Aalto University

Security & Machine Learning

MSS – Lecture 7
Samuel Marchal, Mika Juuti

What will you learn

- **Basics concepts of machine learning**
- **When and how to use machine learning for security**
- **Threats against machine learning systems**

Schedule

- **Introduction to Machine Learning (ML)** **[30 min]**
  - Data and ML
  - Supervised learning
  - Challenges in ML applications
- **Machine Learning for Security** **[30 min]**
  - General concept & Anomaly detection
  - Use case: phishing detection (Off-the-Hook)
- **Specificity of ML for security applications** **[25 min]**
  - Pitfalls
  - Recommendations
- **ML in the presence of adversaries** **[15 min]**

Data and Machine Learning

Machine learning

Machine learning (ML) is a subfield of AI that deals with teaching computer systems to do infer decisions based on data:

1. Is the skin tumor malicious or benign?
2. What animal is in the picture?
3. How do you represent this English sentence in french?

”Field of study which gives computers the ability to learn without being explicitly programmed.”[1]

[1] Arthur Samuel, 1959

Representing data

Finding a suitable representation of data

How would you represent the following sentence?

Alice bought 5 apples

Representing data (base)

Finding a suitable representation of data

How would you represent the following sentence?

Alice bought 5 apples

- Word-level (>5000 symbols):
  - [Alice, bought, 5, apples]
- Character-level (~100 symbols):
  - [A,l,i,c,e, ,b,o,...,p,p,l,e,s]
- Part-of-speech tags (~20 symb.):
  - [Noun, verb, count, noun]
- Word2vec embedding

Representing data (n-grams / shingles)

Finding a suitable representation of data

How would you represent the following sentence?

Alice bought 5 apples

Unigrams:

[Alice, bought, 5, apples]

Bigrams:

[(Alice,bought),
(bought,5),
(5,apples)]

Representing data (attributes)

Finding a suitable representation of data

How would you represent the following sentence?

Alice bought 5 apples

Sentence length = 4

Sentiment = neutral

Number of verbs = 1

Number of dots = 0

Number of adjectives = 0

Matrix representation of data

Let’s take the attribute-based features in previous page. Group sentences as rows, attributes/features/variables in columns, e.g.

Alice bought 5 apples →

Bob bought 6 apples →

David loves red apples →

Feat. 1	Feat. 2	Feat. 3	Feat. 4	Feat. 5
4	2	1	0	0
4	2	1	0	0
4	2	1	0	1

Is this a good representation of data?

Choosing the right representation

- **The right representation is always context-dependent**
  - A set of features that do well in predicting party affiliation might do horrible at predicting age
- **Typical evaluation metric: *performance on test set***

Supervised learning of information

Supervised learning

Algorithmically finding meaningful structure in data $\boldsymbol{X}$ that explains some target variable y

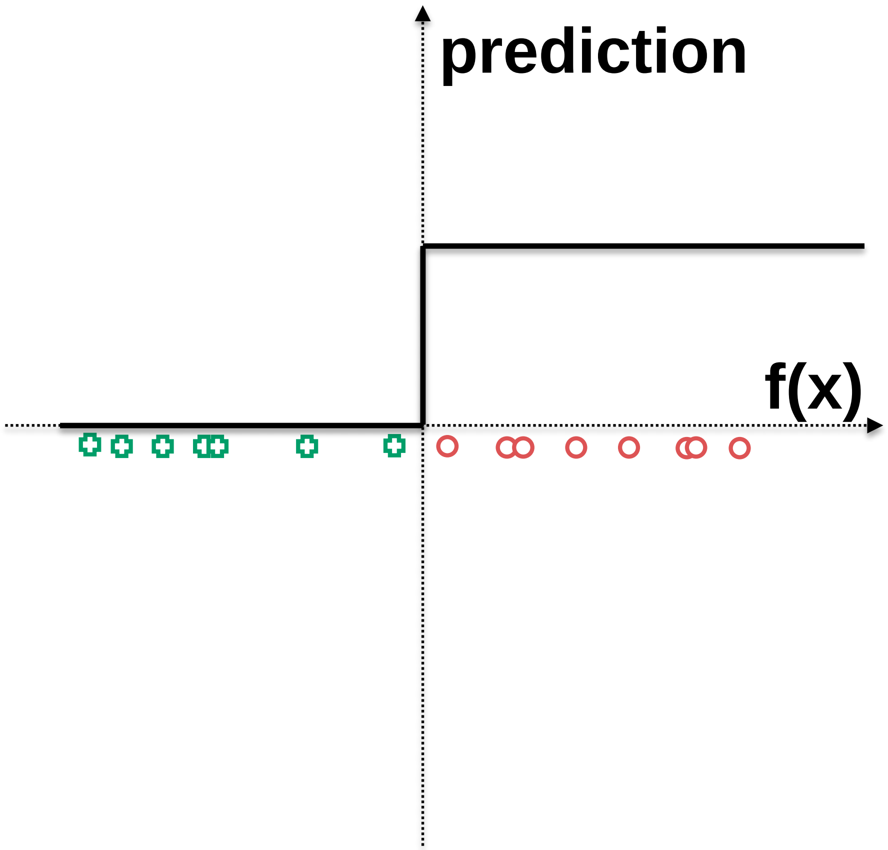
$$\mathrm{f}(\boldsymbol{X}) \approx y$$

1) **Target variable continuous: regression**
   a) Linear regression, decision trees etc
2) **Target variable categorical: classification**
   a) Typically ***easier*** to do than regression in high-dimensional space
   b) Generally the ***more classes*** (*unique values in y*), the ***more difficult*** it is

Classification

In two-class classification, the purpose is to find a function *f* that separates the data X such that

1) f(X) > 0 → 1 ○
2) f(X) < 0 → 0 ✙



Confusion matrix (spam classification)

		Predicted class		
		REAL	**SPAM1**	**SPAM2**
True class	**REAL**	5000	550	750
	SPAM1	500	500	250
	SPAM2	1000	450	500

Accuracy (spam classification)

Predicted class

True class

	REAL	SPAM1	SPAM2
REAL	5000	550	750
SPAM1	500	500	250
SPAM2	1000	450	500

$$accuracy = \frac{diagonal\ items}{all\ elements}$$

$$accuracy = \frac{6000}{9500} \approx 63.2\%$$

Is this good?

