Special Topics: Machine Learning (ML) for Networking

COL867

Holi, 2025

Security

Tarun Mangla

Application Performance Monitoring

Tradition N/w mo ← Bottom up data collection

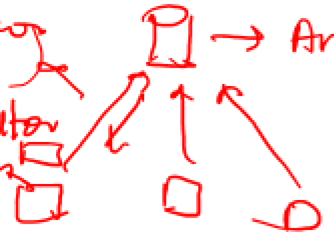
Top-down collection + Curren[illegible]

- Two ML-based methods for video streaming performance monitoring
  - Features derived based on the knowledge of the underlying network protocol
  - Explainability of features NetMicroscope
  - Scalability concerns → monitoring & analysis

communication & storage → Analyzer

Monitor & analysis

Streaming / sketch algorithm



- What about other applications?
  - Video conferencing → ① Underlying protocols | ② What are the metrics?
  - Web browsing

Page load time

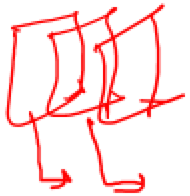
Packet delay

③ A/V synchronization

④ Frame rate

① Frame delay

② Video quality → resolution



⑤ Frame jitter

Quiz

Case Studies: ML for Specific Network Learning Tasks

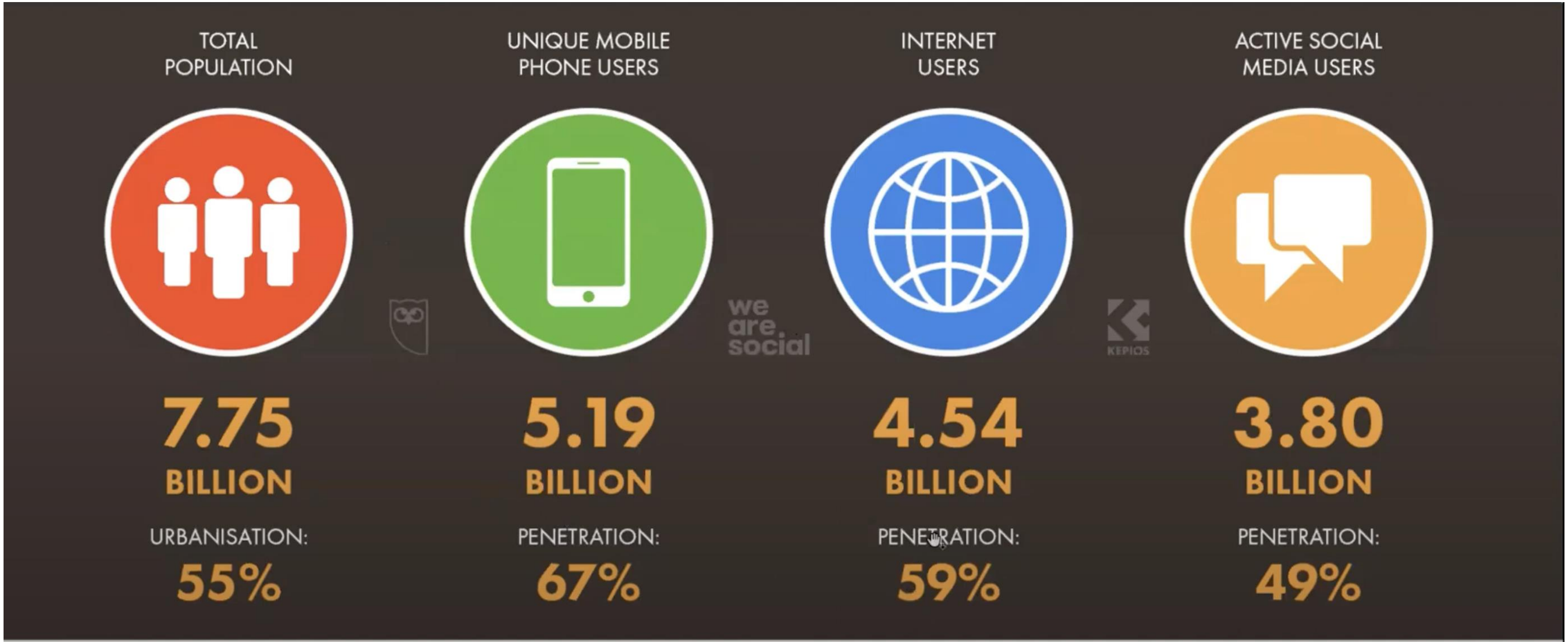
- Application Classification
- Application Performance Monitoring
- **Security**
- Resource Allocation

Agenda

- Background on network security
- Why ML is a good candidate for network security?
- Two kinds of techniques
  - Misuse detection
  - Anomaly detection
- Anomaly detection techniques
  - Detecting Spearphishing Attacks
  - Kitsune

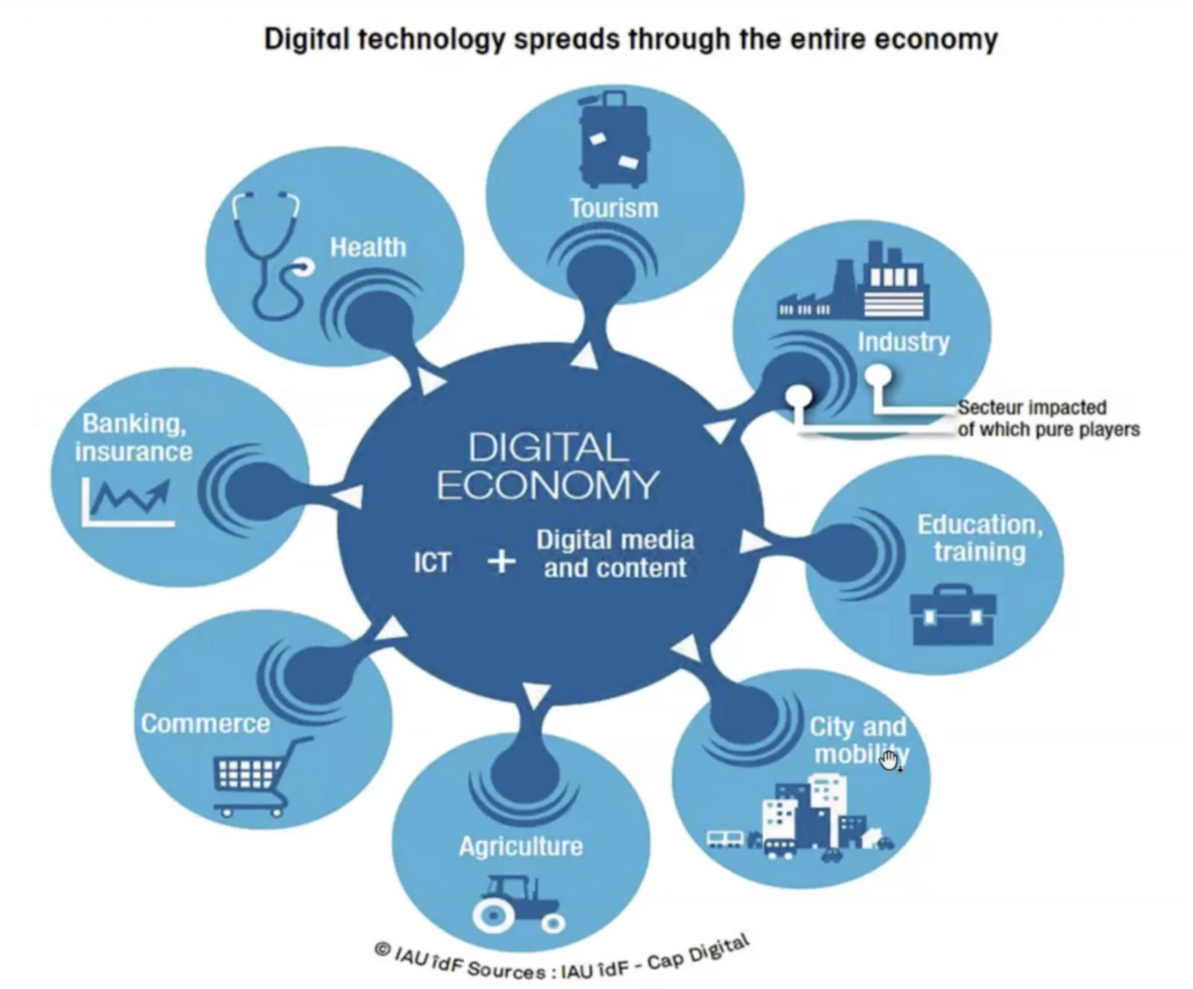
Some slides borrowed from Rajiv Barua’s talk

We live in an increasingly connected world



Source: United Nations, Local Govt bodies, HootSuite

We need these systems to be secure



Distributed
Denial of service

Types of Cyberattacks

Server
Network becomes the medium to propagate

Malware

Phishing

Denial of Service

Insider attacks

④ Authentication

At the same time remotely accessible..

