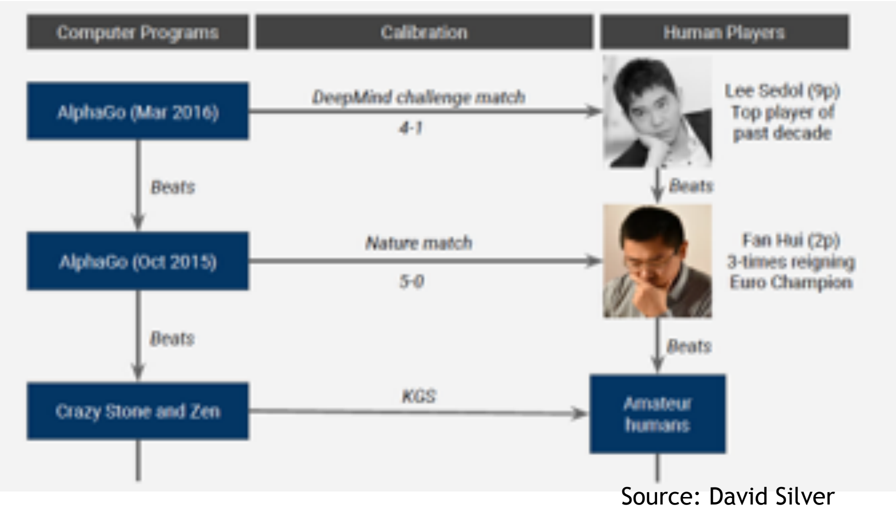
AI and Security: Lessons, Challenges & Future Directions

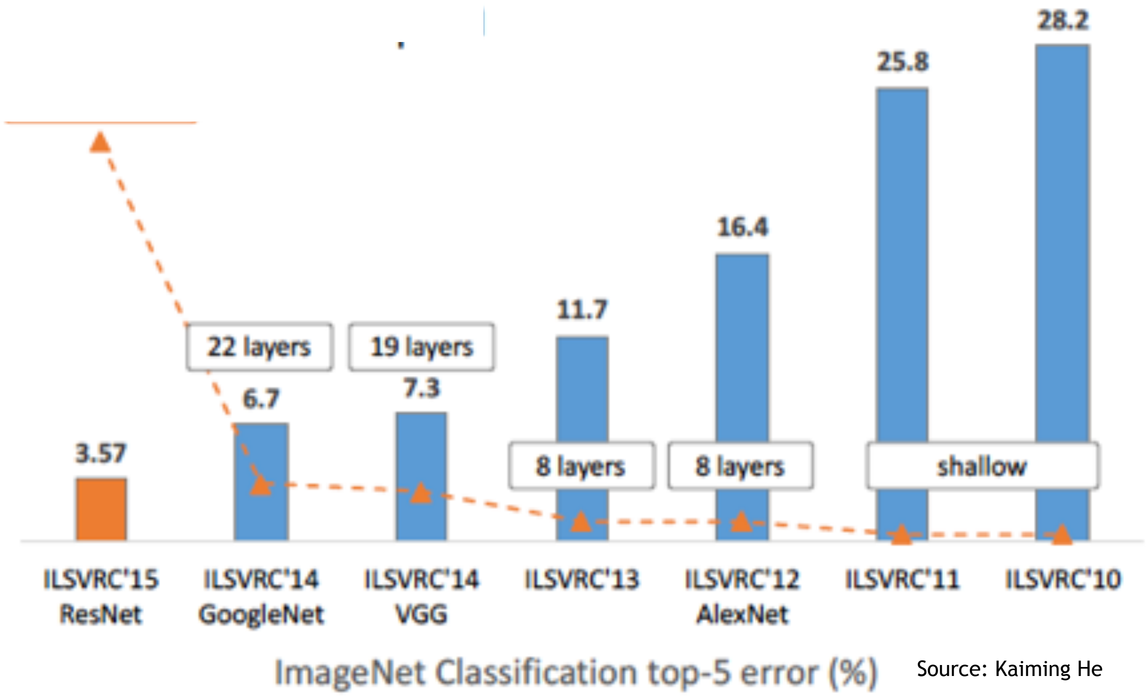
Dawn Song
UC Berkeley

AlphaGo: Winning over World Champion



Source: David Silver

Achieving Human-Level Performance on ImageNet Classification



Source: Kaiming He

Deep Learning Powering Everyday Products

pcmag.com

theverge.com

Attacks are increasing in scale & sophistication

Massive DDoS Caused by IoT Devices

source: Incpasula

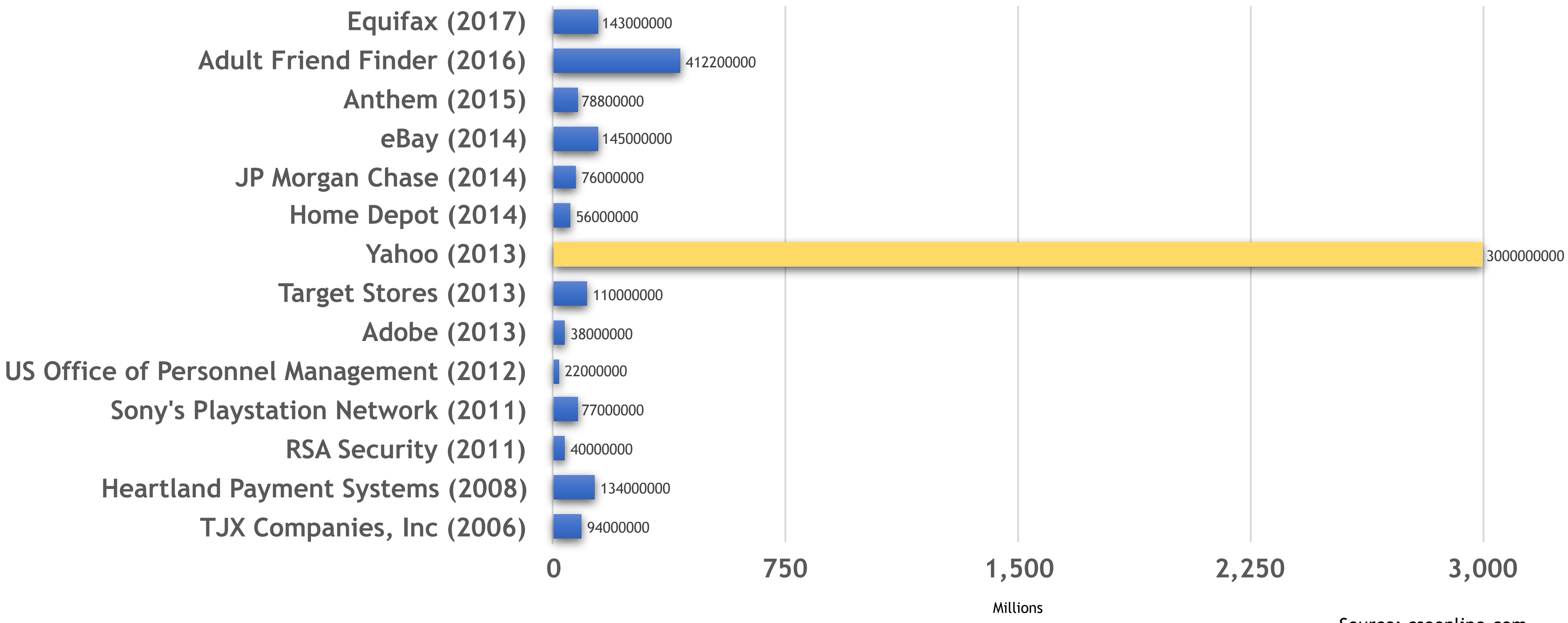
Geographical distribution of Mirai bots in recent DDoS attack

- Botnet of over 400,000 Mirai bots over 160 countries
    - Security cameras/webcams/baby monitors
    - Home routers
- One of the biggest DDoS attacks
    - Over 1Tbps combined attack traffic

WannaCry: One of the Largest Ransomware Breakout

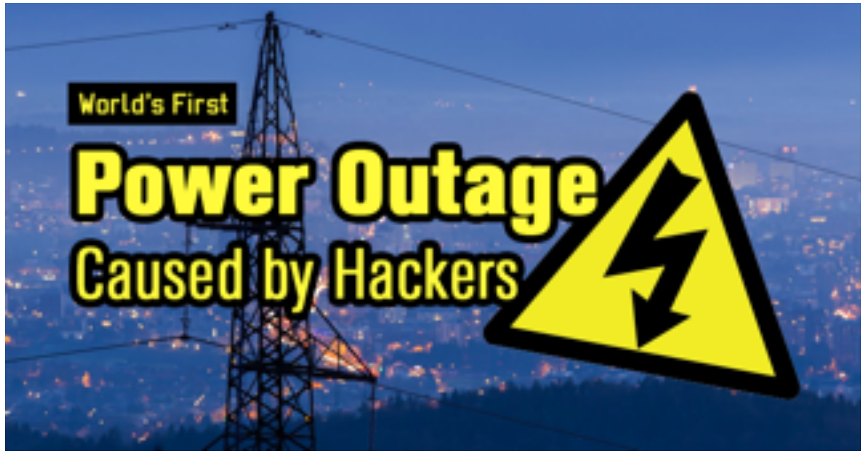
- Used EternalBlue, an exploit of Windows’ Server Message Block (SMB) protocol.
- Infected over 200,000 machines across 150 countries in a few days
- Ask for bitcoin payment to unlock encrypted files