Can you think of a way to achieve better accuracy by changing class labels?

Predicted

True

	REAL	SPAM
REAL	5000	1300
SPAM	1500	1700

$$accuracy = \frac{6700}{9500} \approx 70.5\%$$

How can we do better than accuracy?

Predicted class

True class

	REAL	SPAM1	SPAM2	Σ	recall
REAL	**5000**	**550**	**750**	**6300**	**0.79**
SPAM1	500	500	250	1250	
SPAM2	1000	450	500	1950	

1) **True positive rate**
   - How often real class is recognized correctly

$$True\ positive\ rate = recall$$

$$TPR = \frac{tp}{tp + fn} = \frac{tp}{detection}$$

How can we do better than accuracy?

Predicted class

True class

	REAL	SPAM1	SPAM2	Σ	recall
REAL	5000	550	750	6300	0.79
SPAM1	500	500	250	1250	0.40
SPAM2	1000	450	500	1950	0.26

1) **True positive rate**
   - How often real class is recognized correctly

$$True\ positive\ rate = recall$$

$$TPR = \frac{tp}{tp + fn} = \frac{tp}{detection}$$

How can we do better than accuracy?

Predicted class

True class

	REAL	**SPAM1**	**SPAM2**	**Σ**	**recall**
REAL	5000	550	750	6300	0.79
SPAM1	500	500	250	1250	0.40
SPAM2	1000	450	500	1950	0.26
Σ	6500				
Precis.	**0.77**				

1) **True positive rate**
   - How often real class is recognized correctly
2) **Precision**
   - How often predicted class was real class

$$precision = \frac{tp}{tp + fp} = \frac{tp}{PPV}$$

How can we do better than accuracy?

		Predicted class				
		REAL	**SPAM1**	**SPAM2**	**Σ**	**recall**
True class	**REAL**	5000	550	750	6300	0.79
	SPAM1	500	500	250	1250	0.40
	SPAM2	1000	450	500	1950	0.26
	Σ	6500	1500	1500		
	Precis.	**0.77**	**0.33**	**0.33**		

1) **True positive rate**
   - How often real class is recognized correctly
2) **Precision**
   - How often predicted class was real class

$$precision = \frac{tp}{tp + fp} = \frac{tp}{PPV}$$

How can we do better than accuracy?

Predicted class

True class

	REAL	SPAM1	SPAM2	Σ	recall
REAL	5000	550	750	6300	**0.79**
SPAM1	500	500	250	1250	**0.40**
SPAM2	1000	450	500	1950	**0.26**
Σ	6500	1500	1500		
Precis.	**0.77**	**0.33**	**0.33**		
Fscore	**0.78**	**0.36**	**0.29**		

$$Fscore = \frac{2 \cdot (prec \cdot rec)}{prec + rec}$$

1) **True positive rate**
   - How often real class is recognized correctly
2) **Precision**
   - How often predicted class was real class
3) **F-score**
   - Harmonic mean of the two above
     - Low if either one is low
     - High if both are high

Class imbalance affects accuracy

		Predicted class		
		REAL	**SPAM1**	**SPAM2**
True class	**REAL**	5000	550	750
	SPAM1	500	500	250
	SPAM2	1000	450	500

If dataset imbalanced:

A) Upsample/downsample training and test data

OR

B) Use metrics that do not depend on dataset imbalances (ROC curve)

Always check that classes roughly equally sized before reporting accuracy.

Challenges in ML

Challenge 1. Training/test set

- **The goal of machine learning is not to achieve good performance on the training set, but on the actual data it will be used with**
  - Any performance metric estimated on a training set can easily reach 100% accuracy
    - This can be done simply by memorizing all training data!
- **Performance needs to be evaluated on a separate test set**
  - Practically by reserving a portion of the data for evaluation
- **Ideally, testing should be done with off-training set data**
  - Better estimate for future performance

No free lunch theorem [1]

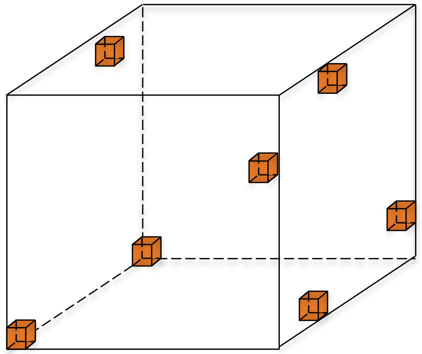
Alpaydin 2010 [2]:

”as stated by the No Free Lunch theorem (Wolpert 1995), ***there is no such thing as the ”best” learning algorithm.*** *For any learning algorithm, there is a dataset where it is very accurate and another dataset where it is very poor.* ***When we say that a learning algorithm is good, we only quantify how well its inductive bias matches the properties of the data.****”*

[1] Wolpert, David (1995), "The Lack of *A Priori* Distinctions between Learning Algorithms", *Neural Computation*, pp. 1341-1390.
[2] Alpaydin, Ethem (2010), “Introduction to Machine Learning”, *MIT press*