① Detect malicious traffic
② Encryption
③ Access control

Network access need to be secure..

System

N/W Security

Networks are attacked frequently

Ransomware

1. Cyberattack on AIIMS

In December 2022, responding to a quer
Union government disclosed that the All
cyberattack, resulting in the encryption o

The Minister of Electronics and Informati
incident was categorized as a "cyber secu
network due to

8. BharatPay hacked: Breaching financial trust

In August 2022, BharatPay, a digital financial services provider in India, experienced a significant dat
breach resulting in the exposure of personal data and transaction details of approximately 37,000 users.

The compromised information includes user names, hashed passwords, mobile phone numbers, UP
IDs, and official email IDs of employees from Indian insurance and banking firms.

The breach was discovered on August 12 by XVigil, the threat intelligence arm of CloudSEK.

Domino's India data theft

The Indian arm of Domino's Pizza revealed in April 2021 that a threat actor had hacked their database and sold the compromised data on a hacking forum.

The actor claimed to have laid their hands on **13 TB** of information comprising data of 18 million orders reflecting customer names, addresses, delivery locations, and phone numbers, along with the credit card information of 1 million individuals from the database of Domino's India. However, the pizza chain claimed that customer credit card data wasn't compromised as they don't maintain the financial records of their clients.

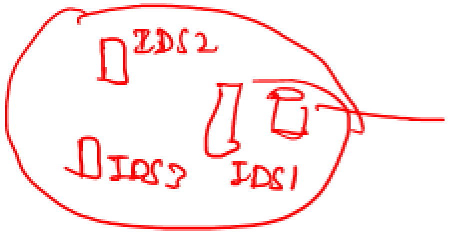
Approach to securing networks..

- Prevention mechanisms include zero trust
  - Traffic encryption
  - Authorization (e.g. 2FA)
  - Access Control

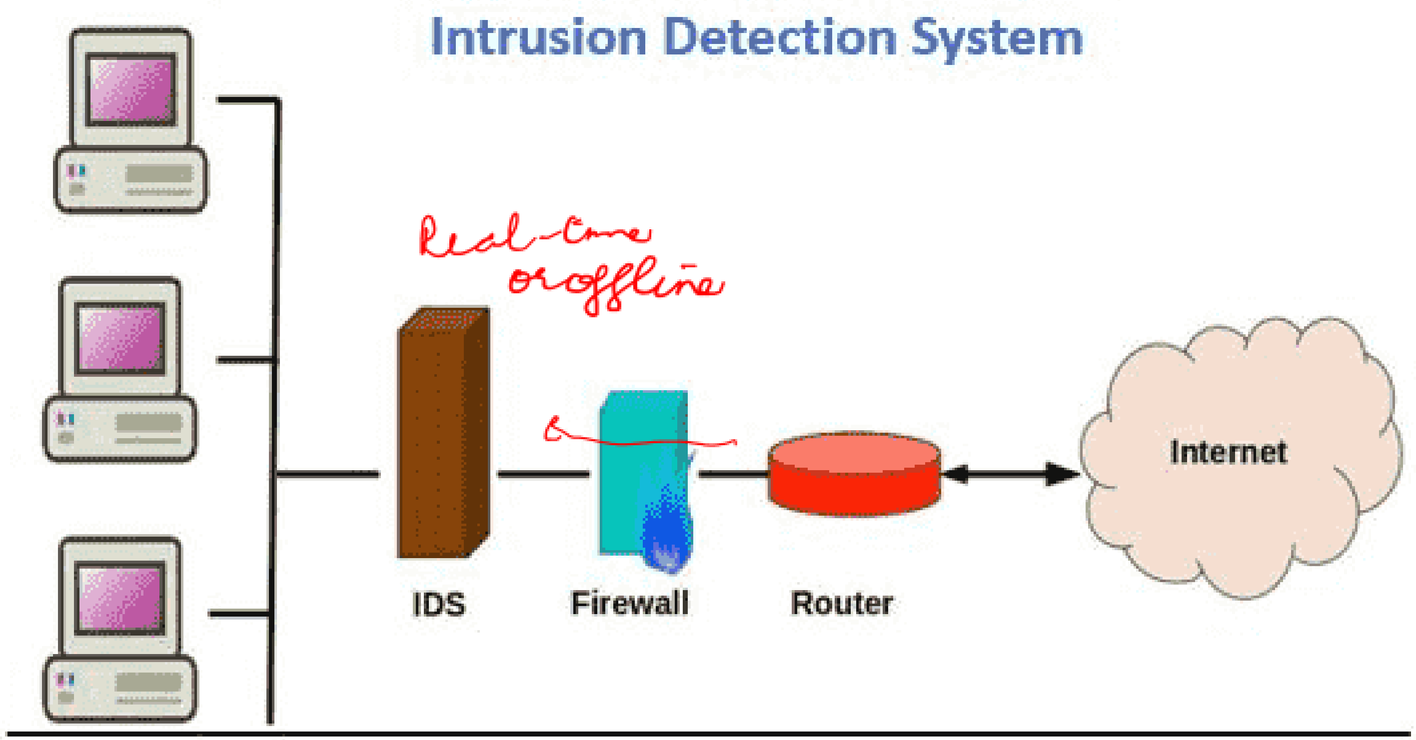
- However, not enough…why?

- Second line of defense:
  - Detect early symptoms of threats
  - Intrusion Detection Systems (IDS)

zero trust



What is an IDS?



Why is ML suitable for Network IDS?

Manual heuristic are not suitable

- Detect rare events
- Zero-day attacks
- Heterogeneous data sources

  Access logs

  Firewall logs

  SMTP logs
- Fast-evolving attacks/Real-time analysis

Two Categories of IDS

Record traffic for Dos attacks

- Misuse-based
  - Match network traffic against known attack signatures
  - What kind of ML paradigm? Supervised
  - Does not work against unknown attacks or zero-day attacks

- Anomaly-based A
  - Assumption: attacks are deviations from expected behavior
  - Is that true?
  - What kind of ML paradigm?

Detecting Credential Spearphishing Attack in Enterprise Settings, USENIX Security’17

What is Spearphishing?

- Targeted communication that tricks victim into giving attacker privileged capabilities

53% of Indian organisations were victim of 'spear phishing' in 2022: Barracuda report

Spear-phishing attacks make up only 0.1 percent of all e-mail-based attacks, according to Barracuda data, but they are responsible for 66 percent of all breaches.

Two Stages of Spearphishing Attack

- **Lure** Alice by embedding a sense of authority or trust in the communication
  - Address spoofing
  - Name spoofing
  - Previously unseen attacker
  - Lateral attacker
- **Exploit** the trust by inducing Alice to perform some dangerous action
  - links containing malware
  - links asking for user credentials
  - out-of-band actions (e.g., transfer money)

<u>Real User</u>
"Alice Good"
<alice@enterpriseX.com>

<u>Address Spoofer</u>
"Alice"
<alice@enterpriseX.com>

<u>Name Spoofer</u>
"Alice Good"
<alice@evil.com>

<u>Previously Unseen Attacker</u>
"Enterprise X IT Staff"
<director@enterpriseY.com>

<u>Lateral Attacker</u>
"Alice Good"
<alice@enterpriseX.com>

Four different impersonation models

Two Stages of Spearphishing Attack

- **Lure** Alice by embedding a sense of authority or trust in the communication
  - Address spoofing
  - Name spoofing
  - Previously unseen attacker
  - Lateral attacker
- **Exploit** the trust by inducing Alice to perform some dangerous action
  - links containing malware
  - links asking for user credentials
  - out-of-band actions (e.g., transfer money)

Real User
"Alice Good"
<alice@enterpriseX.com>

Address Spoofer
"Alice"
<alice@enterpriseX.com>

Name Spoofer
"Alice Good"
<alice@evil.com>

Previously Unseen Attacker
"Enterprise X IT Staff"
<director@enterpriseY.com>

Lateral Attacker
"Alice Good"
<alice@enterpriseX.com>

Four different impersonation models

Overview

- **Goal:** Detect emails containing credential spearphishing attacks in enterprise settings
- **Naïve Approach**: Use anomaly detection to identify attack emails
  - E.g., check if email headers are different from prior headers

Alice Good
<alice@evil.com>

Alice Good
<alice@good.com>

Past emails

- **Challenges:**
  - Limited prior history
  - Churn in header values
- **So what, flag them all!**

Issue with Flagging Them All

- High false positive rate
- Total emails can be in millions (monthly)
- Even 1% FP rate means a significant number of false positive events