Biggest Data Breaches Of the 21st Century



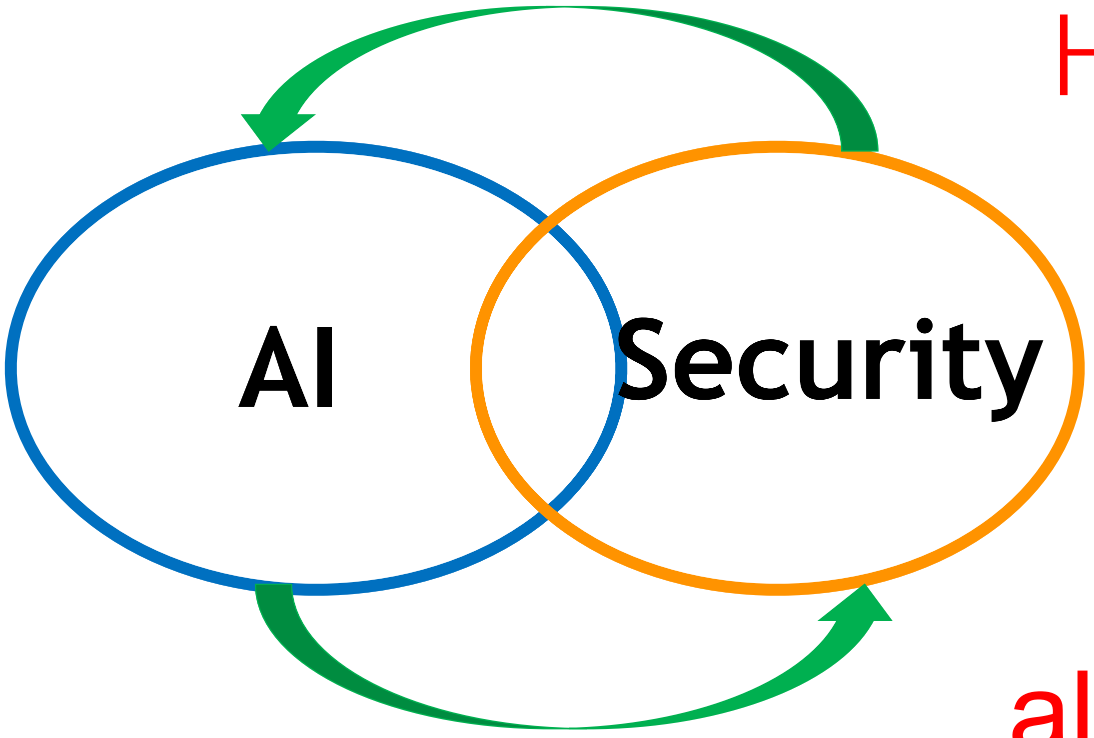
Source: csoonline.com

Attacks Entering New Landscape



Ukrain power outage by cyber attack impacted over 250,000 customers

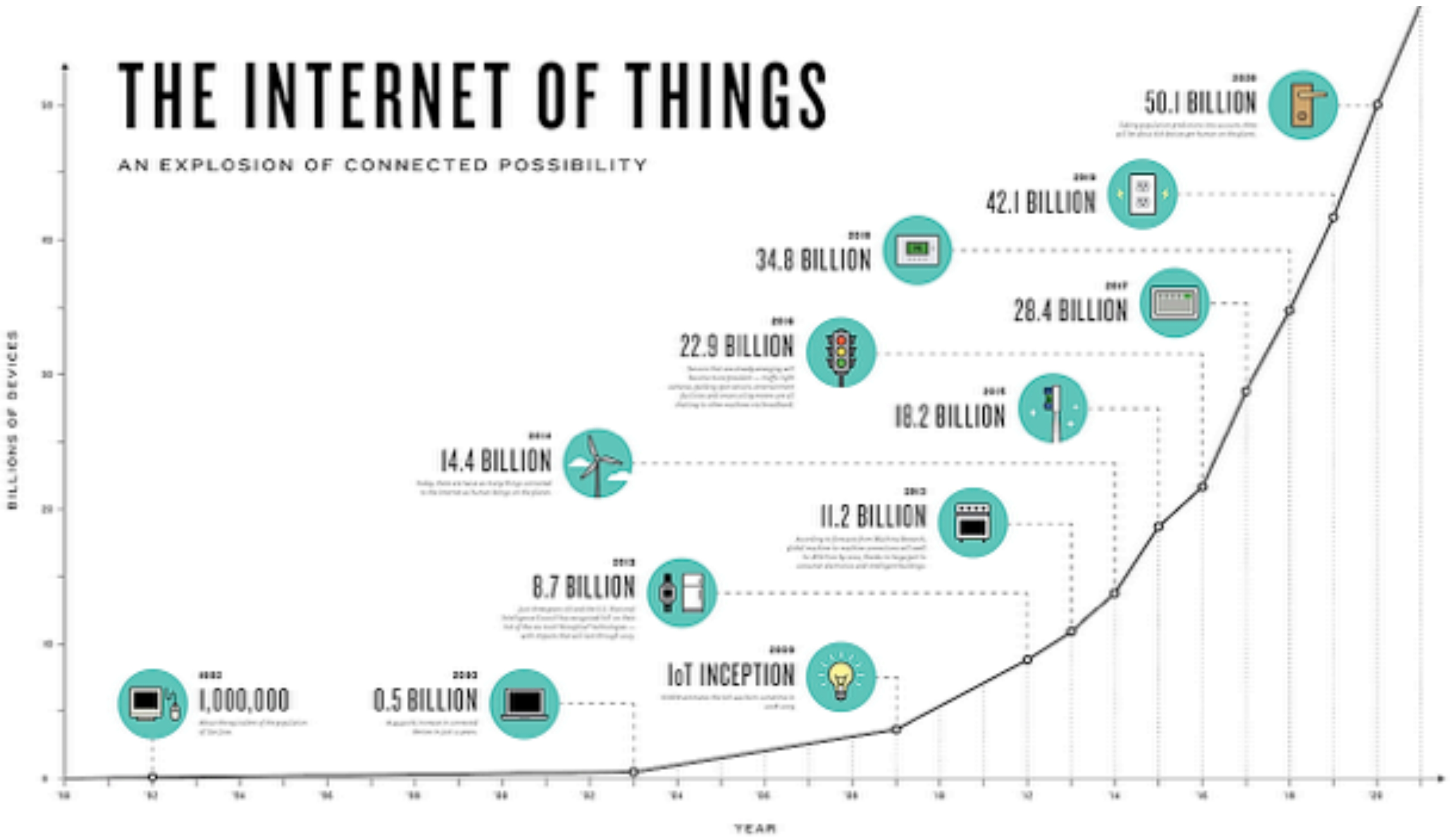
Millions of dollars lost in targeted attacks in SWIFT banking system



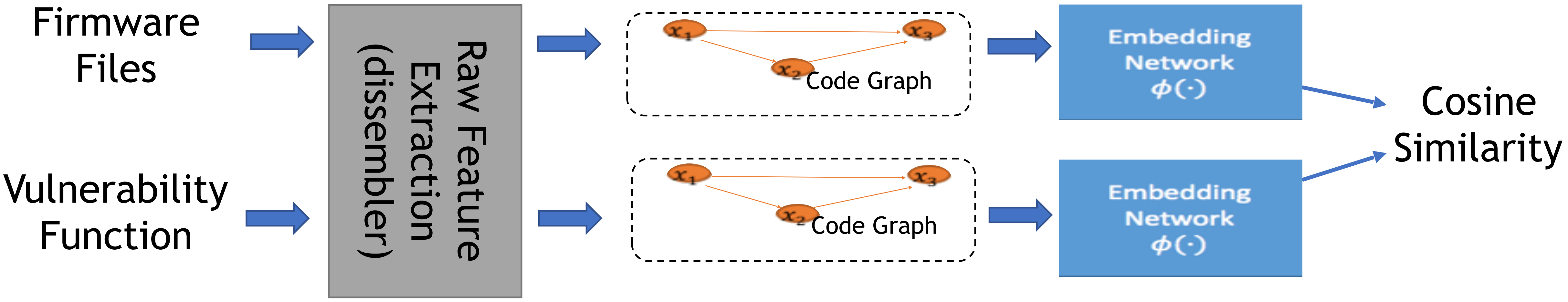
How will (in)security impact the deployment of AI?

How will the rise of AI alter the security landscape?