Challenge 2. Curse of dimensionality

- **Consider D-dimensional cube in $[0,1]^D$ with N non-overlapping data points**
  - Each point occupies a small space around itself, e.g. A volume $(0.1)^D$



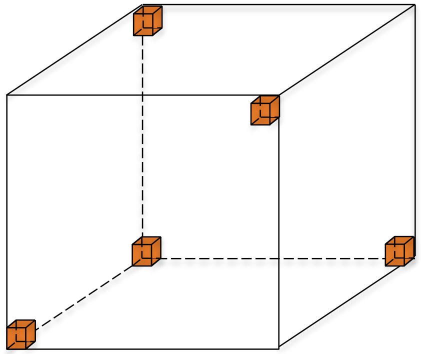
- **Occupied space volume:**
  - $\boldsymbol{N \times 0.1^D}$
  - ***Occupied space*** *grows* ***linearly*** for each new added *data point*
- **Unoccupied space volume:**
  - $\boldsymbol{1 - (N \times 0.1^D)}$
  - ***Unoccupied space*** *grows* ***exponentially*** with each added *feature*

E.g. optimal coverage of data points:

- 100 points in 2D: 100 % coverage
- 100 points in 3D: 10 % coverage
- ...
- 100 points 10D: 0.000001% coverage

Curse of dimensionality (sparse data)

- **Consider D-dimensional cube in $[0,1]^D$ with N non-overlapping data points**
  - Each point occupies a small space around itself, e.g. A volume $(0.1)^D$



- **If data is sparse, problems are more profound**
  - Euclidean point-to-point distances are quantized, possible values are:
    - $[0, \sqrt{1}, \sqrt{2}, \sqrt{3}, \dots, \sqrt{D}]$
  - *Minimum distance between two points is 1, unless points superimposed*
    - *Point-to-point distances are not distributed around 0*

Demo.

100 million data points[1] in **100 dimensional unit cube**.

Q1: What is the *average* distance between points?

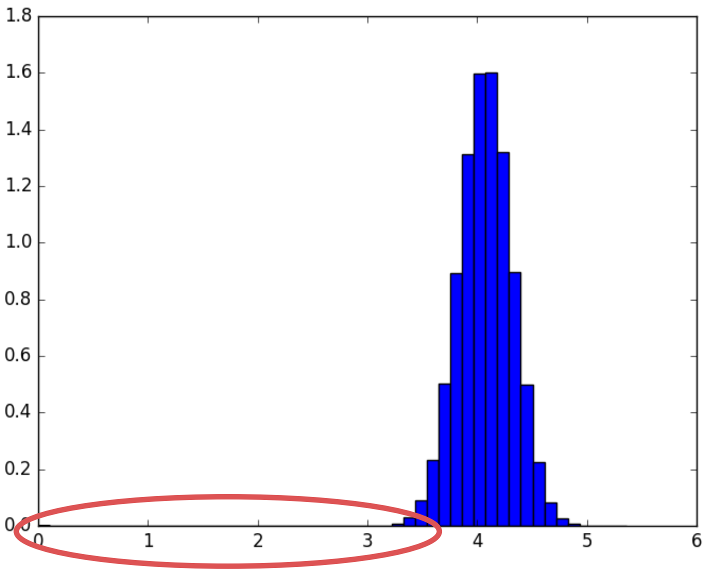
Q2: How close are points? E.g. Closest 1%?

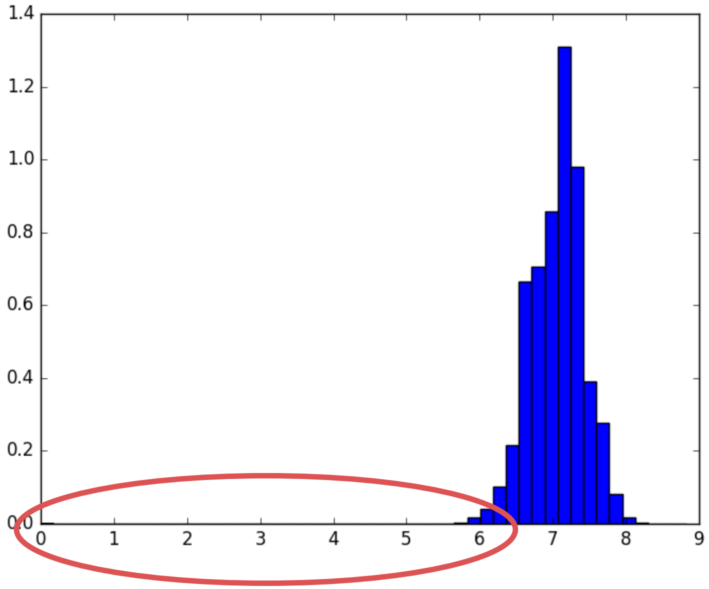
Case 1: Dense data.

- **All > 3.5!**

Case 2: Sparse data.

- **All > 5.5!**





No data is especially close to each other

1: Sampled uniformly at random

Curse of dimensionality

- **Difficult for the human mind to make sense out of (severely) multidimensional data.**
  - High-dimensional data can also result in slower execution
- **Problems becomes more profound with *sparse* data.**
  - Only a small amount of the attributes are non-zero
- **Traditional statistics relying on well-established distributions don’t work well in high-dimensional spaces.**
  - E.g. Confidence intervals based on gaussian distributions

Representing data

Sometimes there is no clear way to represent data

→ **Data representation choice often based on performance metrics**

→ This is **problematic** in **security/privacy** problems!

