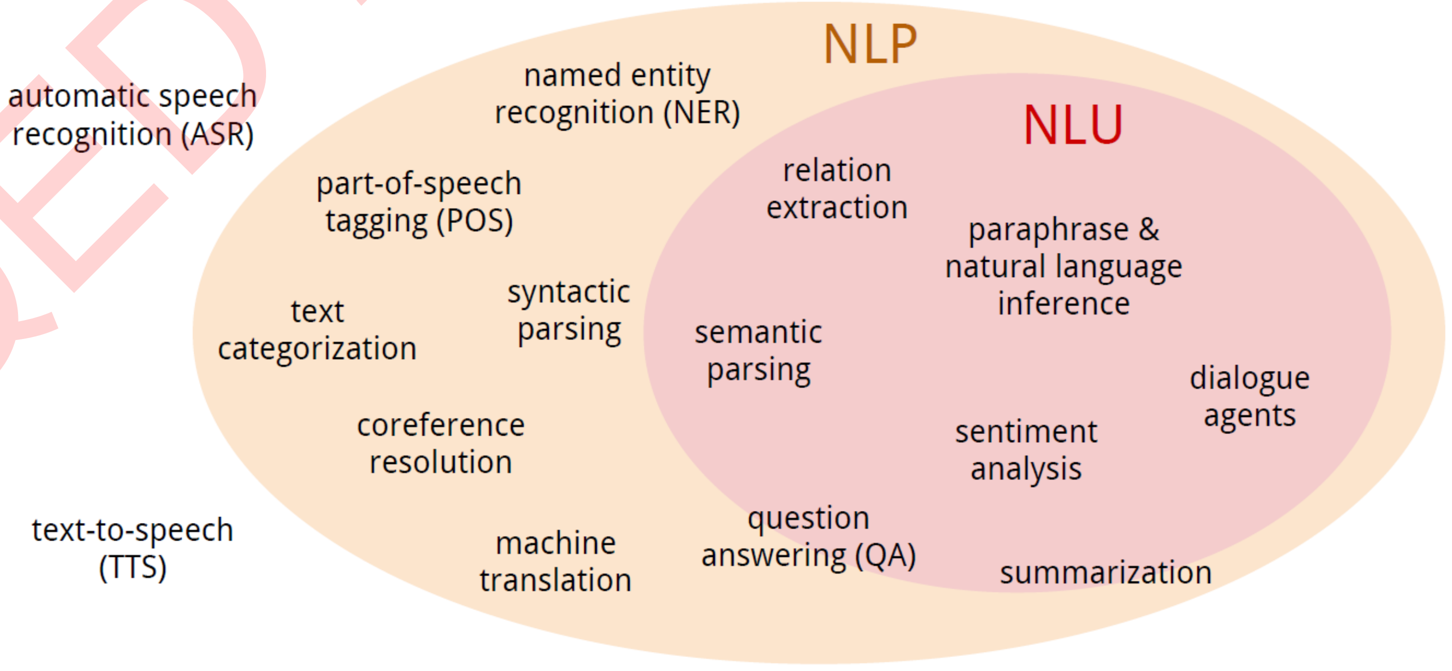


Google Lens

- **Augmented Reality**
  - **Gaming** – adaptive integration with real-world
  - **Augmented Retail** – E.g. Clothes Fitting