IoT devices are plagued with vulnerabilities from third-party code



Deep learning for vulnerability detection in IoT Devices



Neural Network-based Graph Embedding for Cross-Platform Binary Code Search
[XLFSSY, ACM Computer and Communication Symposium 2017]

Deep learning for vulnerability detection in IoT Devices

Training time:
  Previous work: > 1 week
  Our approach: < 30 mins

Serving time (per function):
  Previous work: a few mins
  Our work: a few milliseconds
    10,000 times faster

Function Name	Vendor	Firmware	Binary File	Similarity
ssl3_get_new_session_ticket	D-Link	DAP-1562_FIRMWARE_1.00	wpa_supplicant.acfgs	0.962374508
port_check_v6	D-Link	DES-1210-28_REVB_FIRMWARE_3.12.015	in.ftpd.acfgs	0.955408692
sub_42EE7C	TP-Link	TD-W8970B_V1_140624	racoon.acfgs	0.954742193
sub_42EE7C	TP-Link	TD-W8970_V1_130828	racoon.acfgs	0.954742193
prsa_parse_file	TP-Link	Archer_D5_V1_140804	racoon.acfgs	0.949814439
sub_4128BC	TP-Link	TD-W8970B_V1_140624	racoon.acfgs	0.949583828
sub_4128BC	TP-Link	TD-W8970_V1_130828	racoon.acfgs	0.949583828
ssl3_get_new_session_ticket	DD-wrt	dd-wrt.v24-23838_NEWD-2_K3.x_mega-WNR3500v2_VC	openvpn.acfgs	0.94668287
ucSetUsbipServer	TP-Link	WDR4900_V2_130115	httpd.acfgs	0.946312308
ssl3_get_new_session_ticket	Netgear	tomato-Cisco-M10v2-NVRAM32K-1.28.RT-N5x-MIPSR2-110-PL-Mini	libssl.so.1.0.0.acfgs	0.945913044
ssl3_get_new_session_ticket	Tomato_by_Shibby	tomato-K26-1.28.RT-MIPSR1-109-Mini	libssl.so.1.0.0.acfgs	0.945913044
ssl3_get_new_session_ticket	Tomato_by_Shibby	tomato-K26USB-1.28.RT-N5x-MIPSR2-110-VPN	libssl.so.1.0.0.acfgs	0.945912984
ssl3_get_new_session_ticket	Tomato_by_Shibby	tomato-E4200USB-NVRAM60K-1.28.RT-MIPSR2-110-PL-BT	libssl.so.1.0.0.acfgs	0.945912984
ssl3_get_new_session_ticket	Tomato_by_Shibby	tomato-E3000USB-NVRAM60K-1.28.RT-MIPSR2-110-BT-VPN	libssl.so.1.0.0.acfgs	0.945912984
ssl3_get_new_session_ticket	Tomato_by_Shibby	tomato-K26USB-1.28.RT-MIPSR1-109-AIO	libssl.so.1.0.0.acfgs	0.945912984
ssl3_get_new_session_ticket	Tomato_by_Shibby	tomato-Netgear-3500Lv2-K26USB-1.28.RT-N5x--109-AIO	libssl.so.1.0.0.acfgs	0.945912984
ssl3_get_new_session_ticket	Tomato_by_Shibby	tomato-E4200USB-NVRAM60K-1.28.RT-MIPSR2-109-AIO	libssl.so.1.0.0.acfgs	0.945912984
ssl3_get_new_session_ticket	Tomato_by_Shibby	tomato-E1550USB-NVRAM60K-1.28.RT-N5x-MIPSR2-110-Nocat-VPN	libssl.so.1.0.0.acfgs	0.945912984
ssl3_get_new_session_ticket	Tomato_by_Shibby	tomato-K26USB-1.28.RT-N5x-MIPSR2-115-PL-L600N	libssl.so.1.0.0.acfgs	0.945912984
ssl3_get_new_session_ticket	Tomato_by_Shibby	tomato-E1550USB-NVRAM60K-1.28.RT-N5x-MIPSR2-110-BT-VPN	libssl.so.1.0.0.acfgs	0.945912984
ssl3_get_new_session_ticket	Tomato_by_Shibby	tomato-E3000USB-NVRAM60K-1.28.RT-MIPSR2-108-PL-VPN	libssl.so.1.0.0.acfgs	0.945912984
ssl3_get_new_session_ticket	Tomato_by_Shibby	tomato-E1550USB-NVRAM60K-1.28.RT-N5x-MIPSR2-110-Mega-VPN	libssl.so.1.0.0.acfgs	0.945912984
ssl3_get_new_session_ticket	Tomato_by_Shibby	tomato-E1200v2-NVRAM64K-1.28.RT-N5x-MIPSR2-108-PL-Max	libssl.so.1.0.0.acfgs	0.945912984
ssl3_get_new_session_ticket	Tomato_by_Shibby	tomato-K26USB-1.28.RT-MIPSR1-109-Mega-VPN	libssl.so.1.0.0.acfgs	0.945912984
ssl3_get_new_session_ticket	Tomato_by_Shibby	tomato-E3000USB-NVRAM60K-1.28.RT-MIPSR2-109-Big-VPN	libssl.so.1.0.0.acfgs	0.945912984
ssl3_get_new_session_ticket	Tomato_by_Shibby	tomato-E4200USB-NVRAM60K-1.28.RT-MIPSR2-108-PL-Nocat-VPN	libssl.so.1.0.0.acfgs	0.945912984
ssl3_get_new_session_ticket	Tomato_by_Shibby	tomato-Netgear-3500Lv2-K26USB-1.28.RT-N5x--110-ND-AIO	libssl.so.1.0.0.acfgs	0.945912984
ssl3_get_new_session_ticket	Tomato_by_Shibby	tomato-E4200USB-NVRAM60K-1.28.RT-MIPSR2-109-Nocat-VPN	libssl.so.1.0.0.acfgs	0.945912984

Identified vulnerabilities among top 50:
  Previous work: 10/50
  Our approach: 42/50

AI Enables Stronger Security Capabilities

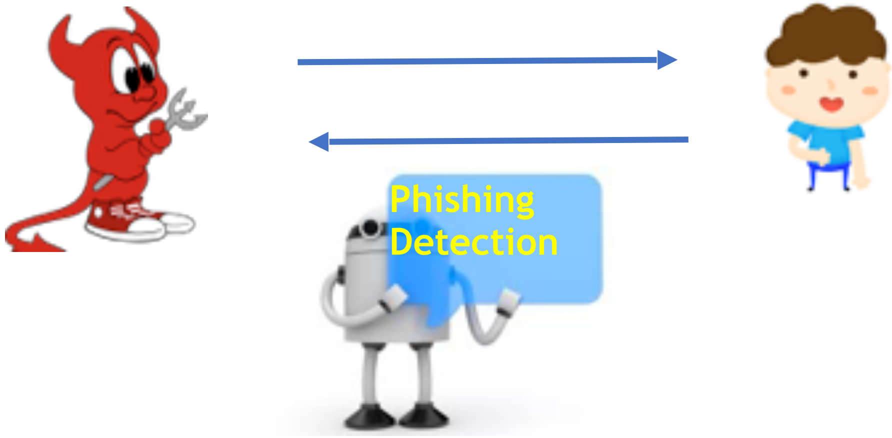
- Automatic vulnerability detection & patching
- Automatic agents for attack detection, analysis, & defense

One fundamental weakness of cyber systems is humans

80+% of penetrations and hacks start with a social engineering attack
70+% of nation state attacks [FBI, 2011/Verizon 2014]

AI Enables Chatbot for Phishing Detection

Chatbot for booking flights,
finding restaurants



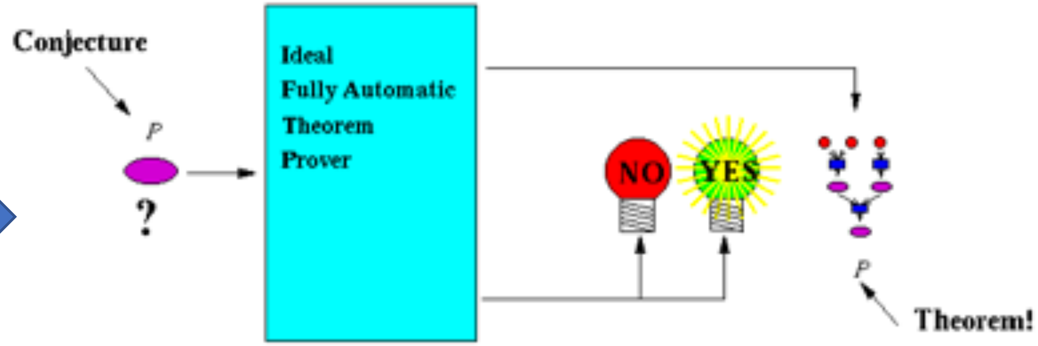
Chatbot for social engineering attack
detection & defense

AI Enables Stronger Security Capabilities

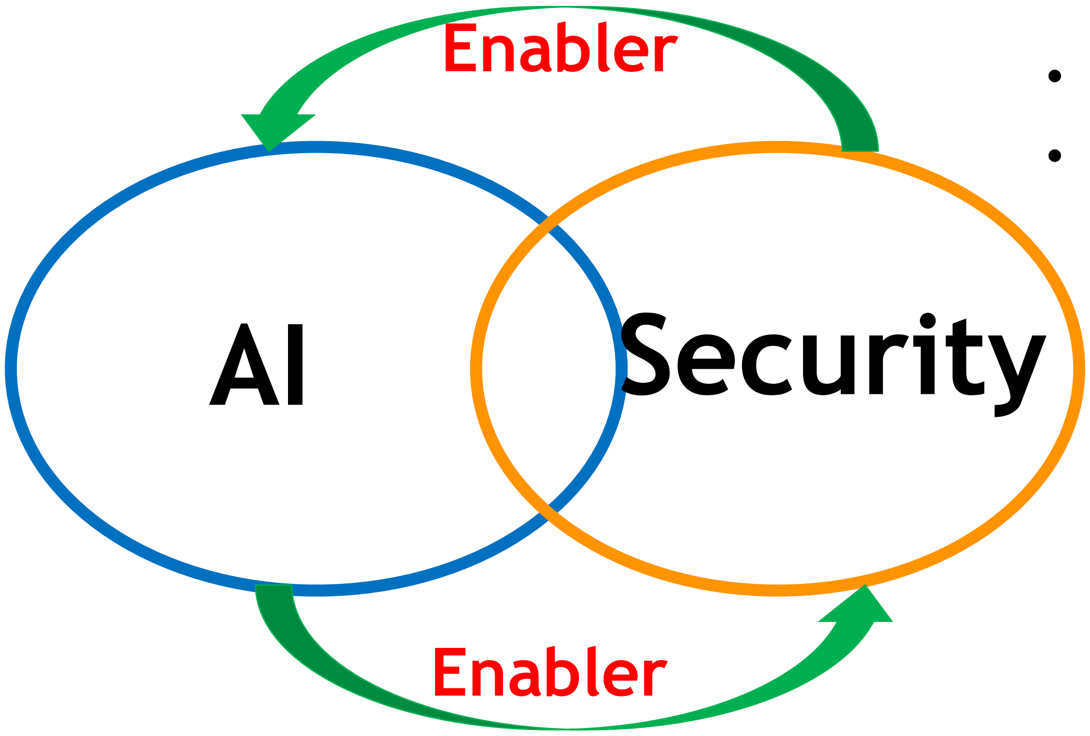
- Automatic vulnerability detection & patching
- Automatic agents for attack detection, analysis, & defense
- Automatic verification of software security