→ **Prefer low-dimensional and dense data whenever possible!**

Machine Learning for Security

Why using ML to secure systems ?

Securing systems by design:

- We know attacks: threat modelling / observation
- Design the system to prevent them

Limitations:

- Attack space too broad and complex to be manually modeled
- Impact on usability (restrict functionalities)
- Maliciousness related to intent
- Attack does not target the system but e.g. users

We may not know attacks!

How to secure systems with ML ?

Use an ML algorithm to automatically detect attacks

- **Supervised learning to distinguish attack / non-attack**
  - Model non-attack → OK
  - What attacks look like ?

- **Anomaly detection to detect unknown attacks**

Securing systems using anomaly detection

- **Anomaly detection**

- **Example anomaly detection applications**
  - NIDS, HIDS, etc.

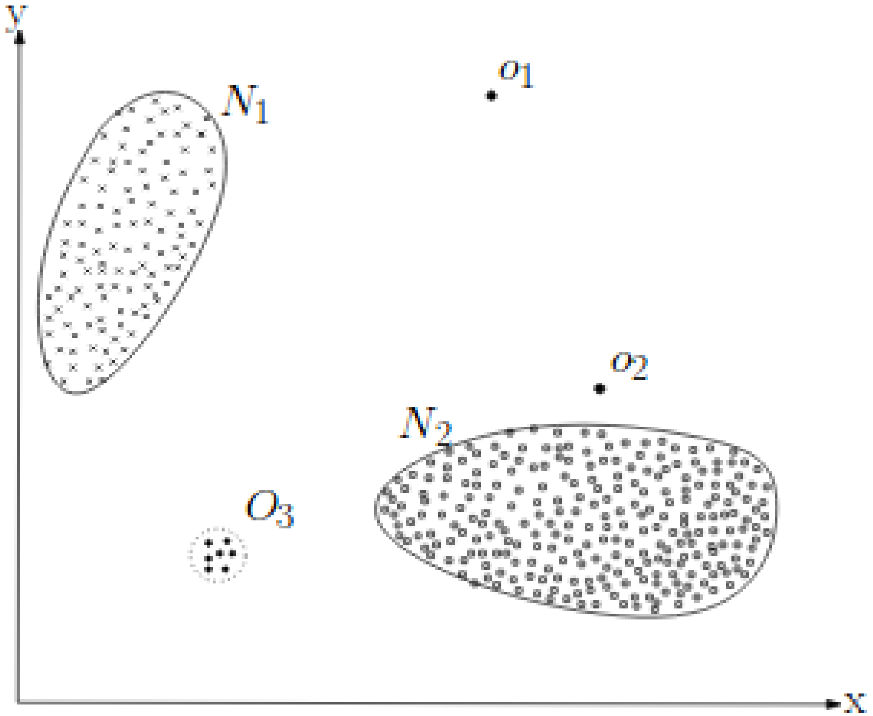
- **Supervised learning example**
  - Phishing webpage detection

What is an anomaly ?

- **Instances that do not fit a “*normal*” pattern**
  - Novelties, outliers, exceptions
  - Not necessarily malicious

- **Typically unknown pattern**

- **Sort of anomalies**
  - Global: o_1
  - Local: o_2



Typical anomaly detection

- **Define what is a “*normal*” pattern ?**
  - Domain specific
  - Can evolve over time
  - Anomalies mimic normal pattern → typical for security applications
- **Requires assumptions on normality**
- **Lack of labeled data for validation**

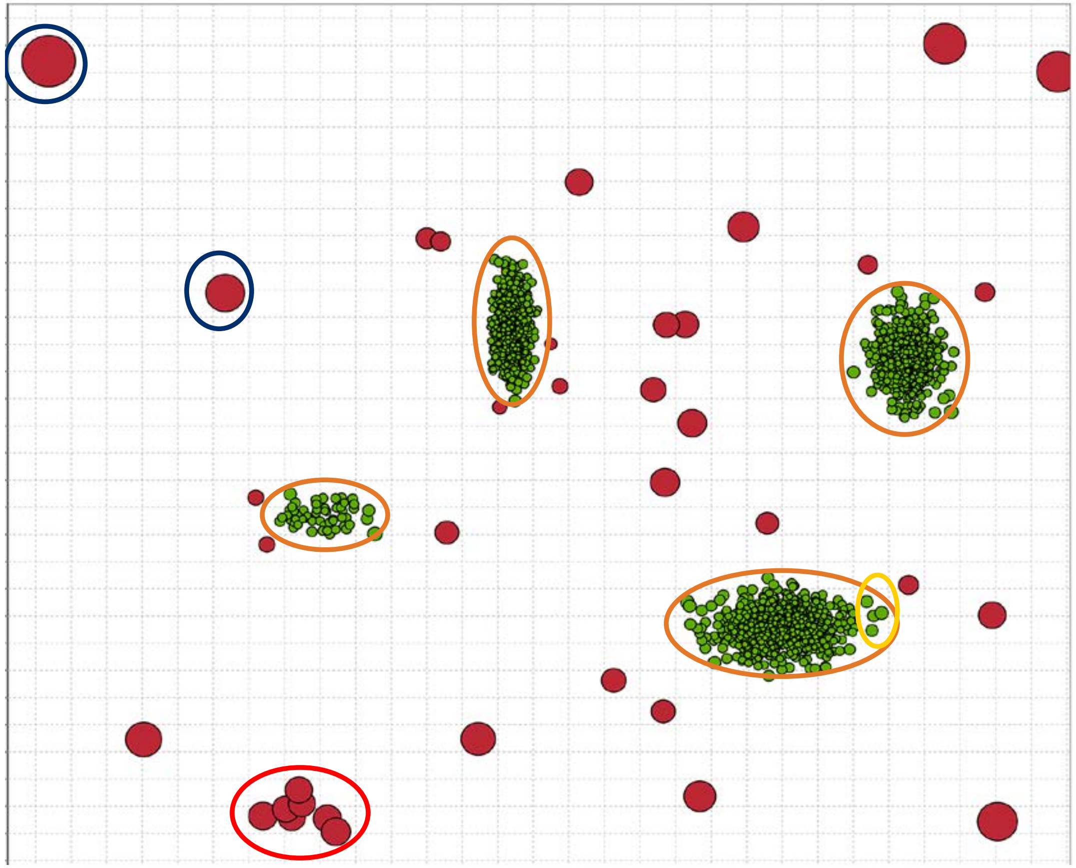
K-Nearest Neighbor (k-NN) global anomaly detection