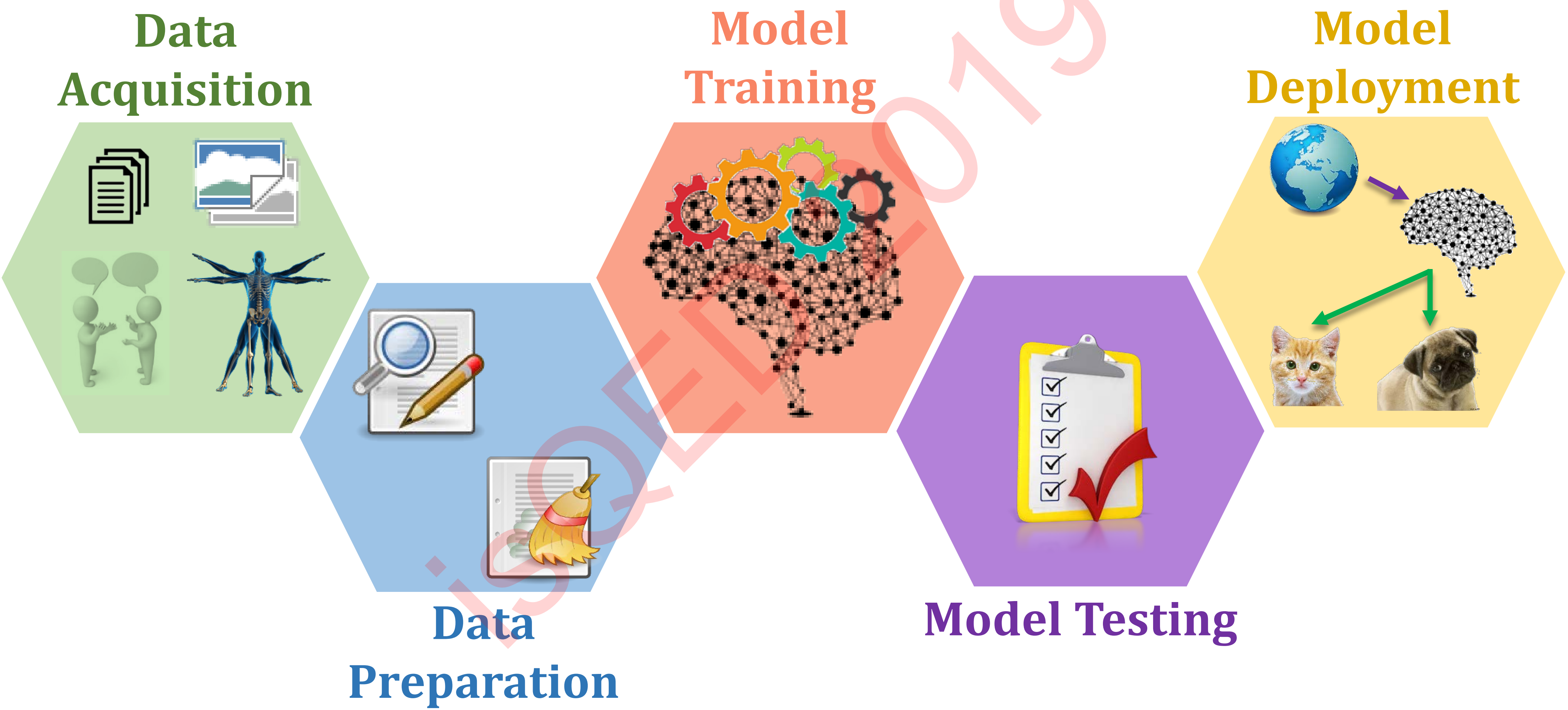
Speech Recognition

- ❖ Envisioned in science fiction since 1960's
  - o HAL 9000, Star Trek
- ❖ Natural Language Processing (NLP) has gained increased importance
  - o Modeling large vocabularies, accents – translation, transcription services
  - o **Smartphones** – Apple Siri, Google Assistant, Samsung Bixby
  - o Home - Amazon's Echo/Alexa,
  - o IBM Watson



http://nlp.stanford.edu/~wcmac/papers/20140716-UNLU.pdf

Machine learning (ML) Process



Machine Learning Security and Privacy

Introduction

- ML algorithms in real-world applications mainly focus on **accuracy** (effectiveness) **or/and efficiency** (dataset, model size)
  - Few techniques and design decisions to keep the ML models ***secure and robust***!

- Machine Learning as a Service (MLaaS) and Internet of Things (IoT) further complicate matters
  - Attacks can **compromise millions of customers**’ security and privacy
  - Concerns about **Ownership** of data, model