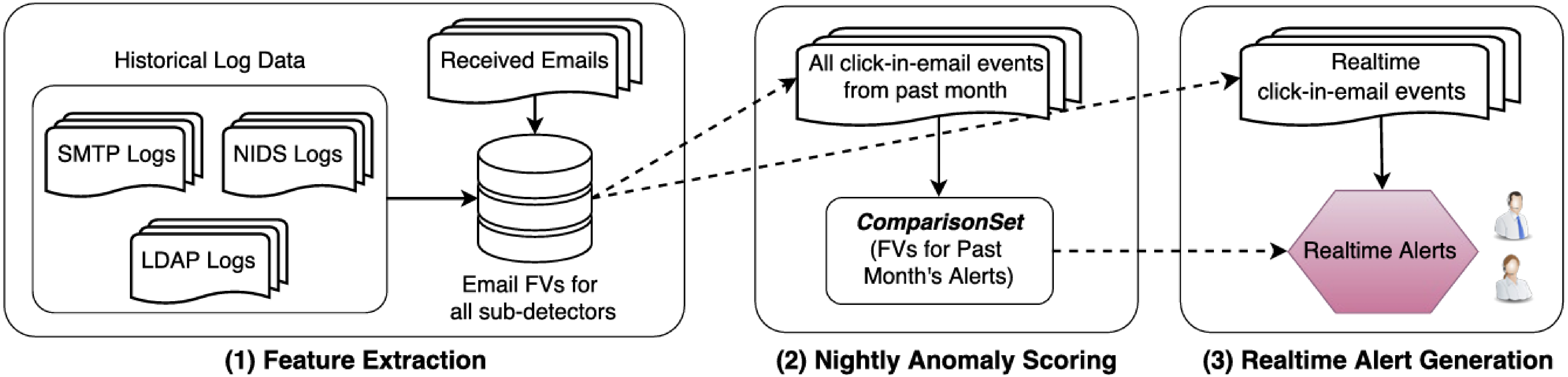
General Issue with anomaly detection for network security

Design Goal: Detect attack emails with high precision and high recall

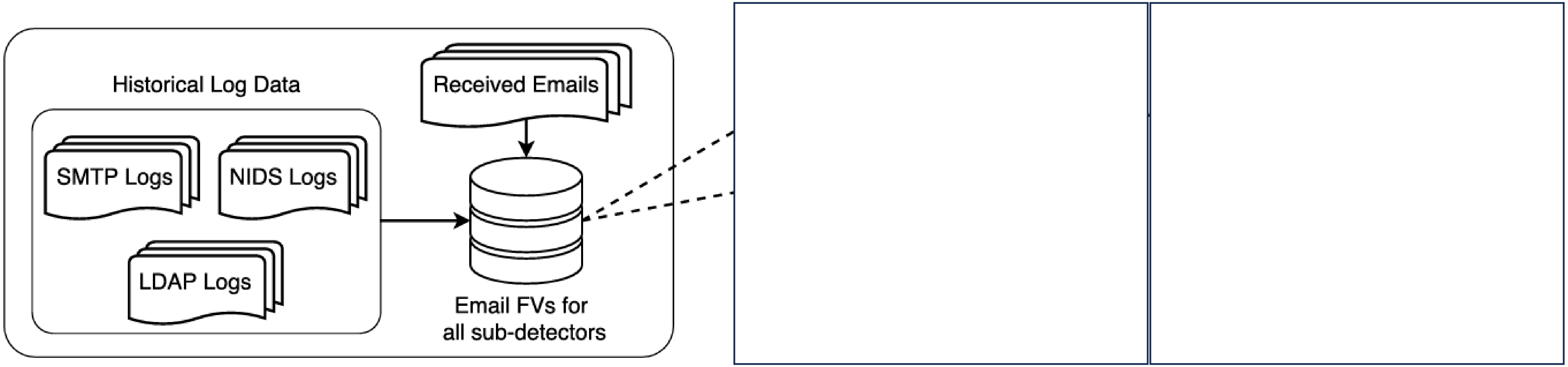
Data Sources

Data Source	Fields/Information per Entry
SMTP logs	Timestamp `From` (sender, as displayed to recipient) `RCPT TO` (all recipients; from the SMTP dialog)
NIDS logs	URL visited SMTP log id for the earliest email with this URL Earliest time this URL was visited in HTTP traffic # prior HTTP visits to this URL # prior HTTP visits to any URL with this hostname Clicked hostname (fully qualified domain of this URL) Earliest time any URL with this hostname was visited
LDAP logs	Employee's email address Time of current login Time of subsequent login, if any # total logins by this employee # employees who have logged in from current login's city # prior logins by this employee from current login's city

Approach Overview



Approach Overview



(1) Feature Extraction **(2) Nightly Anomaly Scoring** **(3) Realtime Alert Generation**

Features

- Sender Reputation (Lure)
  - Three subdetectors, one each for name spoofing, unseen attack, lateral attack
  - Different features for each kind of lure

Lure Features

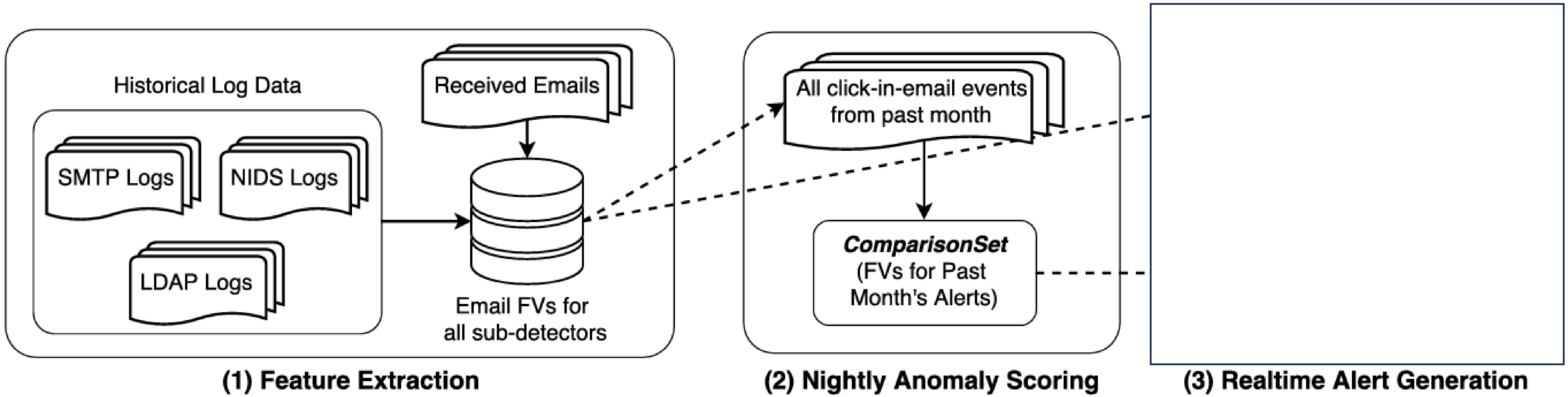
- Name spoofing
  - # days since the same name, email
  - total weeks where name was seen for every weekday (trustworthy name)
- Previously unseen attacker
  - # of prior days *from* address seen
  - # of prior days *from* name seen
- Lateral attacker
  - whether a new IP address detected
  - # of employees that have logged in from the city
  - # previous logins where sender employee logged in from the same IP address

Features

- Sender Reputation (Lure)
  - Three subdetectors, one each for name spoofing, unseen attack, lateral attack
  - Different features for each kind of lure
- Domain Reputation (Exploit)
  - Number of prior visits to the same Fully Qualified Domain Name (not URL)
  - Number of days between the first visit to the FQDN by any employee and the time email initially arrived

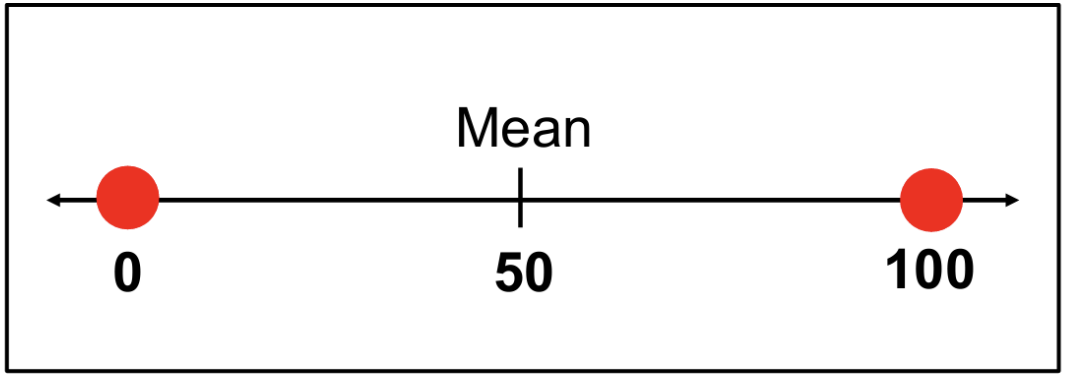
Any other features?