AI Agents to Prove Theorems & Verify Programs

Deep Reinforcement Learning
Agent Learning to Play Go



Automatic Theorem Proving
for Program Verification



- AI enables new security capabilities
- Security enables better AI

Integrity: produces intended/correct results (adversarial machine learning)

Confidentiality/Privacy: does not leak users' sensitive data (secure, privacy-preserving machine learning)

Preventing misuse of AI

AI and Security: AI in the presence of attacker

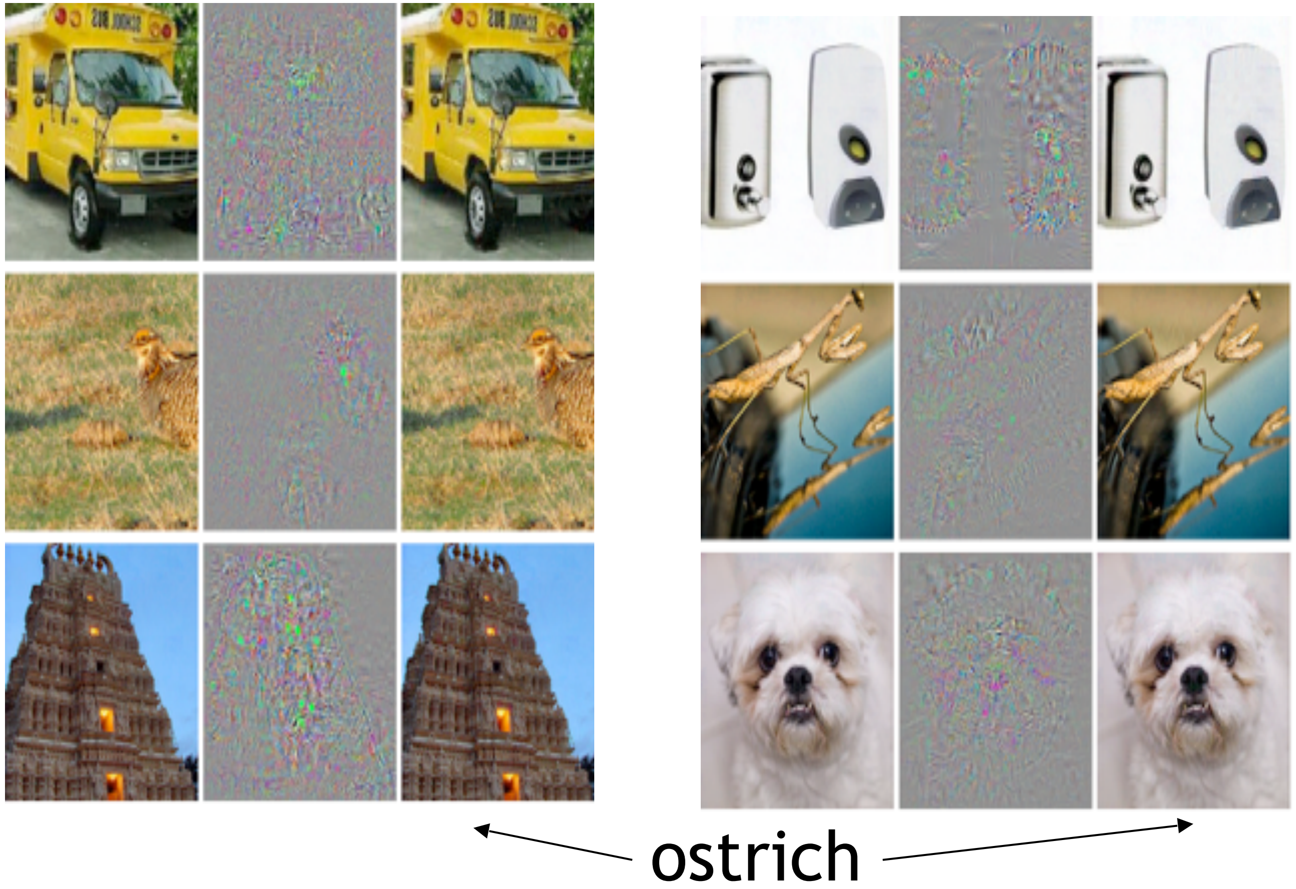
Important to consider the presence of attacker

- History has shown attacker always follows footsteps of new technology development (or sometimes even leads it)
- The stake is even higher with AI
  - As AI controls more and more systems, attacker will have higher & higher incentives
  - As AI becomes more and more capable, the consequence of misuse by attacker will become more and more severe

AI and Security: AI in the presence of attacker

- Attack AI
  - Cause the learning system to not produce intended/correct results
  - Cause learning system to produce targeted outcome designed by attacker
  - Learn sensitive information about individuals
  - Need security in learning systems
- Misuse AI
  - Misuse AI to attack other systems
    - Find vulnerabilities in other systems; Devise attacks
  - Need security in other systems

Deep Learning Systems Are Easily Fooled



$$\frac{\partial \text{output}}{\partial \text{pixels}}$$

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., & Fergus, R. Intriguing properties of neural networks. ICLR 2014.


STOP
SPEED
LIMIT
50

STOP Signs in Berkeley

Adversarial Examples in Physical World

Adversarial examples in physical world remain effective under different viewing distances, angles, other conditions

Subtle Poster | Subtle Poster | Camo Graffiti | Camo Art | Camo Art

Lab Test Summary
(Stationary)

Target Class: **Speed Limit 45**

Misclassify

Evtimov, Ivan, Kevin Eykholt, Earlence Fernandes, Tadayoshi Kohno, Bo Li, Atul Prakash, Amir Rahmati, and Dawn Song. "Robust Physical-World Attacks on Machine Learning Models." *arXiv preprint arXiv:1707.08945* (2017).

Adversarial examples in physical world
&
remain effective under
different viewing distances, angles, other conditions

Drive-by Test

Adversarial Examples Are Prevalent in Deep Learning Systems

Adversarial Examples Prevalent in Deep Learning Systems

- Most existing work on adversarial examples:
  - Image classification task
  - Target model is known

- Our investigation on adversarial examples:

Generative Models

Deep Reinforcement Learning

VisualQA/ Image-to-code

Other tasks and model classes

Blackbox Attacks

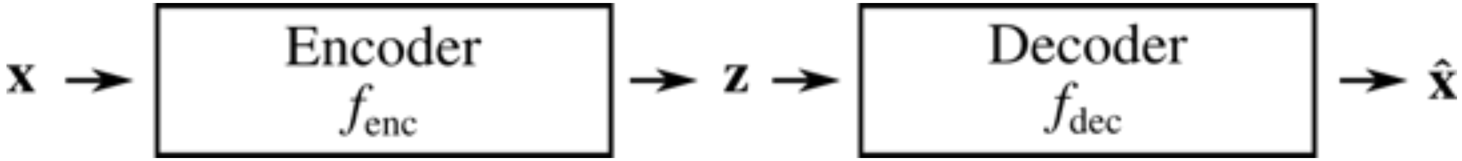
Weaker Threat Models (Target model is unknown)

New Attack Methods

Provide more diversity of attacks

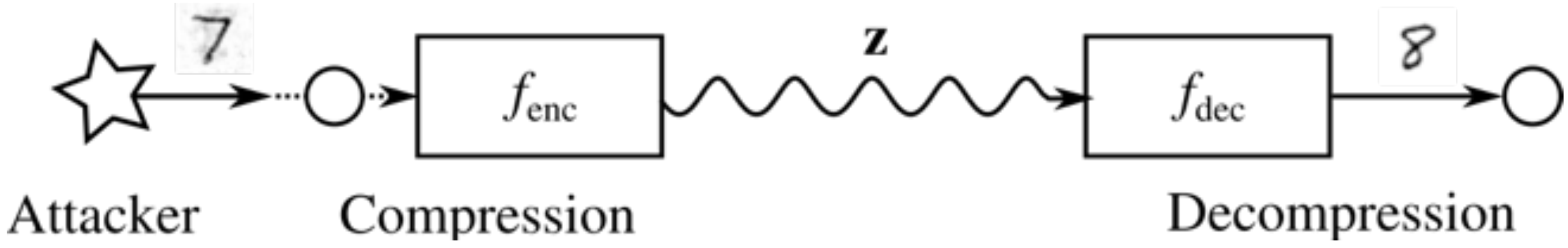
Generative models

- VAE-like models (VAE, VAE-GAN) use an intermediate latent representation
- An **encoder**: maps a high-dimensional input into lower-dimensional latent representation **z**.
- A **decoder:** maps the latent representation back to a high-dimensional reconstruction.