ML Vulnerabilities

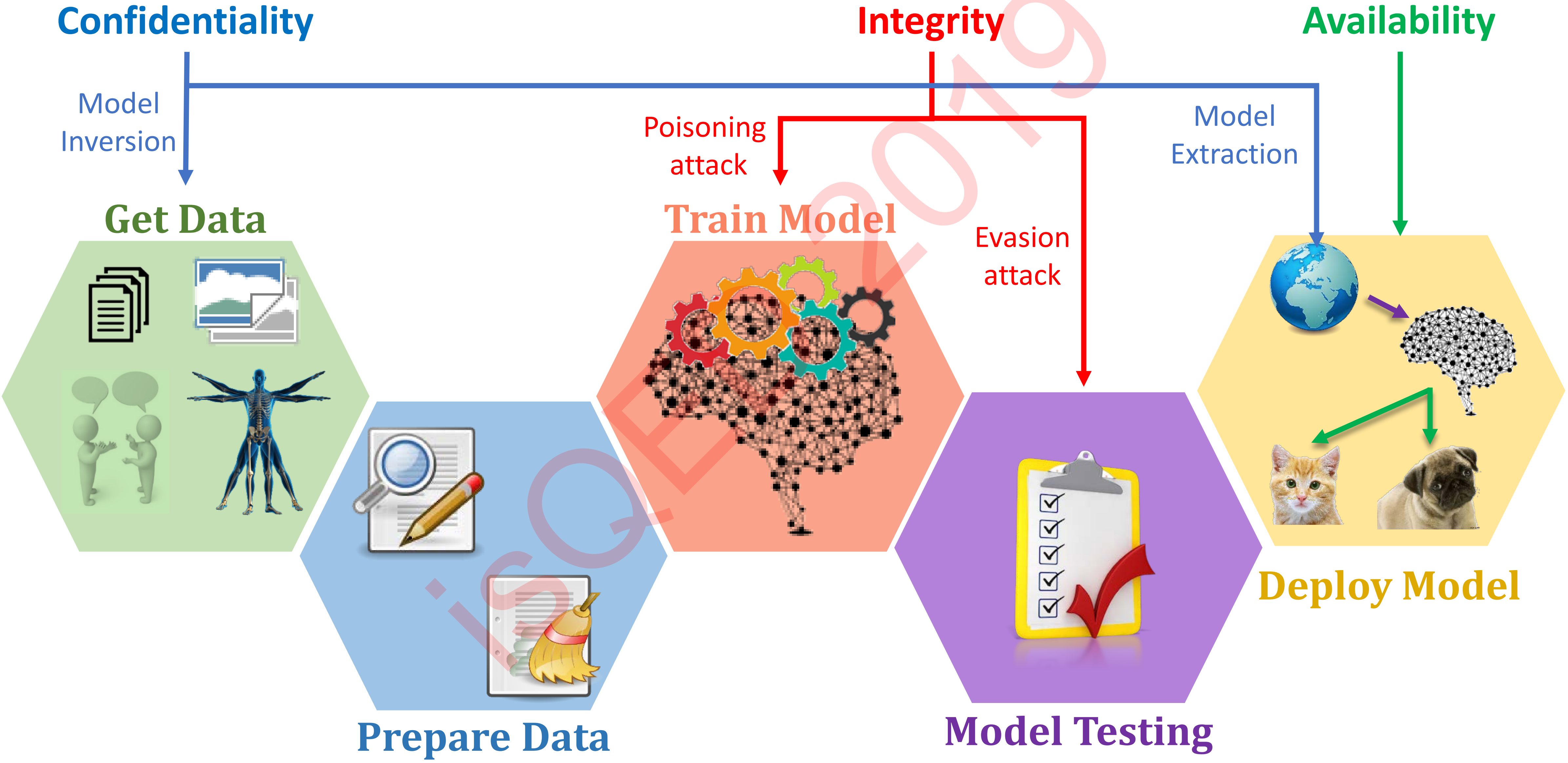
- Key vulnerabilities of machine learning systems
  - ML models often derived from **fixed datasets**
  - Assumption of similar distribution between training and real-world data
    - **Coverage** issues for complex use cases
    - Need **large datasets**, **extensive data annotation**, **testing**

- Strong adversaries against ML systems
  - ML algorithms **established** and **public**
  - Attacker can leverage ML knowledge for **Adversarial Machine Learning** (AML)
    - **Reverse engineering** model parameters, test data – **Financial incentives**
    - **Tampering** with the trained model – **compromise security**

Classification of Security and Privacy Concerns

- Attack Influence
  - **Causative** – manipulate ***training data*** to **introduce** vulnerability
  - **Exploratory** – **find and exploit** vulnerability during ***classification***
- Attack Specificity
  - **Targeted** – **focused** on specific or small set of points
  - **Indiscriminate** – **flexible** goals
- Security Violation
  - **Confidentiality** – **extract** model **parameters** or **private data**
  - **Integrity** – **compromise** model to produce false positives/negatives
  - **Availability** – **render** model **unusable**

Security and Privacy Concerns



Confidentiality

Training Data Confidentiality

- **Training data** is **valuable** and **resource-intensive** to obtain
  - o Collection of **large datasets**
  - o Data **annotation** and **curation**
  - o Data **privacy** in critical applications like healthcare
- Ensuring training data **confidentiality** is **critical**

