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
Attacks are increasing in scale & sophistication
Massive DDoS Caused by IoT Devices

- 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

Geographical distribution of Mirai bots in recent DDoS attack
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

Firmware Files → Raw Feature Extraction (disassembler) → Code Graph

Vulnerability Function → Code Graph

Embedding Network \( \phi(\cdot) \)

Cosine Similarity

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

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 Security

• 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

STOP Signs in Berkeley
Adversarial Examples in Physical World

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

  Other tasks and model classes

  Blackbox Attacks
  Weaker Threat Models (Target model is unknown)

  Generative Models
  Deep Reinforcement Learning
  VisualQA/Image-to-code

  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 a 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

Adversarial examples

Reconstruction of adversarial examples

Target Image

Jernej Kos, Ian Fischer, Dawn Song: Adversarial Examples for Generative Models
Adversarial Examples for VAE-GAN in SVHN

Original images

Reconstruction of original images

Adversarial examples

Reconstruction of adversarial examples

Target Image

Jernej Kos, Ian Fischer, Dawn Song: Adversarial Examples for Generative Models
Adversarial Examples for VAE-GAN in SVHN

Original images

Reconstruction of original images

Adversarial examples

Reconstruction of adversarial examples

Target Image

Jernej Kos, Ian Fischer, Dawn Song: Adversarial Examples for Generative Models
Visual Question & Answer (VQA)

Q: Where is the plane?

Benign image

Mode

Answer: Runway

Fooling VQA

Target: Sky

Adversarial example

Mode

Sky
Q: How many cats are there?

Fooling VQA
Target: 2

Answer:

Benign image

Adversarial example

Mode

Mode

1

2
Adversarial Examples Fooling Deep Reinforcement Learning Agents

Original Frames

Original Frames with Adversarial Perturbation

No. of steps

Score

FGSM Evaluation (0.005)

Training on non-noisy environment

Adversarial Evaluation

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

**Detection**
- Ensemble
- Normalization
- Distributional detection
- PCA detection
- Secondary classification
- Stochastic
- Generative

**Prevention**
- Training process
- Architecture
- Retrain
- Pre-process input
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 [AlSec 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**
- Automatic worm detection & signature/patch generation
- Automatic malware detection & analysis

**Proactive Defense:**
- Bug Finding
- Secure by Construction

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

<table>
<thead>
<tr>
<th>Operation</th>
<th>Regression Testing</th>
<th>Security Testing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Run program</td>
<td>Run program on <strong>normal</strong> inputs</td>
<td>Run program on <strong>abnormal/adversarial</strong> inputs</td>
</tr>
<tr>
<td>Goal</td>
<td>Prevent <strong>normal</strong> users from encountering errors</td>
<td>Prevent <strong>attackers</strong> from finding <strong>exploitable</strong> errors</td>
</tr>
</tbody>
</table>
Regression Testing vs. Security Testing in Learning System

<table>
<thead>
<tr>
<th></th>
<th>Regression Testing</th>
<th>Security Testing</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Training</strong></td>
<td>Train on noisy training data: Estimate resiliency against noisy training inputs</td>
<td>Train on poisoned training data: Estimate resiliency against poisoned training inputs</td>
</tr>
<tr>
<td><strong>Testing</strong></td>
<td>Test on <em>normal</em> inputs: Estimate generalization error</td>
<td>Test on <em>abnormal/adversarial</em> inputs: Estimate resiliency against adversarial inputs</td>
</tr>
</tbody>
</table>
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
- IronClad/IronFleet
- FSCQ
- CertiKOS
- miTLS/Everest
- EasyCrypt
- CompCert
Powerful Formal Verification Tools + Dedicated Teams

Coq + Isabelle + Why3 + Dafny + 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

<table>
<thead>
<tr>
<th>Training data</th>
<th>Input</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>123</td>
<td>452</td>
<td>612</td>
</tr>
<tr>
<td>234</td>
<td>345</td>
<td>367</td>
</tr>
<tr>
<td>357</td>
<td>797</td>
<td>979</td>
</tr>
</tbody>
</table>
Neural Program Synthesis

Training data
123
234
357

452
345
797

612
367
979

Input
Output

Neural Program Architecture

Test input
50
70

Learned neural program

Test output
120
Neural Program Architectures

Neural Turing Machine (Graves et al)
Stack Recurrent Nets (Joulin et al)
Learning Simple Algorithms from Examples (Zaremba et al)


Reinforcement Learning Neural Turing Machines (Zaremba et al)
Neural Programmer (Neelankatan et al)
Neural Programmer-Interpreter (Reed et al)
Neural GPU (Kaiser et al)
Differentiable Neural Computer (Graves et al)

Neural Program Synthesis Tasks: Copy, Grade-school addition, Sorting, Shortest Path
Challenge 1: Generalization

**Training data**
- 123
- 234
- 357

**Neural Program Architecture**

<table>
<thead>
<tr>
<th>Input</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>452</td>
<td>612</td>
</tr>
<tr>
<td>345</td>
<td>367</td>
</tr>
<tr>
<td>797</td>
<td>979</td>
</tr>
</tbody>
</table>

**Test input**
- 34216
- 24320

**Learned neural program**

**Test output**
- 541

**Length**
- length = 3
- length = 5
Challenge 2: No Proof of Generalization

Trainig data

\[
\begin{align*}
123 & & 452 & & 612 \\
234 & & 345 & & 367 \\
357 & & 797 & & 979
\end{align*}
\]

Input

Output

length = 3

\[
\begin{align*}
34216 & \\
24320
\end{align*}
\]

Test input

Neural Program Architecture

Learned neural program

Test output

\[
\begin{align*}
58536
\end{align*}
\]
Our Approach: Introduce Recursion

Learn recursive neural programs

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
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
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

Data Owners → Data
Data → Analyst → Analytics & ML Program → Computation Infrastructure → Results
Current Frameworks Insufficient

Data Owners -> Data

Analyst

Analytics & ML Program

Threat 1: Untrusted program

Computation Infrastructure

Threat 2: Untrusted infrastructure

Results

Threat 3: Sensitive results
Desired Solutions for Confidentiality/Privacy

- Data Owners
- Data
- Analyst

Threats
- Threat 1: Untrusted program
- Threat 2: Untrusted infrastructure
- Threat 3: Sensitive results

Desired Solutions
- Program Rewriting & Verification
- Secure Computation
- Differential Privacy

Analytics & ML Program
Computation Infrastructure
Results
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!