Approach Overview



Using Features

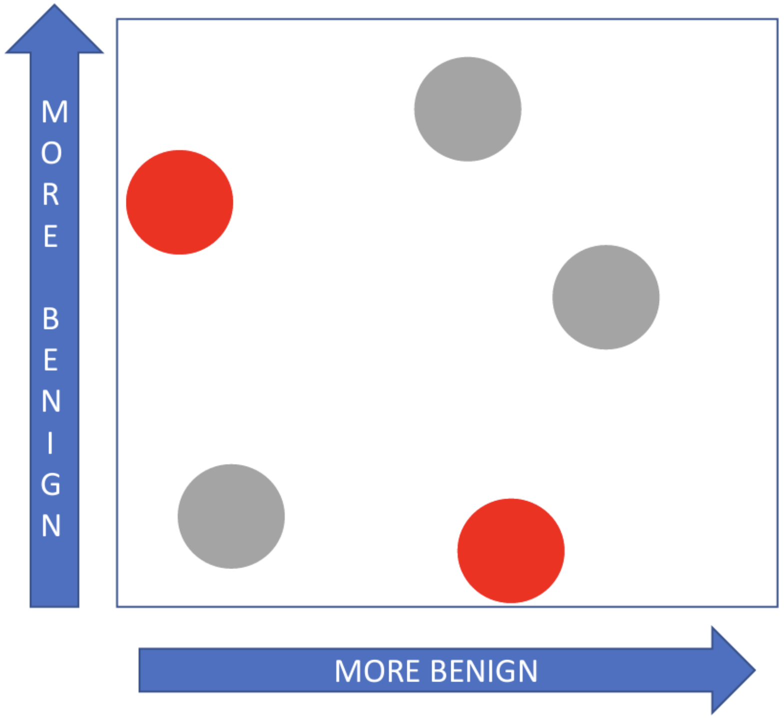
- Combine features to get Feature Vectors
- Use classical anomaly detection
  - K-means, GMM, KDE
- But it has a few issues:
  - **Direction agnostic**



Feature:
prior logins by current employee from city of new IP addr

Using Features

- Combine features to get feature vectors
- Use classical anomaly detection
  - k-means, GMM, KDE
- But it has a few issues:
  - Direction agnostic
  - Alert if anomalous in only one direction



Special Topics: Machine Learning (ML) for Networking

COL867

Holi, 2025

Security

Tarun Mangla

Recap

- Machine learning for network security →

supervised learning zero-day attacks

- Two kinds: signature-based and anomaly detection
- Two papers:
  - Detecting credential spearphishing attack ..
  - Kitsune

Overview: Detecting Credential Spearphishing ..

high FPR → operator has to manually check

- **Goal:** Detect email credential spearphishing with high precision and high recall

3 detectors

Name spoofing

unknown attack

lateral attack

From: "Berkeley IT Staff" <security@berkeley.net>

-----BEGIN PGP SIGNED MESSAGE-----
Hash: SHA1

_AirBears UID 1051850 will be blocked, per the SNS notice associated with tracking number [SNS #902375].

To avoid being blocked from the Airbears network, you must go to the link below and login with your Calnet id and password:

http://auth.berkeley.edu/cas/login/?service=https%3A%2F%2Fsecurity.berkeley.edu%2Flogin%2Fcas

The blocking will be suspended if valid Calnet id and password have been provided no later than 23:59 on Mar 24.

System and Network Security

-----BEGIN PGP SIGNATURE-----
Version: GnuPG v2.0.22 (FreeBSD)

iD8JJlIid+8923ljsdwWTf6yM0oJEJOljwenfiOIEIFFXOwefhliuuNSACeLXka
EJUlyJEoe992webRAURx4xbx=
=6Nch
-----END PGP SIGNATURE-----

Lure: attacker sends catchy email under trusted/authoritative entity

auth.berkley.edu → malicious-xyz.com

-----BEGIN PGP SIGNED MESSAGE-----
Hash: SHA1

_AirBears UID 1051850 will be blocked, per the SNS notice associated with tracking number [SNS #902375].

To avoid being blocked from the Airbears network, you must go to the link below and login with your Calnet id and password:

http://auth.berkeley.edu/cas/login/?service=https%3A%2F%2Fsecurity.berkeley.edu%2Flogin%2Fcas

The blocking will be suspended if valid Calnet id and password have been provided no later than 23:59 on Mar 24.

System and Network Security

-----BEGIN PGP SIGNATURE-----
Version: GnuPG v2.0.22 (FreeBSD)

iD8JJlIid+8923ljsdwWTf6yM0oJEJOljwenfiOIEIFFXOw…uNSACeL…
EJUlyJEoe992webRAURx4…=
=6Nch
-----END PGP SIGNATURE-----

mandrillapp.com/track/click/305639…/auth.berkeley.netne.net?…eyJzIjoiSFA3M1ZvenB5WFRPX094dUozdkpudENM...Zjg3NDA1NjNjZjQ5N1wiLFwidXJsX2lkc1wiOltcImlzN2RiO

Actual Destination for linked text:
auth.berkeley.netne.net

Exploit: embedded links leads to phishing websites

Lure Features

- Name spoofing
  - # days since the same name, email
  - total weeks where name was seen for every weekday (trustworthy name)

SMTP logs

- Previously unseen attacker
  - # of prior days *from* address seen
  - # of prior days *from* name seen
- Lateral attacker
  - whether a new IP address detected
  - # of employees that have logged in from the city
  - # previous logins where sender employee logged in from the same IP address

Name Spoofer
"Alice Good"
<alice@evil.com>

Previously Unseen Attacker
"Enterprise X IT Staff"
<director@enterpriseY.com>

Lateral Attacker
"Alice Good"
<alice@enterpriseX.com>

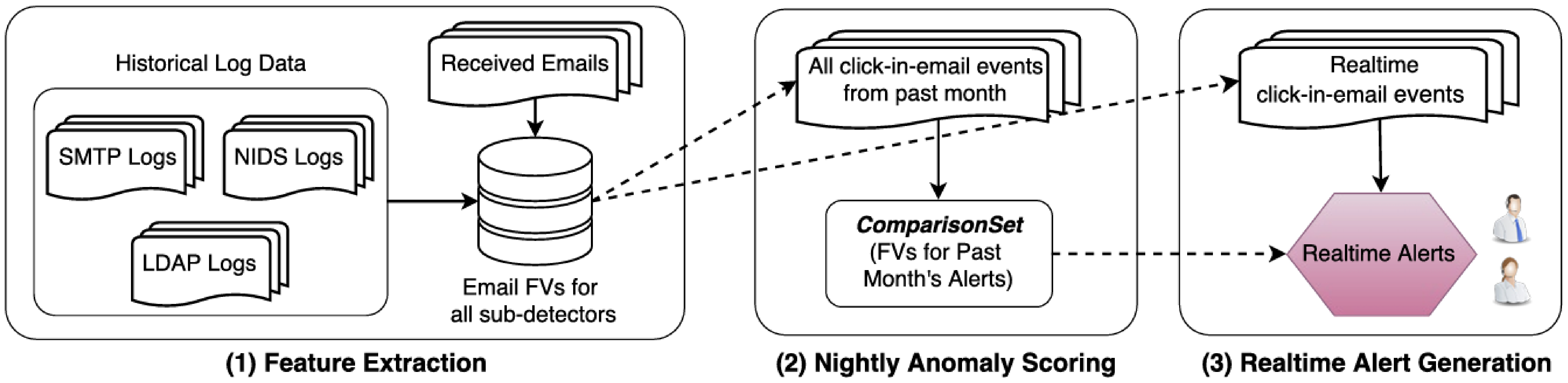
Features

Each detector has 4 features

- Sender Reputation (Lure)
  - Three subdetectors, one each for name spoofing, unseen attack, lateral attack
  - Different features for each kind of lure
- Domain Reputation (Exploit)
  - Number of prior visits to the same Fully Qualified Domain Name (not URL)
  - Number of days between the first visit to the FQDN by any employee and the time email initially arrived

Any other features?