QUARTZ

Waymo's driverless cars have logged 10 million miles on public roads

By Jane C. Hu • October 10, 2018

The New York Times

Sloan Kettering's Cozy Deal With Start-Up Ignites a New Uproar

By Charles Ornstein and Katie Thomas

Sept. 20, 2018

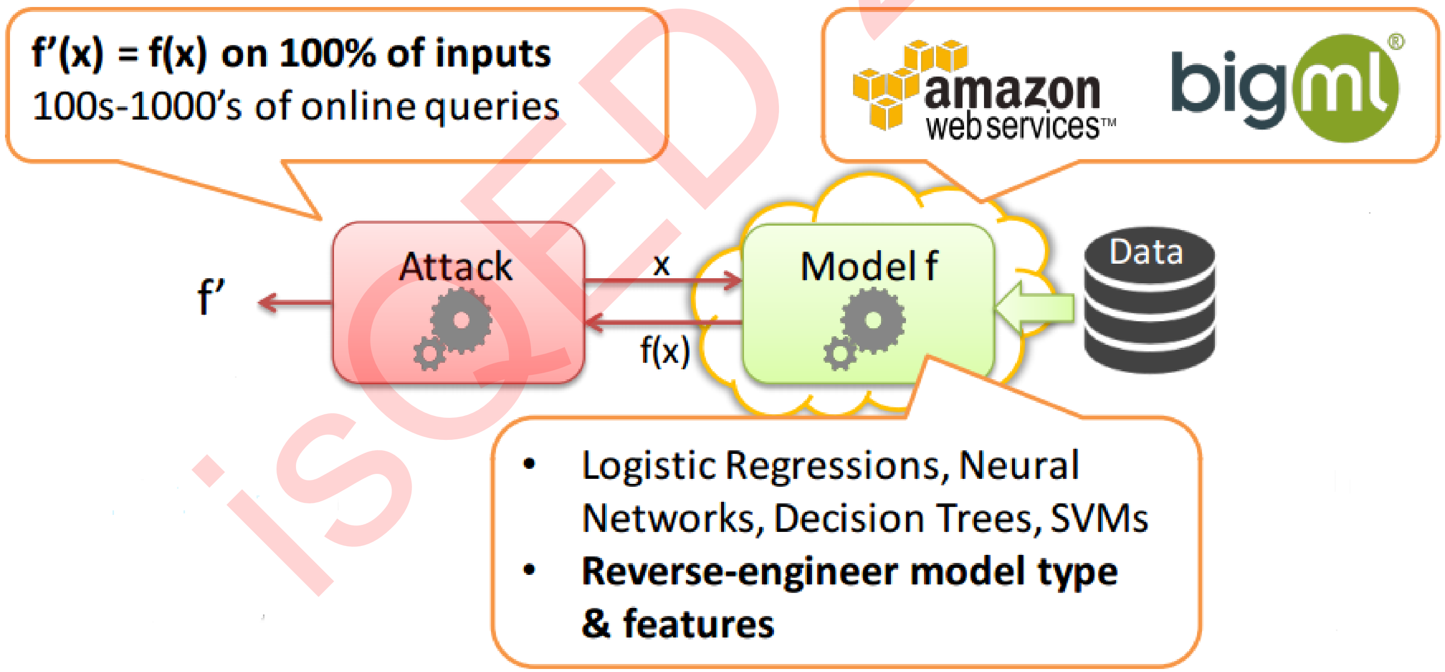
Confidentiality of Machine Learning Model

- Ensuring **confidentiality** of ML model is **critical**
  - Model **IP ownership** - **primary source of value** for company/ service
    - Cloud-based MLaaS models – highly lucrative for attackers
  - Model confidentiality also ensures training **data privacy**

- Attacks
  - **Model Extraction Attack**: Extract **model parameters** via querying the model. Generate equivalent or near-equivalent model.
  - **Model Inversion Attack**: Extract **private and sensitive inputs** by leveraging the outputs and ML model.

Model Extraction

- **Goal:** Adversarial client learns close approximation, *f'*, of *f* using as few queries as possible
  - Service provider prediction APIs themselves used in attack
    - APIs return extra information – **confidence scores**



* Tramer et.al., "Stealing Machine Learning Models via Prediction APIs.", 2016.

Extraction Countermeasures

- **Restrict information** returned
  - E.g. do not return confidence scores
  - **Rounding** – return approximations where possible
- **Strict** query **constraints**
  - E.g. disregard incomplete queries
- **Ensemble methods**
  - Prediction = aggregation of predictions from multiple models
  - Might still be susceptible to *model evasion* attacks
- Prediction API minimization is not easy
  - API should still be useable for legitimate applications

* Tramer et.al., “Stealing Machine Learning Models via Prediction APIs.”, 2016.

Model Inversion Attack

- ❖ **Optimization goal**: Find inputs that maximize returned confidence value to infer sensitive features or complete datapoints from a training dataset
  - o Exploits confidence values exposed by ML APIs

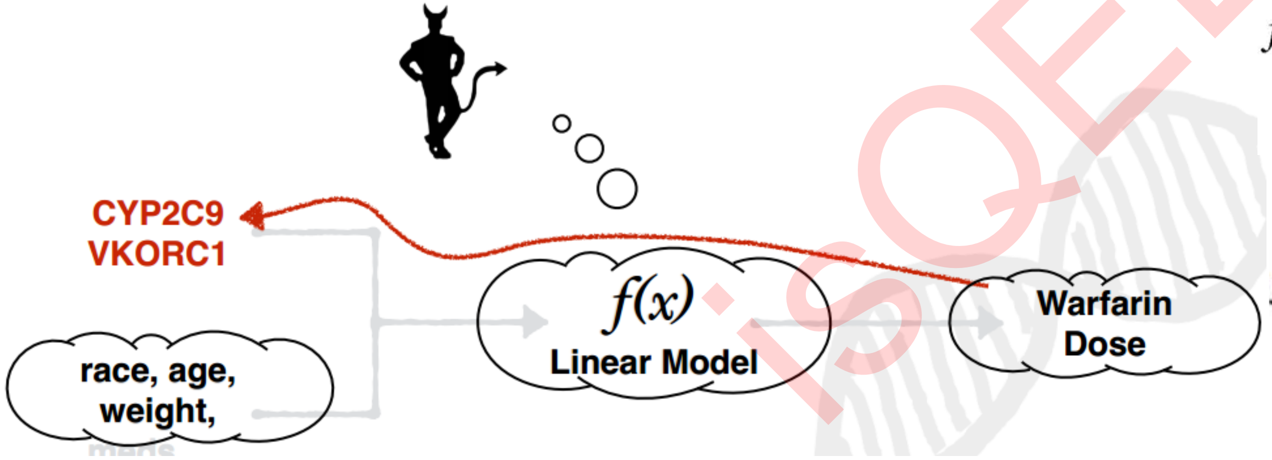
An image recovered using a new model inversion attack (left) and a training set image of the victim (right). The attacker is given only the person's name and access to a facial recognition system that returns a class confidence score.