Adversarial Examples in Generative Models

- An example attack scenario:
  - Generative model used as a compression scheme



- Attacker’s goal: for the decompressor to reconstruct a different image from the one that the compressor sees.

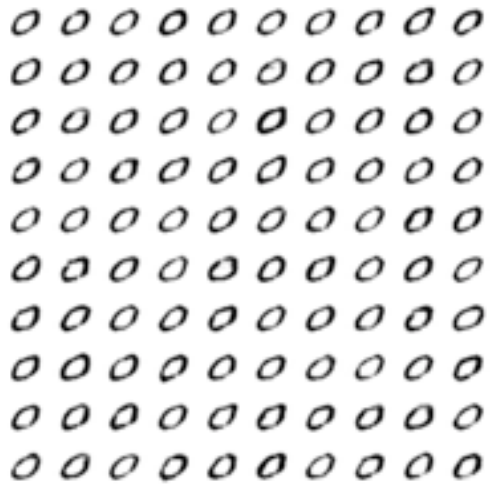
Adversarial Examples for VAE-GAN in MNIST

Original images

Reconstruction of original images

Target Image

Adversarial examples



Reconstruction of adversarial examples

Jernej Kos, Ian Fischer, Dawn Song: Adversarial Examples for Generative Models

Adversarial Examples for VAE-GAN in SVHN

Original images

Reconstruction of original images

Target Image

Adversarial examples

Reconstruction of adversarial examples

Jernej Kos, Ian Fischer, Dawn Song: Adversarial Examples for Generative Models

Adversarial Examples for VAE-GAN in SVHN

Original images

Reconstruction of original images

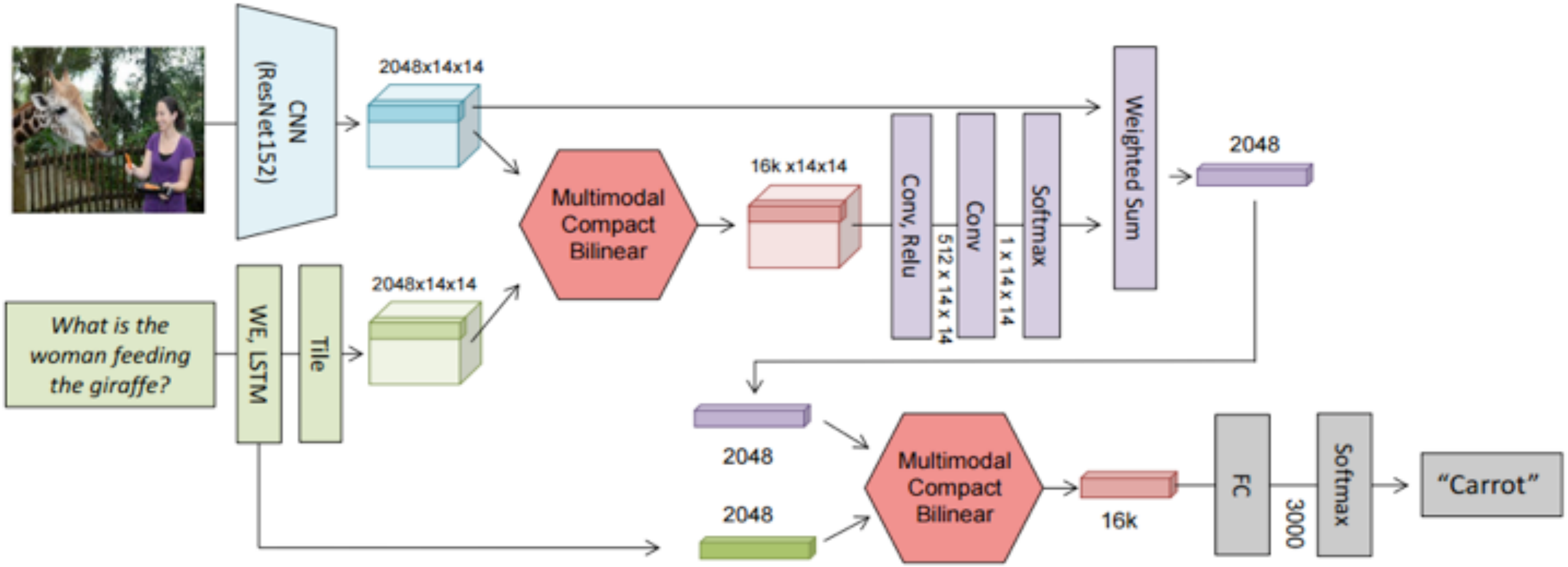
Target Image

Adversarial examples

Reconstruction of adversarial examples

Jernej Kos, Ian Fischer, Dawn Song: Adversarial Examples for Generative Models

Visual Question & Answer (VQA)



Multimodal Compact Bilinear Pooling for Visual Question Answering and Visual Grounding, Fukui et al., https://arxiv.org/abs/1606.01847

Q: Where is the plane?

Benign image

Mode

Answer:

Runway

Fooling VQA

Target: Sky

Adversarial example

Mode

Sky

Q: How many cats are there?

Benign image

Mode

Answer:

1

Fooling VQA

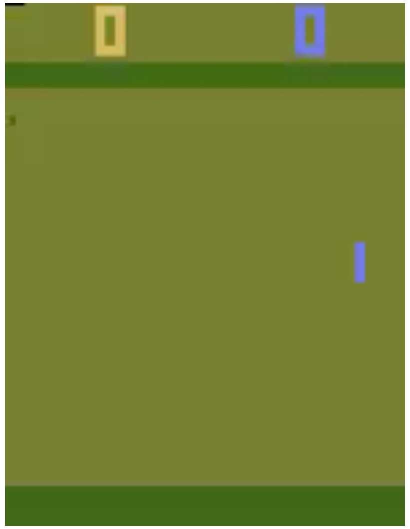
Target: 2

Adversarial example

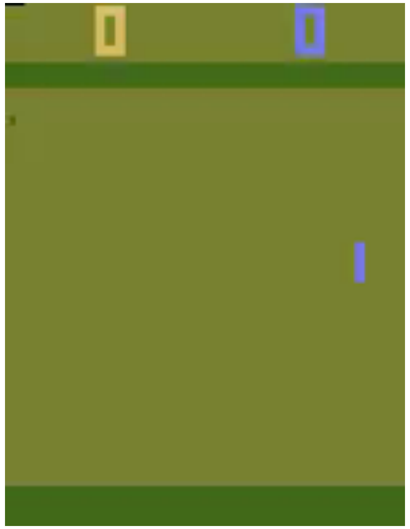
Mode

2

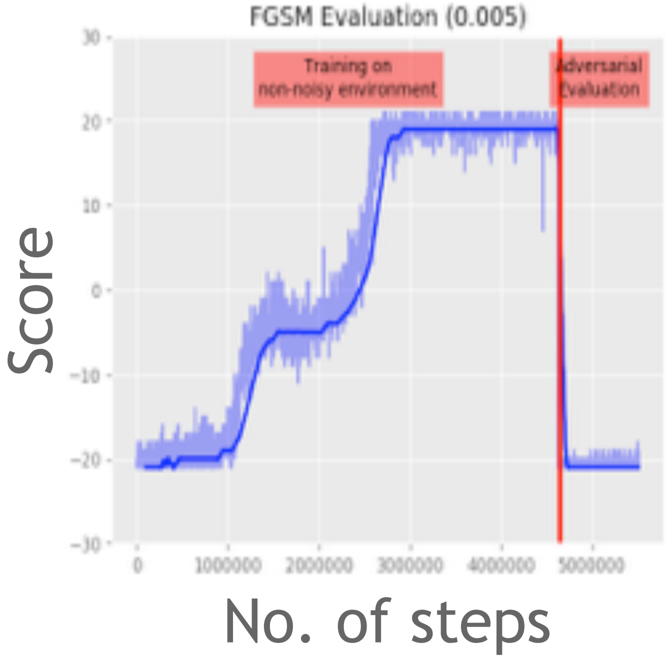
Adversarial Examples Fooling Deep Reinforcement Learning Agents



Original Frames



Original Frames with Adversarial Perturbation



Jernej Kos and Dawn Song: Delving into adversarial attacks on deep policies [ICLR Workshop 2017].

