Data and Related Issues in Applying Machine Learning to Cyber Security

James Halvorsen
School of EECS, WSU

CySER Seminar
Nov 28, 2022

A Reflection on Machine Learning

Machine learning uses a dataset, and some training algorithm, to learn a function or discover knowledge

Types

Supervised (data has labels y, tries to learn y = f(x))

Unsupervised (data has no labels, tries to separate data into categories)

Reinforcement (try to learn a policy)

Algorithms for training ML models

Decision trees/random forests

Support vector machines

Neural networks

etc

Applications of Machine Learning in Cyber Security

Intrusion Detection/Prevention

- Most well-researched
- Tries to learn if cyber monitoring data is a threat
- Different implementation types
  - Host vs Network
  - Signature vs Anomaly
- False positive rate is most concerning metric

Incident Response

- Tries to find appropriate responses to detected attacks
- May need to answer higher level questions about attacker (i.e. what is their goal)

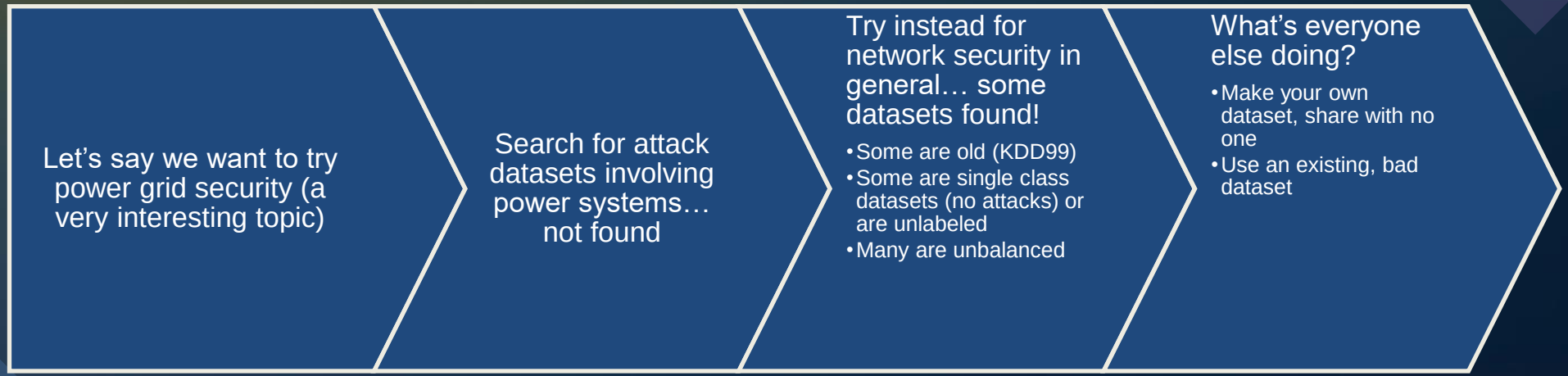
Automated Red Teaming

- Tries to find an ideal strategy to find weaknesses in a network
- Existing work: uses reinforcement learning and simulated environments

Why is Data Important?

- Suppose you are a CS PhD student
- Want to research intrusion detection – automate away the issue of cyber attacks
- Need a few things to implement any solutions:
  - A testbed for conducting experiments
  - A dataset for training the IDS

Let’s Look for Some Data Then



Let's… Make Some Data Then?

- First we setup a testbed – what's our hardware budget again?
- Then we'll simulate some cyber attacks
  - Which ones, and how many?
  - Do we need a pentester, or can we use some adversary emulation tools?
  - How much time do we have to spend on creating a dataset?
- And we'll collect some data while doing it
  - And then we have to label it all.

Say, What Data Do We Collect Anyways?

Types of network data:

- Netflows (summary of network transactions)
- PCAPs (very fine-grained)

Types of host data

- Security logs
- Events (process creation, file creation, driver loaded, etc…)

How much do we collect?

- Everything: needle in a haystack
- Nothing: no record of attacks
- Answer is *somewhere* in the middle

Observability: Measuring Data Relevance to a Network

Suppose we have some network of computers

Want to know what data to collect from each computer

Data should show attacks we could (and would likely) receive

Data should not show normal behavior as attacks

Current solution: TOMATO (Threat Observability & Monitoring Assessment Tool).

- Uses conditional probabilities + Markov Analysis
- Could be improved upon

Synthetic Data – Improving Existing Datasets

Instead of making our own dataset, how about improving an existing one?

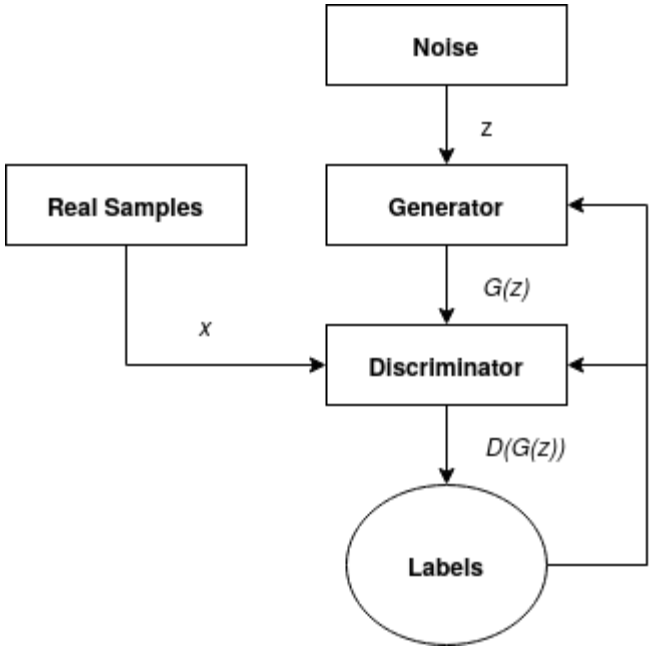
Available tools:

Generative Adversarial Networks
Variational Autoencoders
Diffusion Models

Existing research suggests generative models can solve balance problems

Some issues specific to security data when using generative models

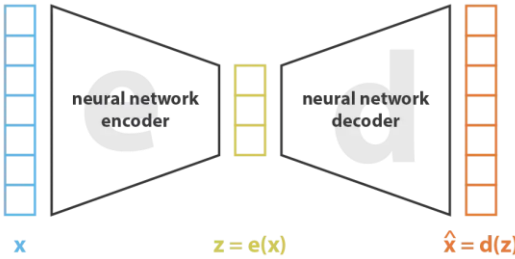
Generative Adversarial Networks (GANs)



GAN Training Process

- Most well-researched generative ML method
- Trains generator and discriminator via adversarial method
  - Generator: converts random noise into data
  - Discriminator: distinguishes between synthetic and real data
- Common Issue: Mode Collapse (keeps producing same samples)
  - Less of an issue with Wasserstein GANs
- Many different variants
  - BiGAN
  - Conditional GAN
  - Deep Convolutional GAN
  - etc

Variational Autoencoders (VAEs)



$$\text{loss} = \|x - \hat{x}\|^2 = \|x - d(z)\|^2 = \|x - d(e(x))\|^2$$

General architecture of a variational autoencoder with loss function. Diagram is from a web article “Understanding Variational Autoencoders (VAEs)” by Joseph Rocca, 2019.

Source: https://towardsdatascience.com/understanding-variational-autoencoders-vaes-f70510919f73

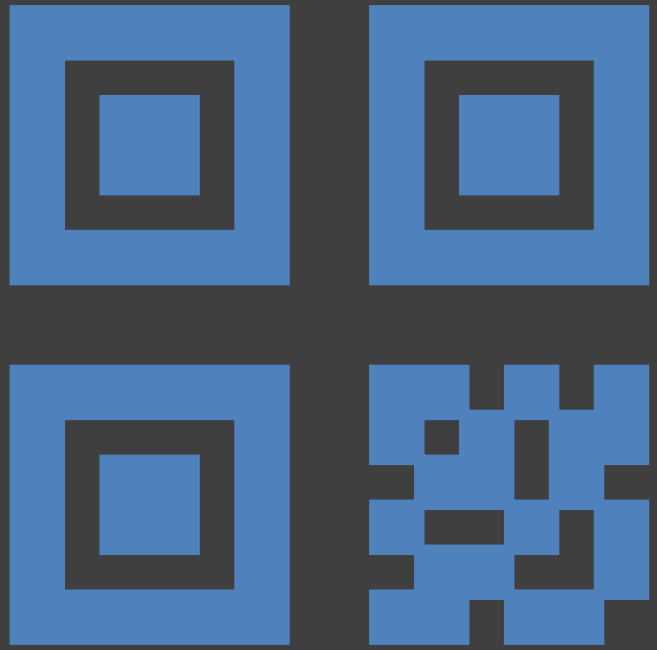
- Method of training an “autoencoder” model
  - Encoder: Convert data into latent representation
  - Decoder: Convert noise/latent representation into data
- Uses a probabilistic approach for training, different from standard autoencoder training
- Roughly as old as GANs
- When used for generating images, results may appear blurrier
- Variant: Conditional VAEs
  - Can generate “structured” data (i.e. (x,y) instead of x)

Diffusion Models

Gradual denoising applied to CIFAR-10 image dataset.

From the paper “Denoising Diffusion Probabilistic Models” by Ho et al, 2020

- Relatively new method for generating data
- Uses a denoising process for generation
  - Progressively add more noise to training data
  - Learn to reverse process
- Major success an AI generated images (DALL-E 2, Stable Diffusion)
- Not much use in cyber security yet.



Feature Extraction – An Issue for All Types of Security Data

- ML models operate on feature vectors
  - Constant width, IEEE float array
  - Errors will occur, should correlate with magnitude (off by 1 vs off by 200)
- Most security features categorical, hard to map over
- Available techniques:
  - One-hot encoding (creates large feature vectors)
  - Binary encoding
  - Dimensionality reduction

Feature Extraction (cont'd)

- Netflow
  - Categorical features: IP Addresses, Ports
  - Integer features: # of packets, total data sent
- Security/Event logs
  - Some text fields are purely categorical (i.e. event type)
  - Some fields may involve filepaths – difficult to represent
- Malware Samples
  - More complicated to generate, can't encode instructions easily
  - Can encode function call graphs, system calls
  - Often captured using honeypots

Evaluation – Another Issue for Synthetic Security Data

- Most generative ML research performed on images
  - Easy to evaluate visually
  - Easy to evaluate using distance metrics, data already used continuous features
- Security datasets *can* be evaluated visually… if you have a lot of time
- Want metrics that can evaluate realism

Evaluation – Types of Metrics

Train on Synthetic, Test on Real

- Turns problem into evaluating a classifier (easy)
- Needs a point of comparison

Distance Metrics

- Example: Frechect Inception Distance (FID)
- May have issues between datasets of different sizes

Metrics Tailored to Dataset

- Can use domain knowledge about invalid states
- May take time to design a custom metric

Other Issues: Data Anonymity

Sensitive information may be captured depending on testbed

Real data should be anonymized in some way

Synthetic data should not propagate sensitive info

Summary

Machine learning has several applications in cyber security, but depends heavily on data

Datasets for cyber security may be non-existent or of poor quality depending on domain

Creating data for cyber security requires answering some difficult questions

What types of data are useful in this domain?	How do we extract features that are useful for our machine learning models?	How do we evaluate the quality of data?