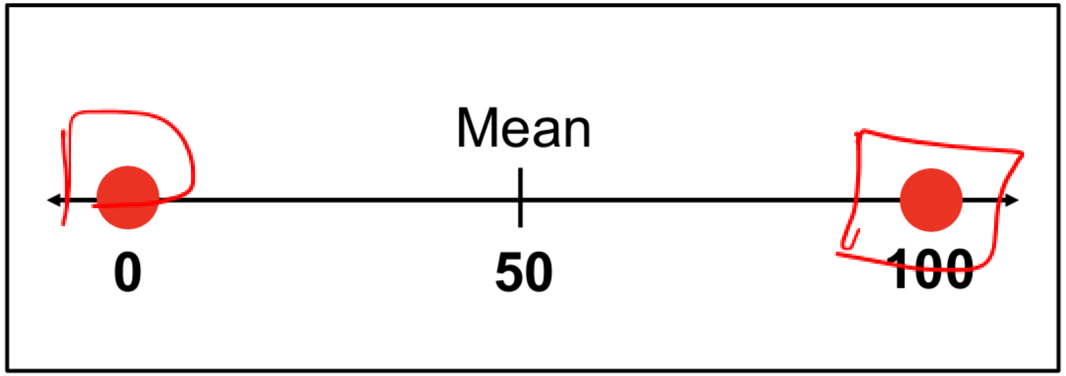
Approach Overview



Using Features

- Combine features to get Feature Vectors
- Use classical anomaly detection
  - k-means, GMM, KDE → anomalous even if it is in one direction
- But it has a few issues:
  - **Direction agnostic** → domain knowledge and define which direction is anomalous

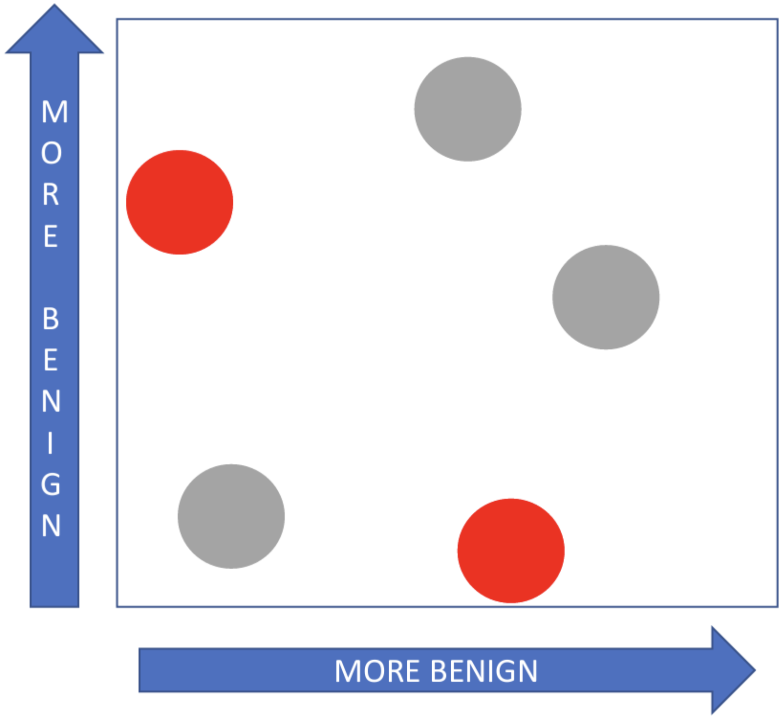
Anomalies



Feature:
prior logins by current employee from city of new IP addr

Using Features

- Combine features to get feature vectors
- Use classical anomaly detection
  - k-means, GMM, KDE
- But it has a few issues:
  - Direction agnostic
  - Alert if anomalous in only one direction

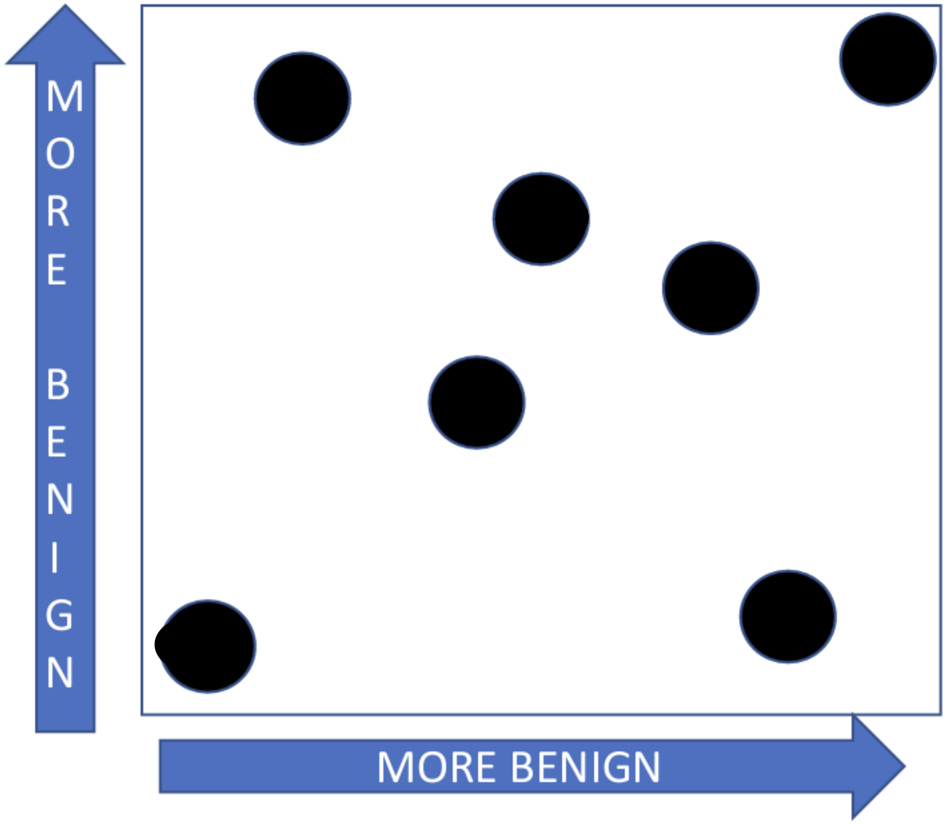


Use Directed Anomaly Scoring (DAS)

- Security analyst with limited budget: specify $\boxed{\textbf{B} = \text{alert budget}}$
- For set of events, assign each event a suspiciousness score
- Rank event by their suspiciousness
- Output the **B** most suspicious events for security team

DAS Example

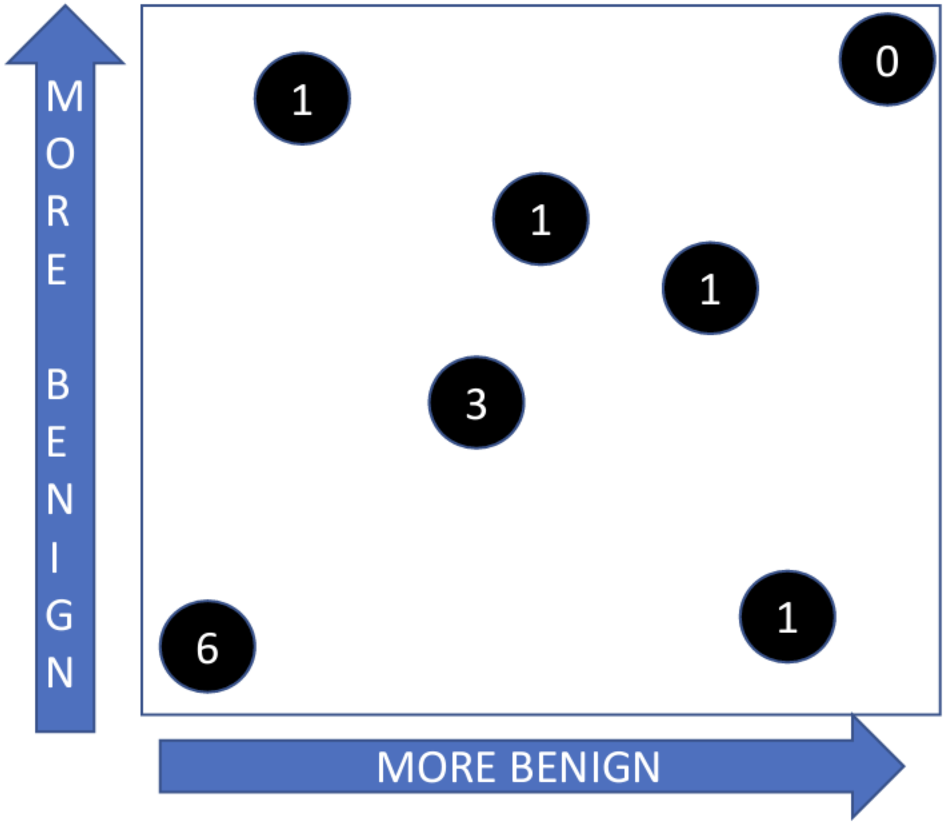
- Score(Event X) = # of other events that are as benign as X in every dimension
  - Large score: many other events are more benign than X