* Fredrikson et.al., "Model Inversion Attacks that Exploit Confidence Information and Basic Countermeasures.", 2015

Privacy of the Training or Test Data

- Extracting patients' genetics from *pharmacogenetic dosing models*
  - **Queries** using *known information* – E.g. demographics, dosage
  - **Guess** unknown information and check model's response - assign weights
  - Return guesses that produce **highest confidence score**

age	height	weight	race	history	vkorc1	cyp2c9	dose
50-60	176.2	185.7	asian	cancer	A/G	*1/*3	42.0



$f(x)$

age	height	weight	race	history	vkorc1	cyp2c9	dose		
50-59	176.53	144.2	white				42.0	49.7	*p=0.23*
50-59	176.53	144.2	white				42.0	42.0	*p=0.75*
50-59	176.53	144.2	white				42.0	39.2	*p=0.01*

$f(x)$

age	height	weight	race	history	vkorc1	cyp2c9	dose		
50-59	176.53	144.2	white	Cancer	A/G	*1/*1	42.0	49.7	*p=0.23*
50-59	176.53	144.2	white	Heart	G/G	*1/*3	42.0	42.0	*p=0.75*
50-59	176.53	144.2	white	Diabetes	A/A	*2/*3	42.0	39.2	*p=0.01*

Fredrikson et.al., "Privacy in Pharmacogenetics: An End-to-End Case Study of Personalized Warfarin Dosing ", 2014.

Inversion Countermeasures

- Incorporate model inversion metrics to increase robustness
  - **Identify** sensitive features
  - Analyze **effective feature placement** in algorithm – E.g. sensitive features at top of a *decision tree* maintain accuracy while preventing *inversion* from performing better than guessing
  - **Approximate**/ **Degrade** confidence score output – E.g. decrease gradient magnitudes
    - Works against non-adapting attacker

- Ensuring privacy needs to be balanced against usability
  - **Privacy Budget**
- **Differential Privacy** mechanisms using added noise
  - Might prevent model inversion
  - Risk of compromising legitimate results in critical applications

Integrity

Introduction

- Ensuring **Integrity** of a Machine Learning model is difficult
  - Dependent on **quality** of *training*, *testing* datasets
    - Coverage of *corner cases*
    - Awareness of *adversarial examples*
  - **Model sophistication** – E.g. small model may produce incorrect outputs
  - **Lifetime management** of larger systems
    - Driverless cars will need constant updates
    - Degradation of input sensors, training data pollution
- Adversarial examples may be **Transferable** *
  - Example that fools Model A might fool Model B
  - Smaller model used to find examples quickly to target more sophisticated model

Papernot et. al., "Transferability in Machine Learning: from Phenomena to Black-Box Attacks using Adversarial Samples", 2016

Integrity Attacks

- Adversary can cause misclassifications of attacks to appear as normal (false positives/ negatives)
  - Attack on training phase: **Poisoning (Causative) Attack**: Attackers attempt to learn, influence, or corrupt the ML model itself
    - Compromising data collection
    - Subverting the learning process
    - Degrading performance of the system
    - Facilitating future evasion
  - Attack on testing phase: **Evasion (Exploratory) Attack**: Do not tamper with ML model, but instead cause it to *produce adversary selected outputs*.
    - Finding the blind spots and weaknesses of the ML system to evade it

Adversarial Detection of Malicious Crowdsourcing

- Malicious crowdsourcing, or ***crowdturfing*** used for tampering legitimate applications
  - **Real users** paid to promote malicious intentions
  - Product reviews, Political campaigns, Spam
- Adversarial machine learning attacks
  - Evasion Attack: workers evade classifiers
  - Poisoning Attack: crowdturfing admins tamper with training data