- **Assumptions:**
  - Normal data located in *densely populated area* of the feature space
  - Anomalies are *isolated* instances that can be close to only a few other samples

- **Principle:**
  - Compute an anomaly score = average distance to k nearest neighbors
  - Set a threshold on anomaly score to get binary prediction
  - $10 < k < 50$

K-Nearest Neighbor (k-NN) global anomaly detection

- “*Normal*” data
- Global anomalies
- Group of anomalies
- Local anomalies



Network anomaly detection (Network Intrusion Detection System – NIDS)

- **Assumptions**
  - Normal/expected network traffic presents *similar characteristics* over time
  - Anomalies (i.e. signs of defect, misconfiguration, attacks) *differ from normal traffic*

- **Find a representation for these characteristics**
  - Model the normal traffic
  - Distinguish normality from anomaly
  - → *Features* and feature space

Network anomaly detection
(Network Intrusion Detection System – NIDS)

Example features for NIDS

- **Model traffic**
  - Communication protocol (HTTP/FTP/DNS)
  - IP source / destination or subnet
  - Global load, e.g. at router, per time period
  - Flow duration between two entities

- **Distinguish anomalies (use knowledge from existing attacks)**
  - Denial of Service → packet rate per IP source
  - Code injection → payload based features, e.g. n-grams

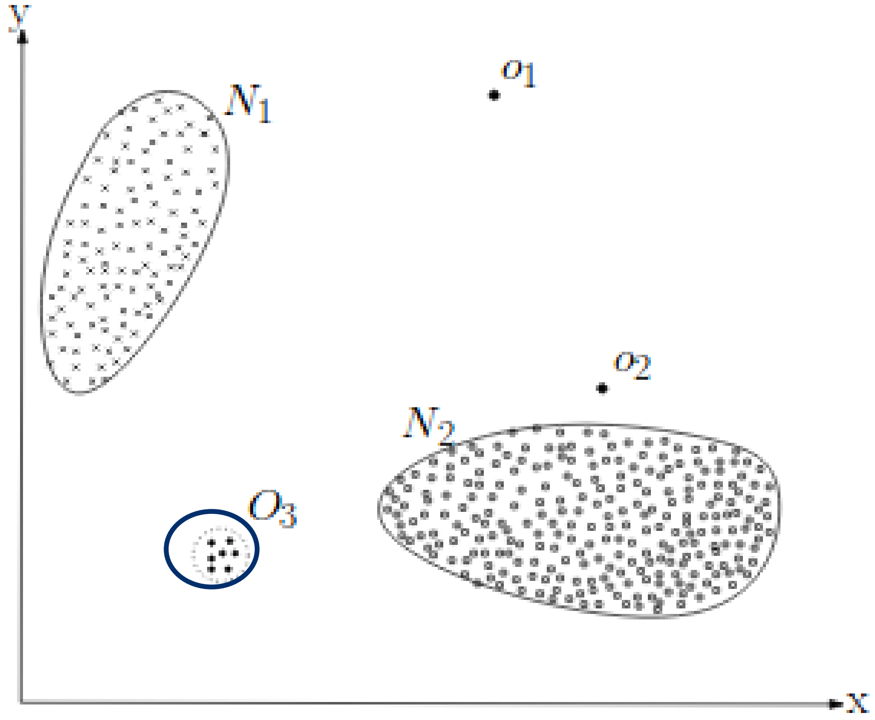
Other example applications

- **Host-based Intrusion Detection System (IDS)**
  - User ID
  - Command line invoking a process
  - File accessed
  - Previous accessed file

- **Credit card transactions (fraud detection)**
  - Retailer information, e.g. location
  - Time of transaction
  - Amount