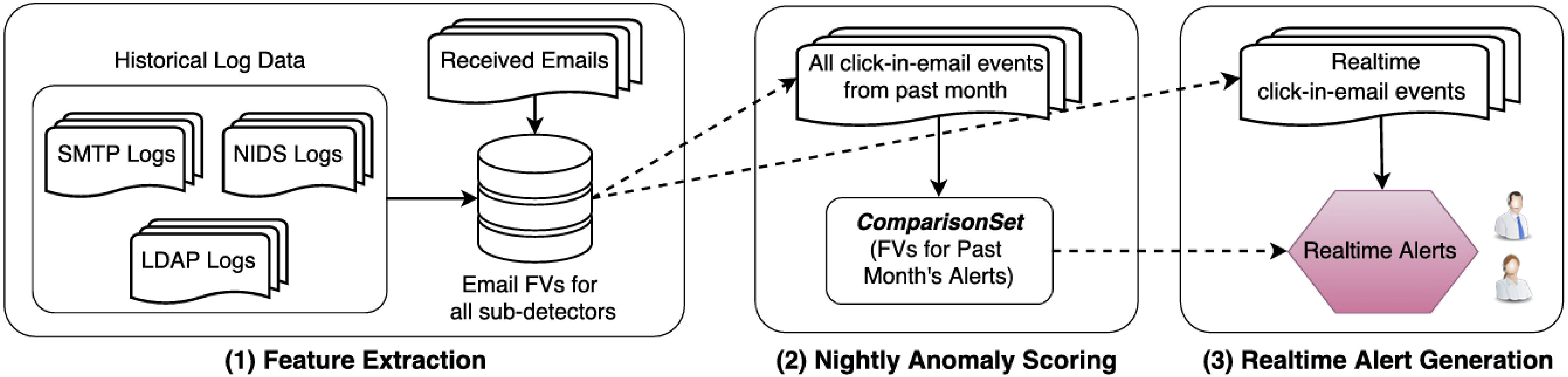
① Random Subset
② Mix of benign, suspicious
③ More benign..
④ Most suspicious email] Benign but
X_1 □ x_1, x_2] → Uni feature
x_3, x_n] DR feature
$[y_1 \ldots y_n]$

DAS Example

- Score(Event X) = # of other events that are as benign as X in every dimension
  - Large score: many other events are more benign than X



Approach Overview



How to Make it Real Time

- Ideally, look at a batch of emails for best results. But not *real-time*
- Keep a Comparison Set of 30xB most suspicious events
- For a new event, check if it is as suspicious as any of the 30xB events in the Comparison Set

N

n_i were -

DAS_i > DAS (comparison set)

Detection Results

- Real-time detector on 370 million emails over 4 years

- Ran detector with a budget of 10 alerts/day

- Detected 17/19 spearphishing attacks (89% TP) → 15 alerts per day
  - 2 / 17 detected attacks were *previously undiscovered*

- Best classical anomaly detection: 4/19 attacks for same budget
  - Need budget >= 91 alerts/day to detect same # of attacks as DAS

Food for thought

- Simple yet effective technique situated in practical constraints

- Generalizable to other anomaly detection tasks
  - DAS Assumption

Jan 1 Jan 31

~~Feb 28~~ ~~Mar 1~~

- ML with human in the loop

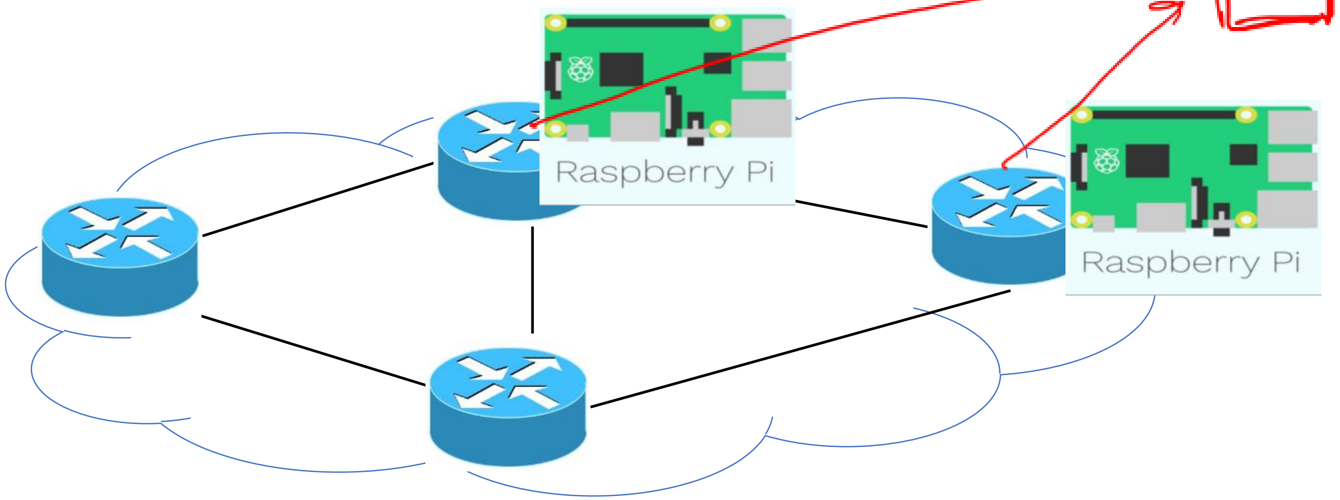
Kitsune: An Ensemble of Autoencoders for Online Network Intrusion Detection (NDSS18)

- Neural networks are useful for Network IDS (NIDS)
  - Can learn complex patterns
- But..
  - Require offline processing of network traces to train models
  - Mostly used in supervised learning – not useful for zero-day attacks
  - High complexity: difficult to run on routers, especially in a distributed NIDS

Kitsune Design Goals

Develop Plug-and-play Neural Network-based NIDS that support:

- Training and testing in online manner: can only store a small number of instances
- Unsupervised learning: no labeled data
- Low complexity: packet processing rate > packet arrival rate



Centralized server

Train → online

Execute → online

Kitsune Idea

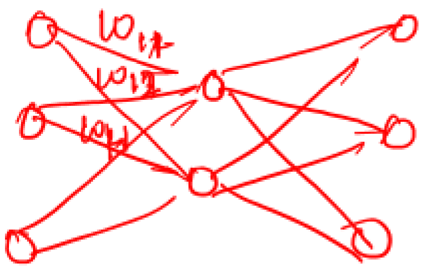
- Use an ensemble of autoencoders for anomaly detection. But how?
  - Consider reconstruction error: RMSE
  - High RMSE → Generate alert

- Lightweight training and execution of autoencoders
  - Feature extraction in a streaming manner
  - On-site training using stochastic gradient descent