BBC

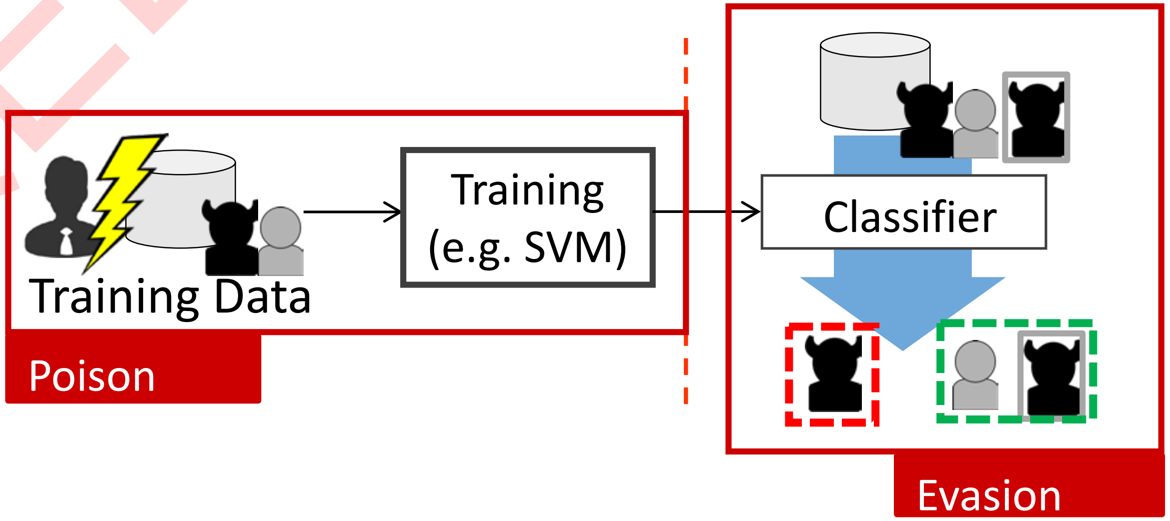
Vietnam admits deploying bloggers to support government

By Nga Pham
12 January 2013

THE VERGE

Samsung fined $340,000 for faking online comments

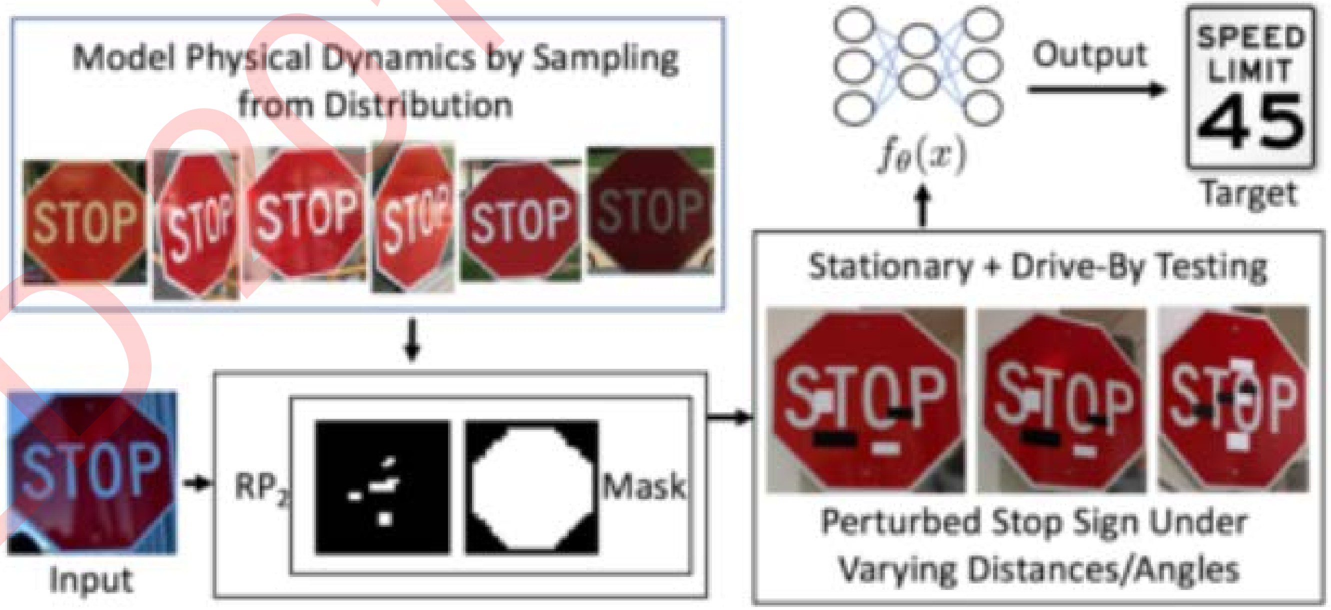
By Aaron Souppouris | Oct 24, 2013, 7:47am EDT



Wang et.al., "Man vs. Machine: Adversarial Detection of Malicious Crowdsourcing Workers ", 2014

Physical Perturbations

- Adversarial perturbations detrimentally affect Deep Neural Networks (DNNs)
  - Cause misclassification in critical applications
  - Requires some knowledge of DNN model
  - Perturbations can be robust against noise in system
- Defenses should not rely on physical sources of noise as protection
  - Incorporate adversarial examples
  - Restrict model information/ visibility
  - **DNN Distillation** – transfer knowledge from one DNN to another
  - **Gradient Masking**