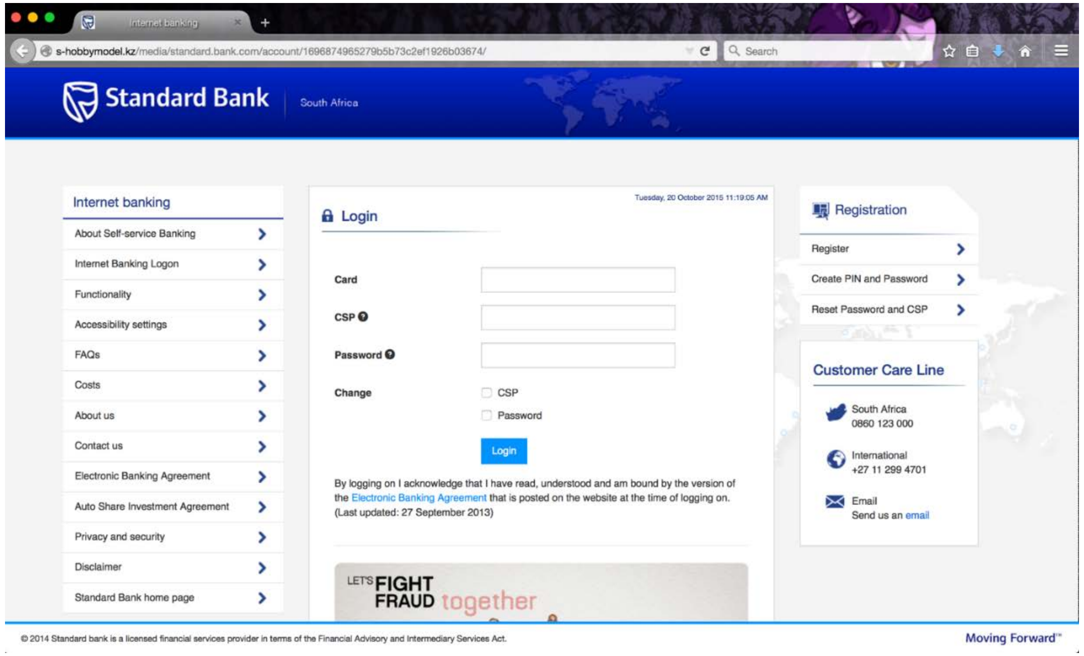
Supervised anomaly detection

- **Anomalies typically follow unknown pattern**
- **Sometimes we *know* specific anomalies for e.g. *malicious samples***
  - Follow a common pattern i.e. group of anomalies
- **We can use supervised learning to detect known anomalies**
  - 2 classes: legitimate vs. malicious



Use case: Phishing webpage detection

Identify a feature space to distinguish phishing from legitimate webpages



Internet banking

s-hobbymodel.kz/media/standard.bank.com/account/1696874965279b5b73c2ef1926b03674/

Standard Bank | South Africa

Internet banking

- About Self-service Banking
- Internet Banking Logon
- Functionality
- Accessibility settings
- FAQs
- Costs
- About us
- Contact us
- Electronic Banking Agreement
- Auto Share Investment Agreement
- Privacy and security
- Disclaimer
- Standard Bank home page

Login

Tuesday, 20 October 2015 11:19:05 AM

Card

CSP

Password

Change □ CSP □ Password

Login

By logging on I acknowledge that I have read, understood and am bound by the version of the Electronic Banking Agreement that is posted on the website at the time of logging on. (Last updated: 27 September 2013)

LET'S FIGHT FRAUD together

Registration

- Register
- Create PIN and Password
- Reset Password and CSP

Customer Care Line

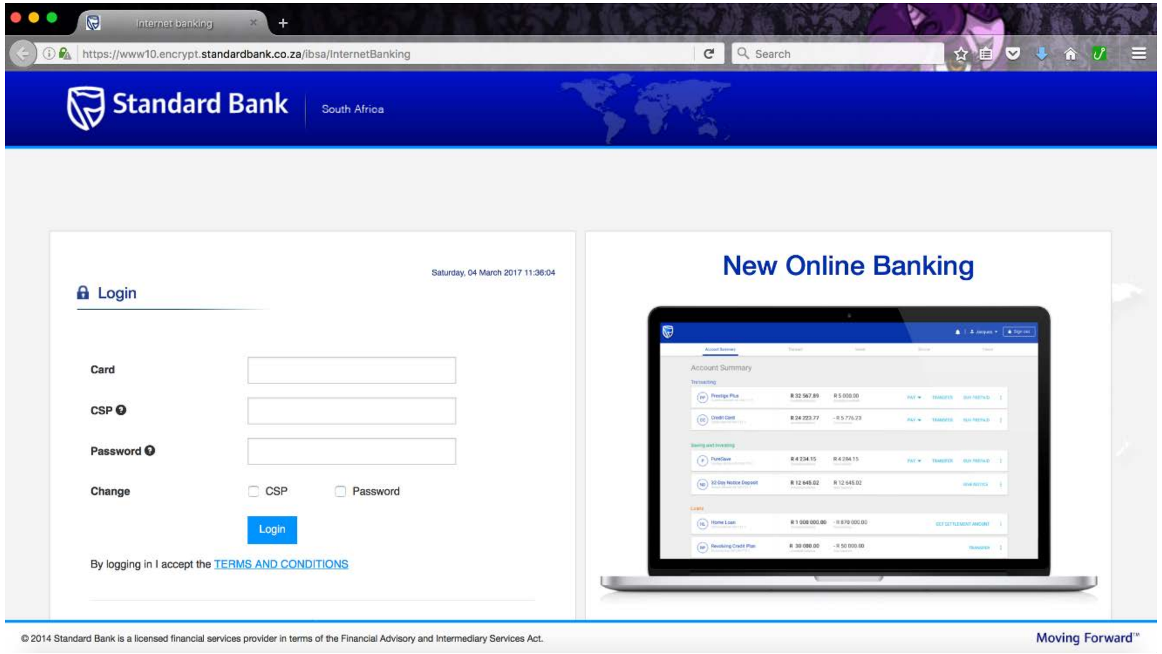
South Africa
0860 123 000

International
+27 11 299 4701

Email
Send us an email

© 2014 Standard bank is a licensed financial services provider in terms of the Financial Advisory and Intermediary Services Act.