(x, y) w_{ij} loss is minimized

Minimum $\|y - \hat{y}\|$

$\|x - \hat{x}\|$



$\hat{y}$

① Batched gradient descent

② Stochastic gradient descent

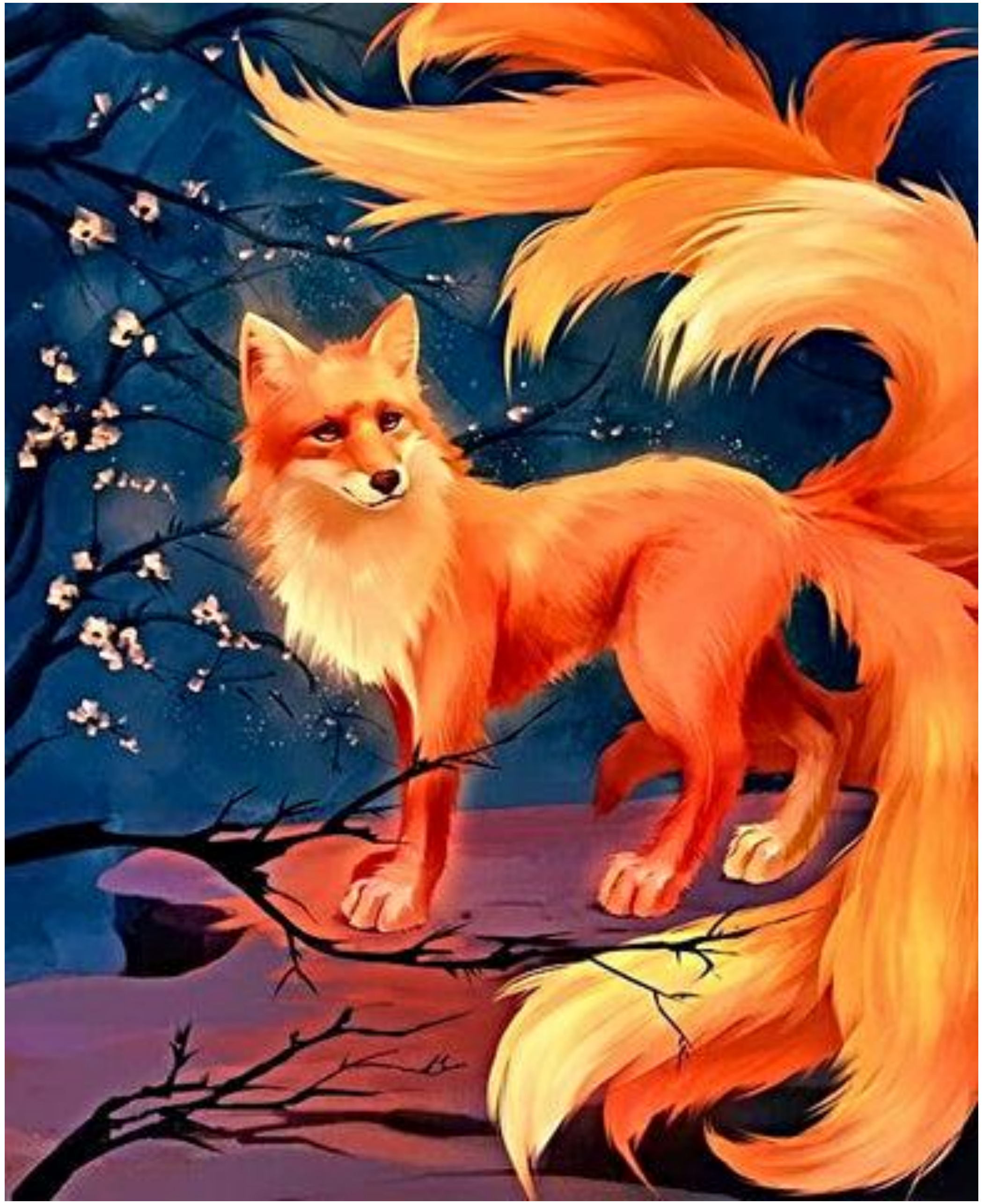
Train phase

- For each packet
  - → compute feature
  - → update weights of ensemble of autoencoder

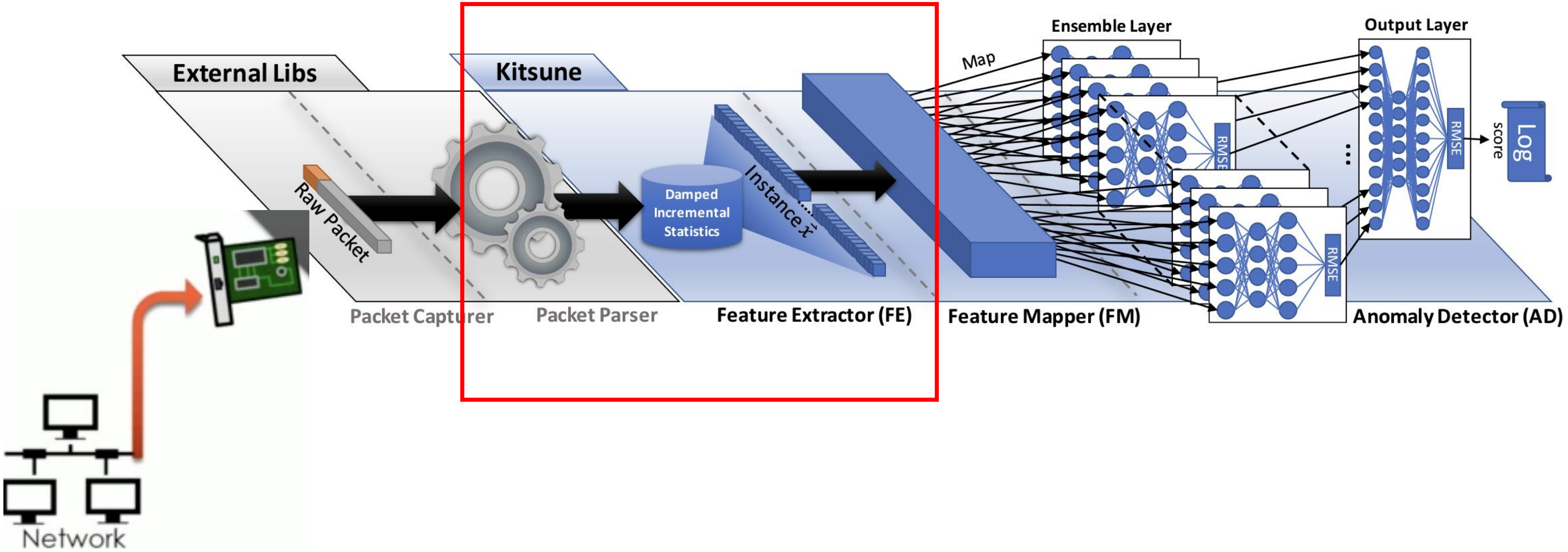
Kitsune Idea

- Use an ensemble of autoencoders for anomaly detection. But how?
  - Consider reconstruction RMSE
  - High RMSE → generate alert

- Lightweight training and execution of autoencoders
  - Feature extraction in a streaming manner
  - On-site training using stochastic gradient descent



Kitsune Architecture



Requirement: more weight to recent packets

O(n) memory

windowed mean → n-packet n variables

[linear sum $= \Sigma x_i$, count $= n$] link

Feature Extractor

- What are important features in a traffic?
  - packet size, inter-arrival times, count

dampening coefficient

Instance)

- Calculate statistics for a feature in a streaming manner

$$IS := (N, LS, SS), \qquad IS \leftarrow (N+1, LS+x_i, SS+x_i^2),$$

- **Challenge**: Keeps long-term history
- **Solution:** Use a damped incremental statistics

Σx_i^2

$$d_\lambda(t) = 2^{-\lambda t} \qquad IS_{i,\lambda} := (w, LS, SS, SR_{ij}, T_{last}),$$

$d_\lambda(t) = 2^{-\lambda(T_{new} - T_{last})} = \gamma$

- Features from 5 time windows:
  - Using different values of lambda

$w \leftarrow \gamma w + 1$

$LS \leftarrow \gamma LS + x_i$

$SS \leftarrow \gamma SS + x_i^2$

Type	Statistic	Notation	Calculation
1D	Weight	w	w
	Mean	μ_{S_i}	LS/w
	Std.	σ_{S_i}	$\sqrt{\lvert SS/w - (LS/w)^2 \rvert}$
2D	Magnitude	$\lVert S_i, S_j \rVert$	$\sqrt{\mu_{S_i}^2 + \mu_{S_j}^2}$
	Radius	R_{S_i,S_j}	$\sqrt{(\sigma_{S_i}^2)^2 + (\sigma_{S_j}^2)^2}$
	Approx. Covariance	Cov_{S_i,S_j}	$\frac{SR_{ij}}{w_i + w_j}$
	Correlation Coefficient	P_{S_i,S_j}	$\frac{Cov_{S_i,S_j}}{\sigma_{S_i}\sigma_{S_j}}$

23 feature x 5 [5 diff time window)

115 x 50

Feature Mapper

$O(n^2)$

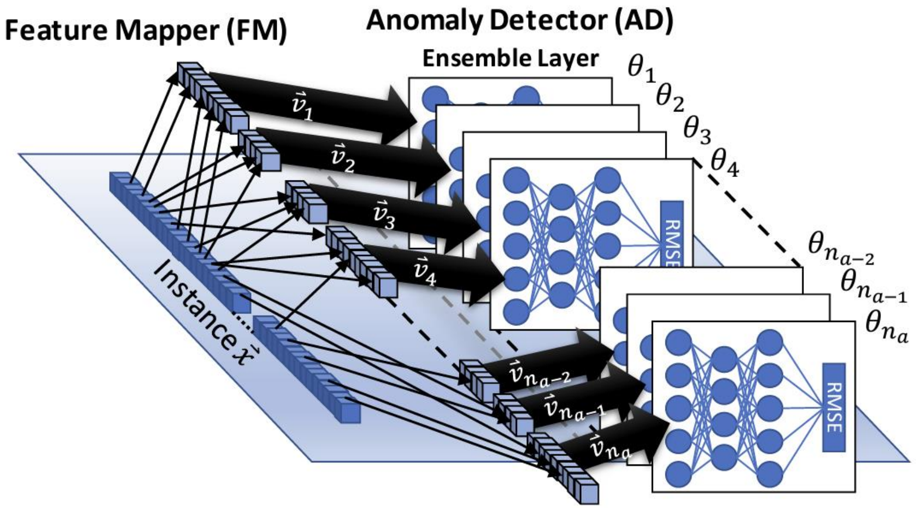
$O(\beta n^2)$

$m \quad \beta n \quad n$

$\beta < 1$

- Feeding features into a single neural network → High complexity

- Instead, create subspace of features, where:
  - Each subspace has no more than m features
  - Each feature is mapped to exactly one subspace
  - The subspace contain correlated features



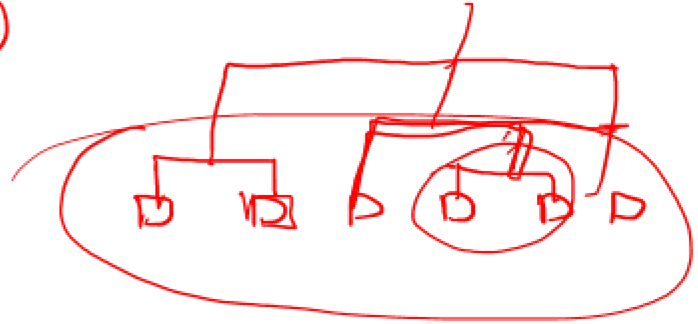
- Does that save computation complexity?