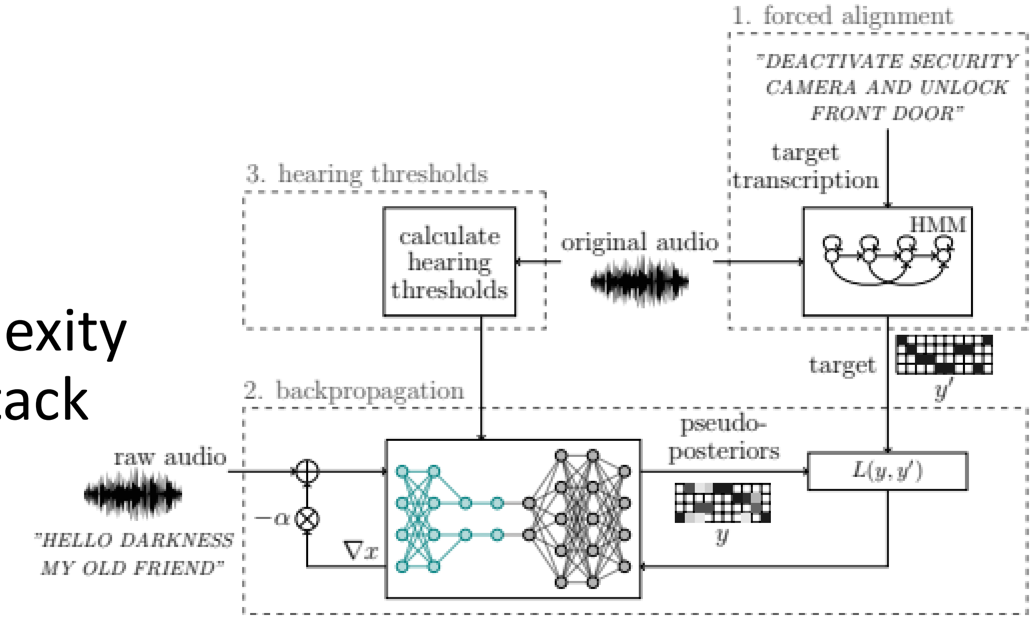
Eykholt et.al., "Robust Physical-World Attacks on Deep Learning Visual Classification", 2018.

Papernot et.al., "Distillation as a Defense to Adversarial Perturbations against Deep Neural Networks", 2015.

Adversarial Attacks Against ASR DNNs

- Automatic Speech Recognition (**ASR**) and Natural Language Understanding (**NLU**) increasingly popular – E.g. Amazon Alexa/ Echo
  - Complex model = **Large parameter space** for attacker to explore
- Attacker goals
  - Psychoacoustic hiding – perceived as noise by human
  - Identify and match legitimate voice features
    - Pitch, tone, fluency, volume, etc
  - Embed arbitrary audio input with a malicious voice command
  - *Temporal alignment* dependencies add complexity
  - Environment/ System *variability* can affect attack
  - Software tools like *Lyrebird* can prove useful

Lyrebird



Lea et.al., "Adversarial Attacks Against Automatic Speech Recognition Systems via Psychoacoustic Hiding", 2018

Defenses Against AML

- Evasion
  - Multiple classifier systems (B. Biggio et al., IJMLC 2010)
  - Learning with Invariances (SVMs)
  - Game Theory (SVMs)
- Poisoning
  - Data sanitization (B. Biggio et al., MCS, 2011)
  - Robust learning (PCA)
  - Randomization, information hiding, security by obscurity
- Randomizing collection of training data (timings / locations)
  - using difficult to reverse-engineer classifiers (e.g., MCSs)
  - denying access to the actual classifier or training data
  - randomizing classifier to give imperfect feedback to the attacker (B. Biggio et al., S+SSPR 2008)

Availability

Model/ Dataset Dissemination

- Model access can be in 3 forms
  - **Local** – Smartphone AI NPUs
  - **Cloud** – Amazon SageMaker, Microsoft Azure ML
  - **Hybrid** – Federated ML

- Training datasets difficult to generate
  - **Open datasets** – useful for small startups
    - Lack details, annotations
  - **Commercial datasets** – no incentive to share
    - Provides large advantage for provider

SageMaker

Azure ML

Source: Gboard - https://ai.googleblog.com/2017/04/federated-learning-collaborative.html

Attacker Goals

- Degrade learner's performance
  - Man-in-the-middle attack during Online Training
  - Generate false positive/negatives for valid inputs