Moving Forward™



Internet banking

https://www10.encrypt.standardbank.co.za/ibsa/InternetBanking

Standard Bank | South Africa

Login

Saturday, 04 March 2017 11:36:04

Card

CSP

Password

Change □ CSP □ Password

Login

By logging in I accept the TERMS AND CONDITIONS

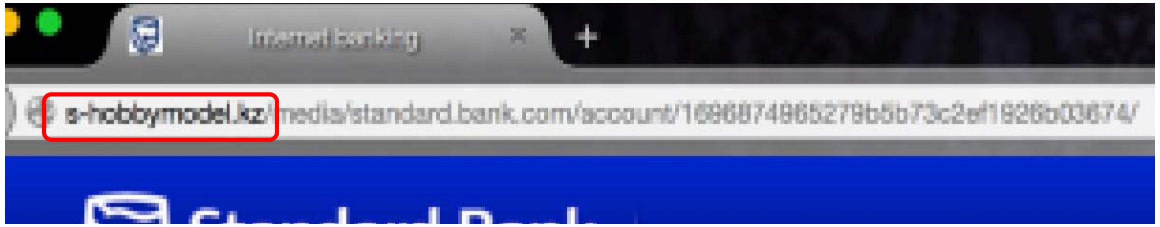
New Online Banking

© 2014 Standard Bank is a licensed financial services provider in terms of the Financial Advisory and Intermediary Services Act.

Moving Forward™

Phishing webpage (phish)

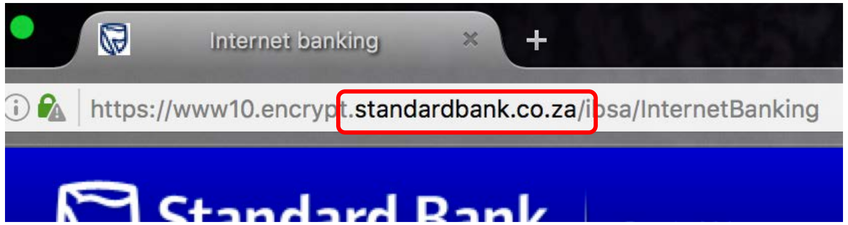
Legitimate webpage



Internet banking

s-hobbymodel.kz/media/standard.bank.com/account/1696874965279b5b73c2ef1926b03674/

Standard Bank

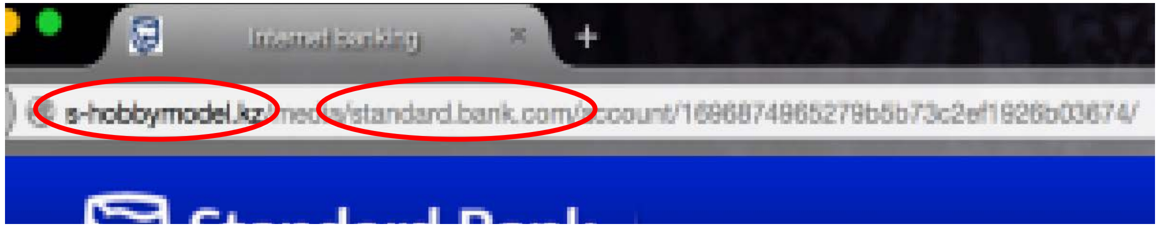


Internet banking

https://www10.encrypt.standardbank.co.za/ibsa/InternetBanking

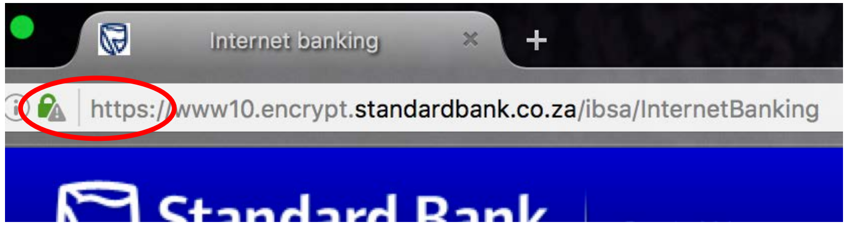
Standard Bank

Feature extraction



Internet banking
s-hobbymodel.kz/media/standard.bank.com/account/16968749865279b5b73c2ef1926b03674/
Standard Bank

Phishing webpage



Internet banking
https://www10.encrypt.standardbank.co.za/ibsa/InternetBanking
Standard Bank