A General Framework for Black-box attacks

- Zero-Query Attack (Previous methods)
  - Random perturbation
  - Difference of means
  - Transferability-based attack
    - Practical Black-Box Attacks against Machine Learning [Papernot et al. 2016]
    - Ensemble transferability-based attack [Yanpei Liu, Xinyun Chen, Chang Liu, Dawn Song: Delving into Transferable Adversarial Examples and Black-box Attacks, ICLR 2017]
- Query Based Attack (new method)
  - Finite difference gradient estimation
  - Query reduced gradient estimation
  - Results: similar effectiveness to whitebox attack
  - A general active query game model

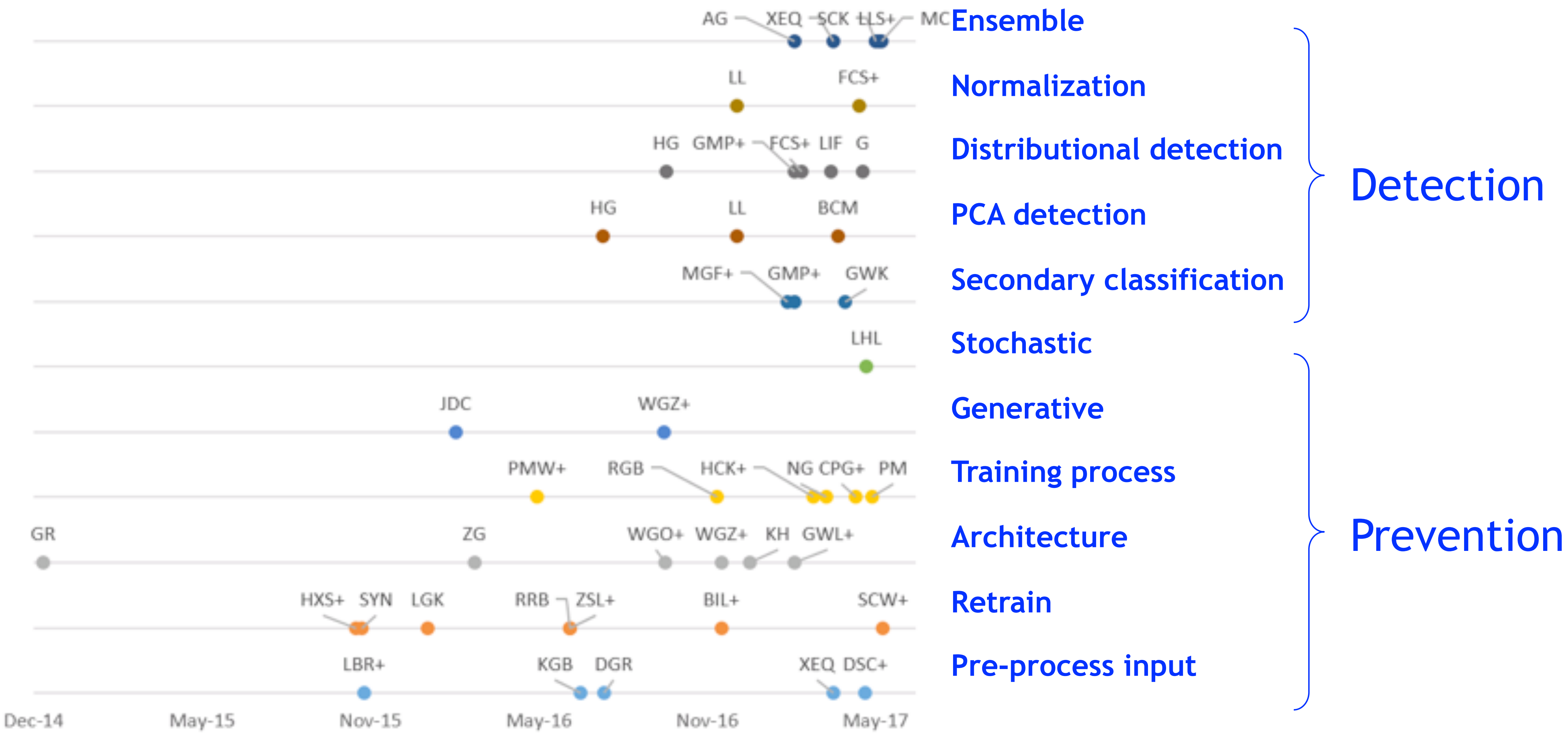
Black-box Attack on Clarifai

Original image, classified as "drug" with a confidence of 0.99

Adversarial example, classified as "safe" with a confidence of 0.96

The Gradient-Estimation black-box attack on Clarifai's Content Moderation Model

Numerous Defenses Proposed



No Sufficient Defense Today

- Strong, adaptive attacker can easily evade today's defenses

- Ensemble of weak defenses does not (by default) lead to strong defense
  - Warren He, James Wei, Xinyun Chen, Nicholas Carlini, Dawn Song [WOOT 2017]

- Adversarial Examples Are Not Easily Detected: Bypassing Ten Detection Methods
  - Nicholas Carlini and David Wagner [AISec 2017]

Adversarial Machine Learning

- Adversarial machine learning:
  - Learning in the presence of adversaries

- Inference time: adversarial example fools learning system
  - Evasion attacks
    - Evade malware detection; fraud detection

- Training time:
  - Attacker poisons training dataset (e.g., poison labels) to fool learning system to learn wrong model
    - Poisoning attacks: e.g., Microsoft's Tay twitter chatbot
  - Attacker selectively shows learner training data points (even with correct labels) to fool learning system to learn wrong model
  - Data poisoning is particularly challenging with crowd-sourcing & insider attack
  - Difficult to detect when the model has been poisoned

- Adversarial machine learning particularly important for security critical system

Security will be one of the biggest challenges in Deploying AI

Security of Learning Systems

- Software level
- Learning level
- Distributed level

Challenges for Security at Software Level

- No software vulnerabilities (e.g., buffer overflows & access control issues)
    - Attacker can take control over learning systems through exploiting software vulnerabilities

Challenges for Security at Software Level

- No software vulnerabilities (e.g., buffer overflows & access control issues)
- Existing software security/formal verification techniques apply

Reactive Defense

Proactive Defense: Bug Finding

Proactive Defense: Secure by Construction

Automatic worm detection & signature/patch generation

Automatic malware detection & analysis

Progression of different approaches to software security over last 20 years

Security of Learning Systems

- Software level
- Learning level
- Distributed level

Challenges for Security at Learning Level

- Evaluate system under adversarial events, not just normal events

Regression Testing vs. Security Testing in Traditional Software System

	Regression Testing	**Security Testing**
Operation	Run program on **normal** inputs	Run program on **abnormal/ adversarial** inputs
Goal	Prevent **normal** users from encountering errors	Prevent **attackers** from finding **exploitable** errors

Regression Testing vs. Security Testing in Learning System

	Regression Testing	**Security Testing**
Training	Train on noisy training data: Estimate resiliency against noisy training inputs	Train on poisoned training data: Estimate resiliency against poisoned training inputs
Testing	Test on **normal** inputs: Estimate generalization error	Test on **abnormal/ adversarial** inputs: Estimate resiliency against adversarial inputs

Challenges for Security at Learning Level

- Evaluate system under adversarial events, not just normal events
  - Regression testing vs. security testing
- Reason about complex, non-symbolic programs

Decades of Work on Reasoning about Symbolic Programs

- Symbolic programs:
  - E.g., OS, File system, Compiler, web application, mobile application
  - Semantics defined by logic
  - Decades of techniques & tools developed for logic/symbolic reasoning
    - Theorem provers, SMT solvers
    - Abstract interpretation

Era of Formally Verified Systems

Verified: Micro-kernel, OS, File system, Compiler, Security protocols, Distributed systems

seL4
Security. Performance. Proof.

IronClad/IronFleet

FSCQ

CertiKOS

miTLS/Everest

EasyCrypt

CompCert

Powerful Formal Verification Tools + Dedicated Teams

Coq

Why3

Z3

No Sufficient Tools to Reason about Non-Symbolic Programs

- Symbolic programs:
  - Semantics defined by logic
  - Decades of techniques & tools developed for logic/symbolic reasoning
    - Theorem provers, SMT solvers
    - Abstract interpretation

- Non-symbolic programs:
  - No precisely specified properties & goals
  - No good understanding of how learning system works
  - Traditional symbolic reasoning techniques do not apply

Challenges for Security at Learning Level

- Evaluate system under adversarial events, not just normal events
  - Regression testing vs. security testing
- Reason about complex, non-symbolic programs
- Design new architectures & approaches with stronger generalization & security guarantees

Neural Program Synthesis

Can we teach computers to write code?

Example Applications:

- End-user programming
- Performance optimization of code
- Virtual assistant

“Software is eating the world” --- az16

Program synthesis can automate this & democratize idea realization

Neural Program Synthesis

Trainin g data	123 234	452 345	612 367	**Input**
	357	797	979	**Output**

Neural Program Synthesis

Training data

123 234	452 345	612 367	**Input**
357	797	979	**Output**

Neural Program Architecture

50
70

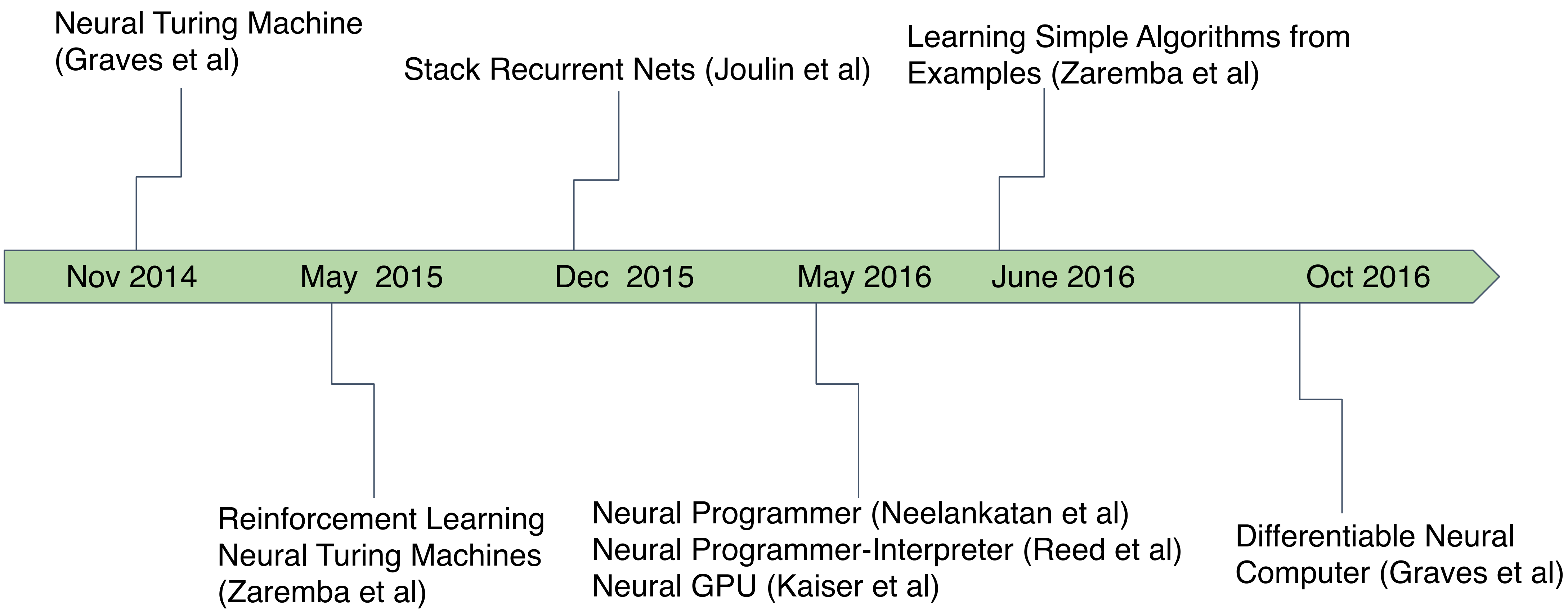
Test input

Learned neural program

120

Test output

Neural Program Architectures



Neural Program Synthesis Tasks: Copy, Grade-school addition, Sorting, Shortest Path

Challenge 1: Generalization

Training data

123 234	452 345	$\overbrace{612}^{\text{length = 3}}$ 367	**Input**
357	797	979	**Output**

Neural Program Architecture

$\overbrace{34216}^{\text{length = 5}}$
24320

Test input

Learned neural program

54 1

Test output

Challenge 2: No Proof of Generalization

Trainin g data

123	452	length = 3 612
234	345	367
357	797	979

Input

Output

Neural Program Architecture

length = 5
34216
24320

Test input

Learned neural program

58536

Test output

Our Approach: Introduce Recursion

Learn recursive neural programs

Jonathon Cai, Richard Shin, Dawn Song: Making Neural Programming Architectures Generalize via Recursion
[ICLR 2017, **Best Paper Award**]

Recursion

- Fundamental concept in Computer Science and Math
- Solve whole problem by reducing it to smaller subproblems (*reduction rules*)
- *Base cases* (smallest subproblems) are easier to reason about

Quicksort

Our Approach: Making Neural Programming Architectures Generalize via Recursion

- **Proof of Generalization**:
  - Recursion enables provable guarantees about neural programs
  - Prove perfect generalization of a learned recursive program via a verification procedure
    - Explicitly testing on all possible base cases and reduction rules (Verification set)

- Learn & generalize faster as well
  - Trained on same data, non-recursive programs do not generalize well

Accuracy on Random Inputs for Quicksort

Length of Array	Non-Recursive	Recursive
3	100%	100%
5	100%	100%
7	100%	100%
11	73.3%	100%
15	60%	100%
20	30%	100%
22	20%	100%
25	3.33%	100%
30	3.33%	100%
70	0%	100%

Jonathon Cai, Richard Shin, Dawn Song: Making Neural Programming Architectures Generalize via Recursion [ICLR 2017, **Best Paper Award**]

Lessons

- Program architecture impacts generalization & provability
- Recursive, modular neural architectures are easier to reason, prove, generalize
- Explore new architectures and approaches enabling strong generalization & security properties for broader tasks

Challenges for Security at Learning Level

- Evaluate system under adversarial events, not just normal events
- Reason about complex, non-symbolic programs
- Design new architectures & approaches with stronger generalization & security guarantees
- Reason about how to compose components

Compositional Reasoning

- Building large, complex systems require compositional reasoning
  - Each component provides abstraction
    - E.g., pre/post conditions
  - Hierarchical, compositional reasoning proves properties of whole system

- How to do abstraction, compositional reasoning for non-symbolic programs?

Security of Learning Systems

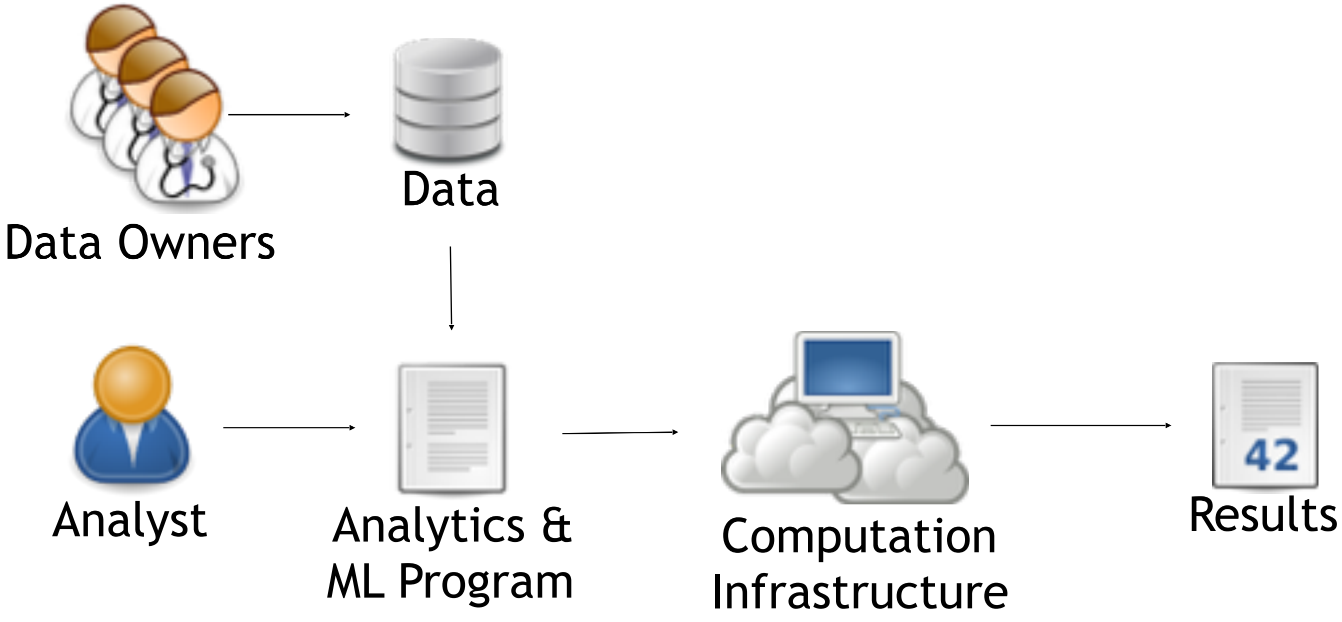
- Software level
- Learning level
  - Evaluate system under adversarial events, not just normal events
  - Reason about complex, non-symbolic programs
  - Design new architectures & approaches with stronger generalization & security guarantees
  - Reason about how to compose components
- Distributed level
  - Each agent makes local decisions; how to make good local decisions achieve good global decision?

AI and Security: AI in the presence of attacker

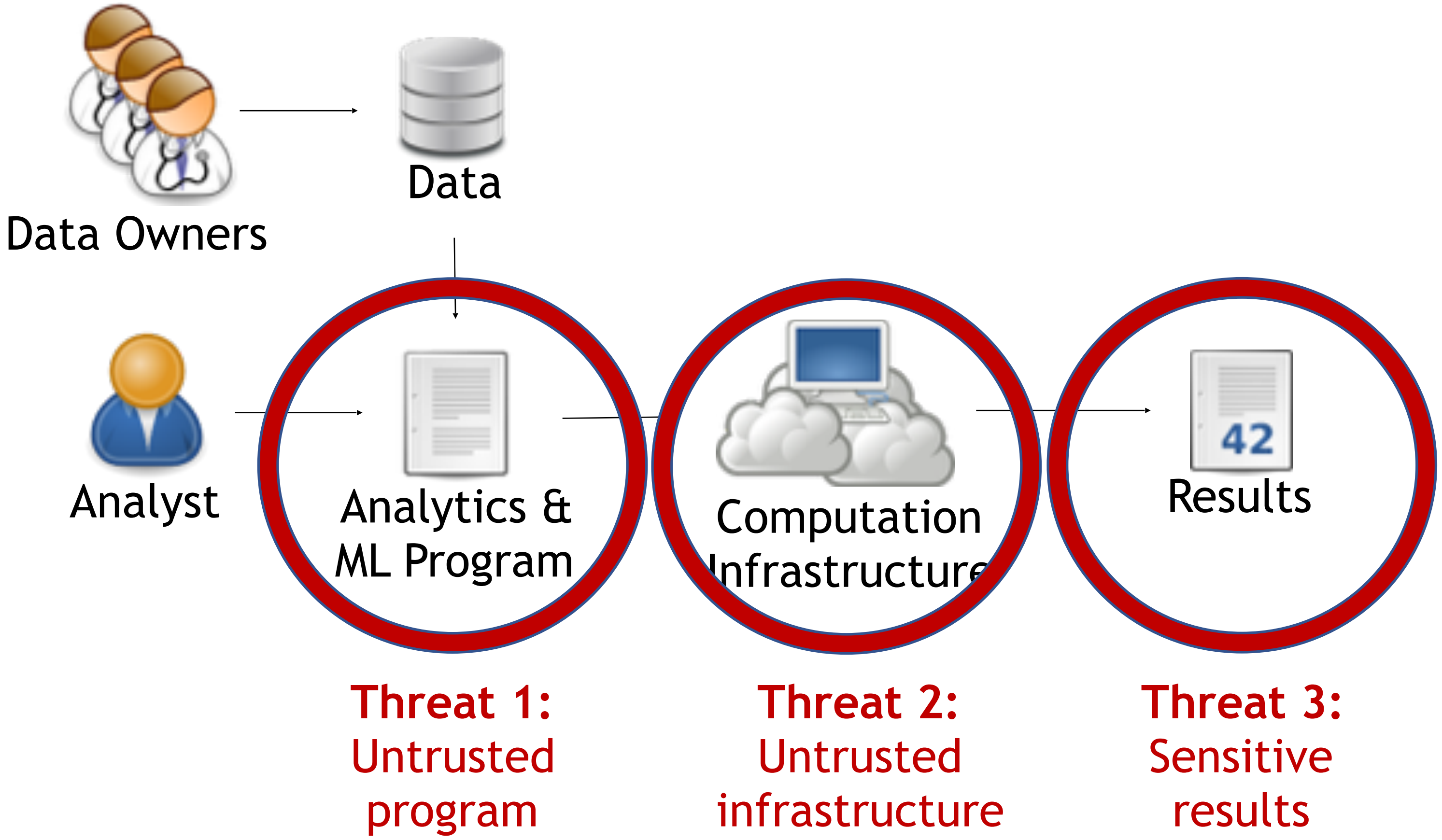
- Attack AI
  - Integrity:
    - Cause learning system to not produce intended/correct results
    - Cause learning system to produce targeted outcome designed by attacker
  - Confidentiality:
    - Learn sensitive information about individuals
  - Need security in learning systems