- Delay output availability in time-critical applications
  - Driverless cars

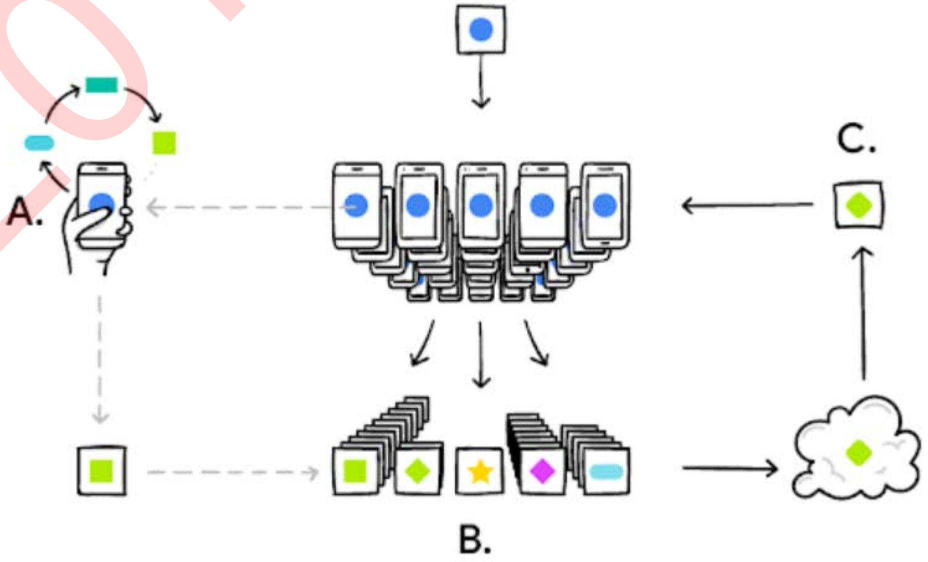
- DDoS attacks on Cloud-based ML models may affect millions of customers

- Access and timing control needed
  - Authentication of training sources
  - Default defensive response for delayed output

Federated ML

- Allows edge devices to update model
  - **No centralized data**
  - Training data stays **local**
  - **Averaging** to generate new shared model
    - *Secure Aggregation* needed
  - Issue of up-to-date access across all connected devices
    - Bandwidth, latency, scheduling
  - Cross-compatibility with different models for same application is difficult

- Still in development



Your phone personalizes the model locally, based on your usage (A). Many users' updates are aggregated (B) to form a consensus change (C) to the shared model, after which the procedure is repeated.

Source: https://ai.googleblog.com/2017/04/federated-learning-collaborative.html

Ensuring Future Robustness of Machine Learning Model

Future Research Areas

- Complexity of Machine Learning itself an issue
  - New attacks models constantly emerging – *timely detection* critical
  - Generation and incorporation of **Adversarial Examples**
  - **Data Privacy** is crucial to enhance ML security
    - *Differential Privacy* has tradeoffs
    - *Homomorphic Encryption* still nascent
- Security introduces overhead and can affect performance
  - **Optimizations** needed to ensure ML effiency
- Tools to increase robustness of Machine Learning need research
  - *Unlearning, re-learning*
  - *ML Testing*
  - *Sensitivity Analysis*

Unlearning and Re-learning

- Ability to **unlearn** is gaining importance
  - **Pollution** attacks or carelessness – *Mislabeling* and *Misclassification*
    - Large changing datasets difficult to maintain
    - Anomaly detection not enough
  - **EU GDPR** regulations – **Privacy**
  - **Completeness** and **Timeliness** are primary concerns *
  - **Statistical Query Learning*** and **Causal Unlearning**** proposed in literature
  - Suitable for **small deletions**
- **Re-learning** or **Online learning**
  - Faces similar issues to un-learning
  - Can be very slow
  - More suitable for large amounts of deletions or new information

* Yinzhi Cao, "Towards Making Systems Forget with Machine Unlearning", 2015

** Cao *et. al.*, "Efficient Repair of Polluted Machine Learning Systems via Causal Unlearning", 2018

ML Testing – Fuzz Testing

- Provide *invalid*, *unexpected* or *random* data to identify defects and vulnerabilities
  - Fuzz Testing works well with *structured inputs*
- Fuzzing can identify exploitable ML **implementation bugs** [1]
  - Valid inputs can compromise system
  - Points of attack
    - **Insufficient integrity checks** during *Feature Extraction*
    - **Overflow/Underflow**
    - **NaN, Loss of precision**
  - Vulnerabilities found in many open-source packages – OpenCV, Scikit-learn
- Fuzz Testing can aid security of general-purpose DNNs [2]
  - *Automation* and *parallelization* important – DNNs can be very big
  - *Input mutations* and coverage-criteria based *feedback* guidance specific to DNNs allow detection of **corner-cases**

[1] Stevens et.al, "Summoning Demons : The Pursuit of Exploitable Bugs in Machine Learning", 2017.
[2] Xie et.al,"DeepHunter: Hunting Deep Neural Network Defects via Coverage-Guided Fuzzing", 2018.

Sensitivity Analysis

- Study of how the uncertainty in the output of a system can be attributed to different sources of uncertainty in its inputs
  - ML feature extraction sensitivity analysis well-researched

- Detection of **biases** in training/test datasets is crucial *
  - Model accuracy dependent on datasets used – ***real-world*** performance can be different
    - Datasets can have **expiration dates**
    - **Privacy** issues can render datasets incomplete
  - Identify training datasets which **generalize** better
  - Study sensitivity of ML accuracy to change in datasets

* Sanders, Saxe, "Garbage In, Garbage Out - How Purportedly Great ML Models Can Be Screwed Up By Bad Data", 2017

Conclusion

- ML supply chain and revenue model is evolving
  - IP protection issue
- Protecting training data set and model IP is necessary for confidentiality
- Protection against evasion, poisoning attacks is necessary for integrity
- Real-time and robustness guarantees are necessary for availability

Thank you