$O\left(\beta\left(\frac{m}{k}\right)^2\right)$

- How do we create the subspaces?

Creating Subspaces

O(n^2)



- Use agglomerative hierarchical clustering

- Use correlation as distance:

O(n^2)

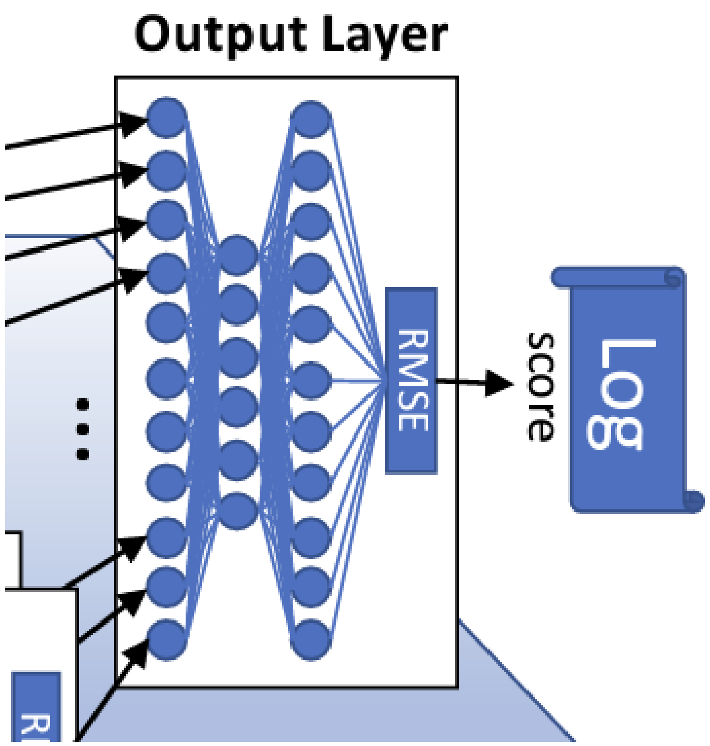
$$d_{cor}(u, v) = 1 - \frac{(u - \bar{u}) \cdot (v - \bar{v})}{\|(u - \bar{u})\|_2 \|(v - \bar{v})\|_2}$$

$$[C_{i,j}] = \sum_{t=0}^{n_t} \left(\left(x_t^{(i)} - \frac{c^{(i)}}{n_t} \right) \left(x_t^{(j)} - \frac{c^{(j)}}{n_t} \right) \right)$$

- One time task during training

k autoencoder

Anomaly Detector



Algorithm 5: The execution algorithm for *KitNET*.

procedure: $\texttt{execute}(L^{(1)}, L^{(2)}, \mathbf{v})$
$\quad$ `// Execute Ensemble Layer`
$\vec{z} \leftarrow \text{zeros}(k)$ $\quad\triangleright$ *init input for* $L^{(2)}$
for $(\theta_i \ in \ L^{(1)})$ **do**
$\quad \vec{v}_i' = \texttt{norm}_{0-1}(\vec{v}_i)$
$\quad A_i, \vec{y_i} \leftarrow h_{\theta_i}(\vec{v}_i')$ $\quad\triangleright$ *forward propagation*
$\quad \vec{z}[i] \leftarrow \texttt{RMSE}(\vec{v_i}', \vec{y_i})$ $\quad\triangleright$ *set error signal*
end
`// Execute Output Layer`
$\vec{z}' = \texttt{norm}_{0-1}(\vec{z})$
$A_0, \vec{y_0} \leftarrow h_{\theta_0}(\vec{z_1}')$ $\quad\triangleright$ *forward propagation*
return $\leftarrow \texttt{RMSE}(\vec{z}', \vec{y_0})$

Thresholding to raise alerts

$O(n^2)$

K autoencoders → no more than n

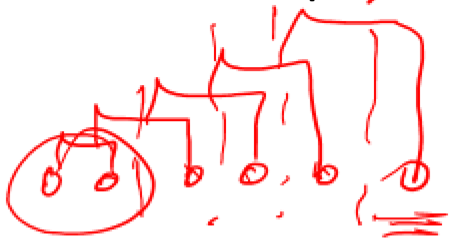
& ↑ final autoencoder output

Computation Complexity

$O(km^2 + k^2) = O(k^2)$

- What is the total computation complexity?
  - Computation complexity of ensembles + computation complexity of output layer

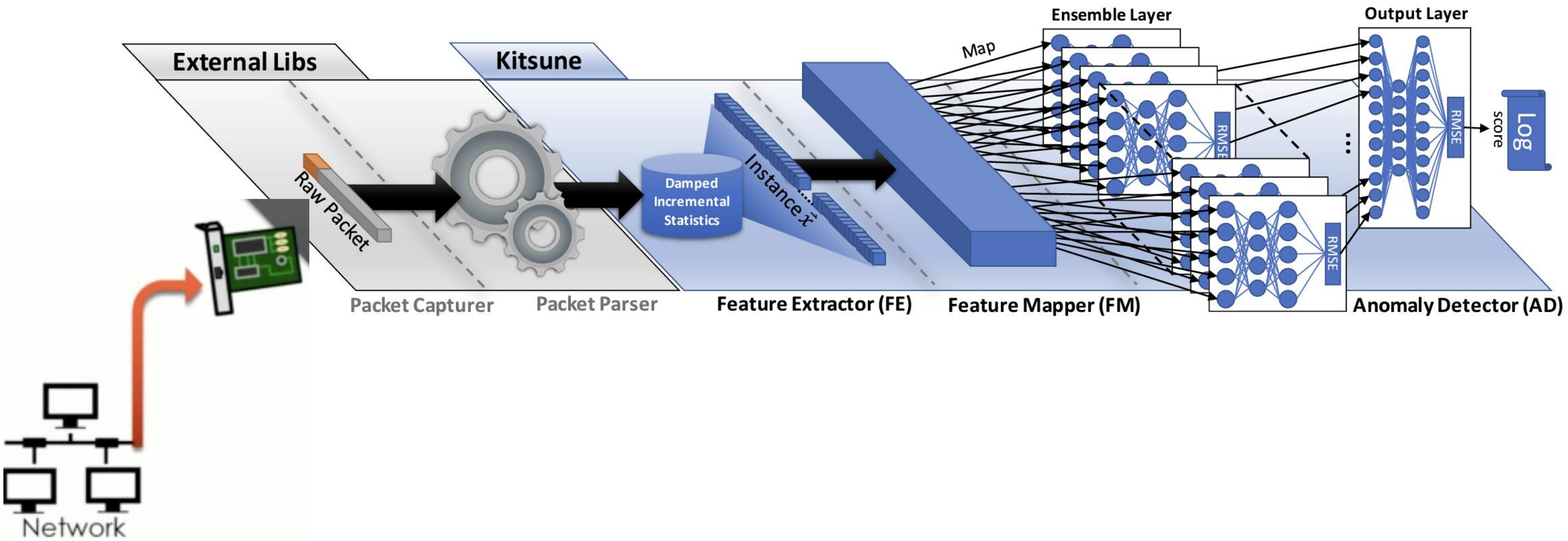
$m>2$



$$O(km^2 + k^2) = O(k^2)$$

- What is the worst-case complexity?
  - Will not likely occur

Kitsune Architecture



① online n...
② anomaly detection

Discussion

- What was the most interesting thing for you in the Kitsune paper?

ML modeling } accuracy

- Compare the two papers: similarities and differences ?

system cost
explainability

accuracy

- State of machine learning for network security

Needle in the haystack

Next Class

- Bring your laptop

- Final learning task: Resource allocation
  - Paper 1: [Neural Adaptive Video Streaming with Pensieve]
  - Paper 2: [A Deep Reinforcement Learning Perspective on Internet Congestion Control]

- Read the papers before next class

- Assignment 1 will be released tonight
  - Work in pair
  - All assignments/projects will be in pair

- Discuss project topics in next class