- Misuse AI
  - Misuse AI to attack other systems
    - Find vulnerabilities in other systems
    - Target attacks
    - Devise attacks
  - Need security in other systems

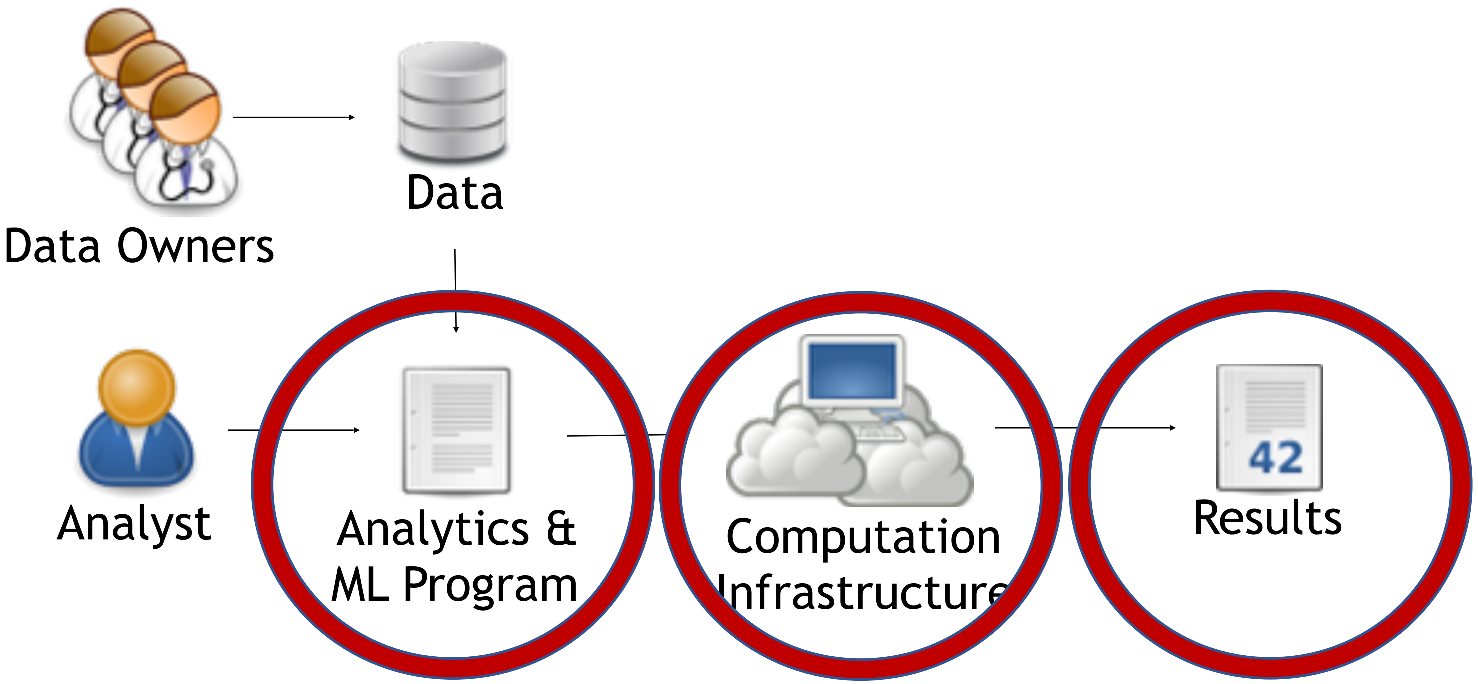
Current Frameworks for Data Analytics & Machine Learning



Current Frameworks Insufficient



Desired Solutions for Confidentiality/Privacy



Threats	**Threat 1:** Untrusted program	**Threat 2:** Untrusted infrastructure	**Threat 3:** Sensitive results
Desired Solutions	**Program Rewriting & Verification**	**Secure Computation**	**Differential Privacy**

AI and Security: AI in the presence of attacker

- Attack AI
    - Integrity:
        - Cause learning system to not produce intended/correct results
        - Cause learning system to produce targeted outcome designed by attacker
    - Confidentiality:
        - Learn sensitive information about individuals
    - Need security in learning systems

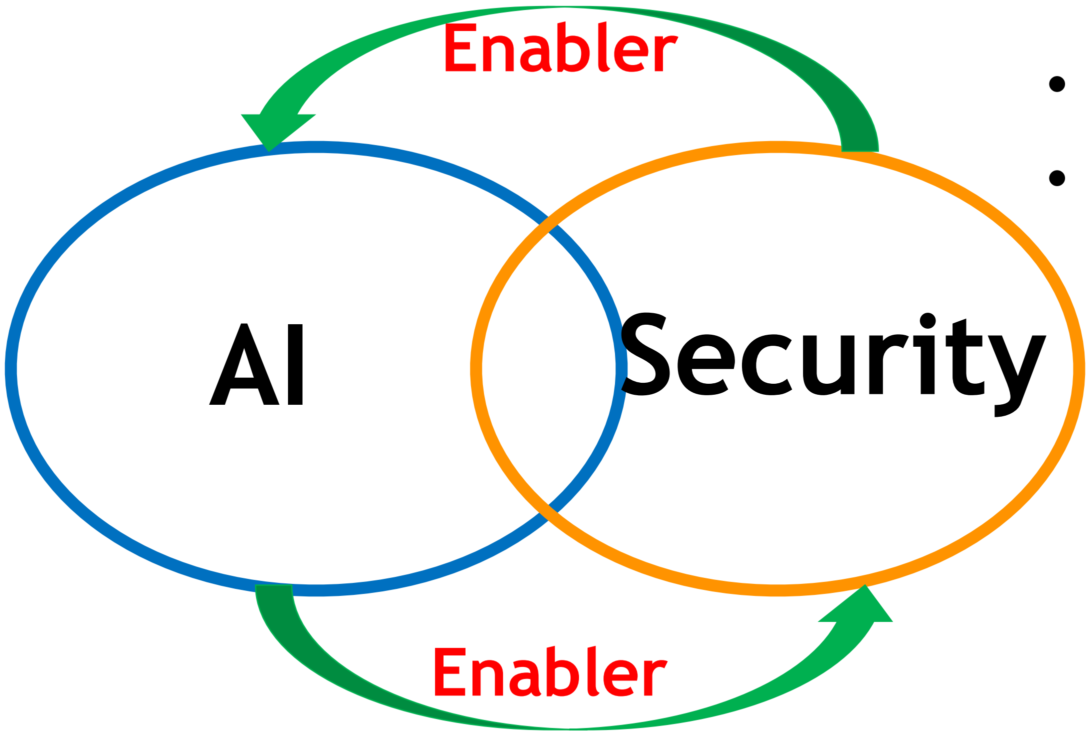
- Misuse AI
    - Misuse AI to attack other systems
        - Find vulnerabilities in other systems
        - Target attacks
        - Devise attacks
    - Need security in other systems

Misused AI can make attacks more effective

Deep Learning Empowered
Bug Finding

Deep Learning Empowered
Phishing Attacks

Deep Learning Empowered
Captcha Solving



- AI enables new security capabilities
- Security enables better AI

Integrity: produces intended/correct results (adversarial machine learning)

Confidentiality/Privacy: does not leak users' sensitive data (secure, privacy-preserving machine learning)

Preventing misuse of AI

Future of AI and Security

How to better understand what security means for AI, learning systems?

How to detect when a learning system has been fooled/compromised?

How to build better resilient systems with stronger guarantees?

How to build privacy-preserving learning systems?

Security will be one of the biggest challenges in Deploying AI.

Let’s tackle the big challenges together!