Legitimate webpage

Differences in landing URL

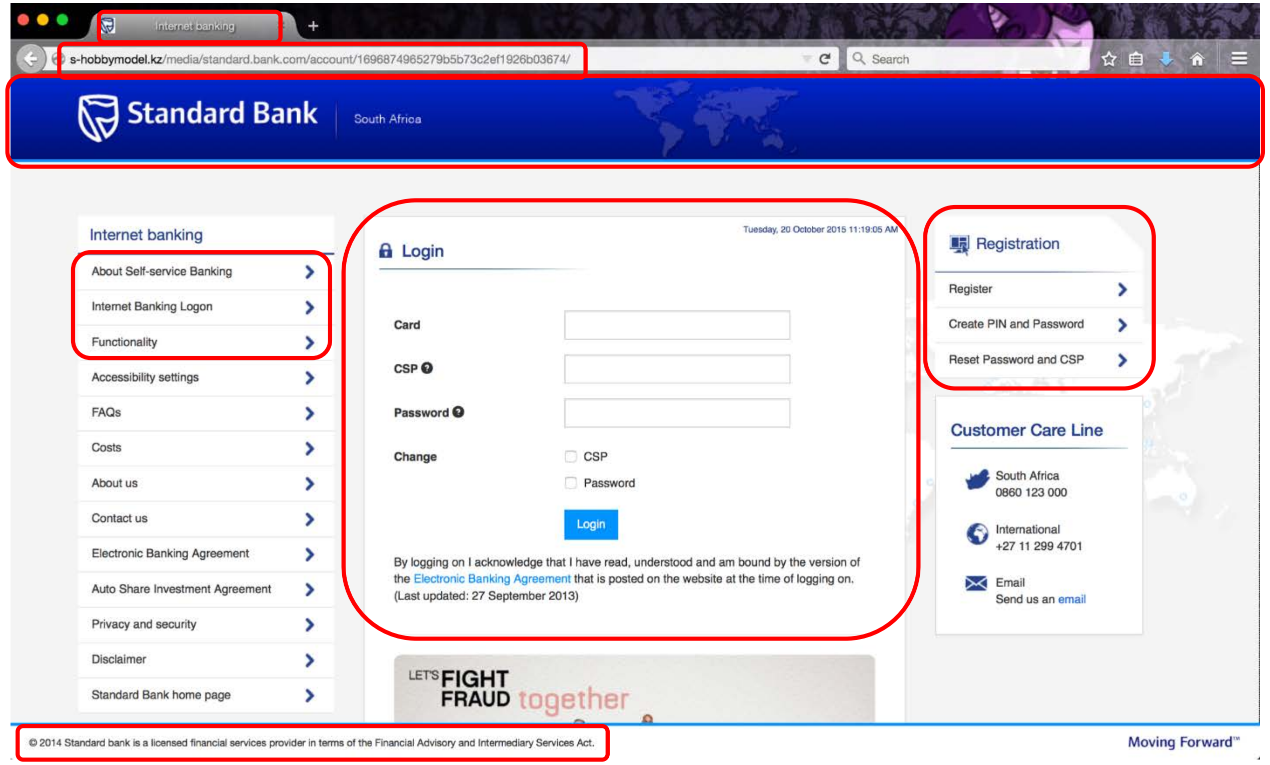
- Use of secure connection (HTTPS)
- URL slightly longer for phishing
- Use of domain name like file path
- Use name of targeted website
- Use unknown domain name
- Etc.

Potential features

- ➢ HTTP/HTTPS
- ➢ Length of URL (characters)
- ➢ Dots "." in path
- ➢ Fixed token "standard"
- ➢ Domain reputation

Feature extraction

Apply similar process to other data sources of interest[1]:



Internet banking

s-hobbymodel.kz/media/standard.bank.com/account/1696874965279b5b73c2ef1926b03674/

Search

Standard Bank | South Africa

Internet banking

- About Self-service Banking
- Internet Banking Logon
- Functionality
- Accessibility settings
- FAQs
- Costs
- About us
- Contact us
- Electronic Banking Agreement
- Auto Share Investment Agreement
- Privacy and security
- Disclaimer
- Standard Bank home page

Tuesday, 20 October 2015 11:19:05 AM

Login

Card

CSP

Password

Change □ CSP

□ Password

Login

By logging on I acknowledge that I have read, understood and am bound by the version of the Electronic Banking Agreement that is posted on the website at the time of logging on. (Last updated: 27 September 2013)

Registration

- Register
- Create PIN and Password
- Reset Password and CSP

Customer Care Line

South Africa
0860 123 000

International
+27 11 299 4701

Email
Send us an email

© 2014 Standard bank is a licensed financial services provider in terms of the Financial Advisory and Intermediary Services Act.

Moving Forward™

Starting URL

Landing URL

Redirection chain

internal

Logged links

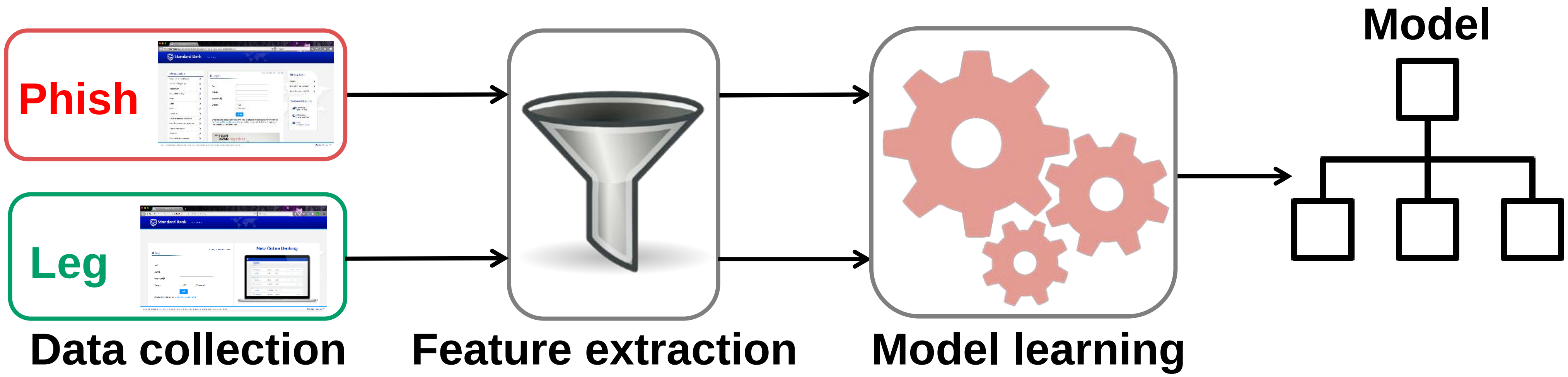
external

HTML source code:

- Text
- Title
- HREF links
- Copyright

Phishing webpage detection process

- *Collect observations* representing phishing/legitimate webpages
- *Compute* the defined *features*
- Automatically *learn differences (model)* between classes (phishing/legitimate) → Machine Learning algorithm
- *Predict* the class of unknown instances → *phishing* or *legitimate*



Phish

Leg

Data collection

Feature extraction

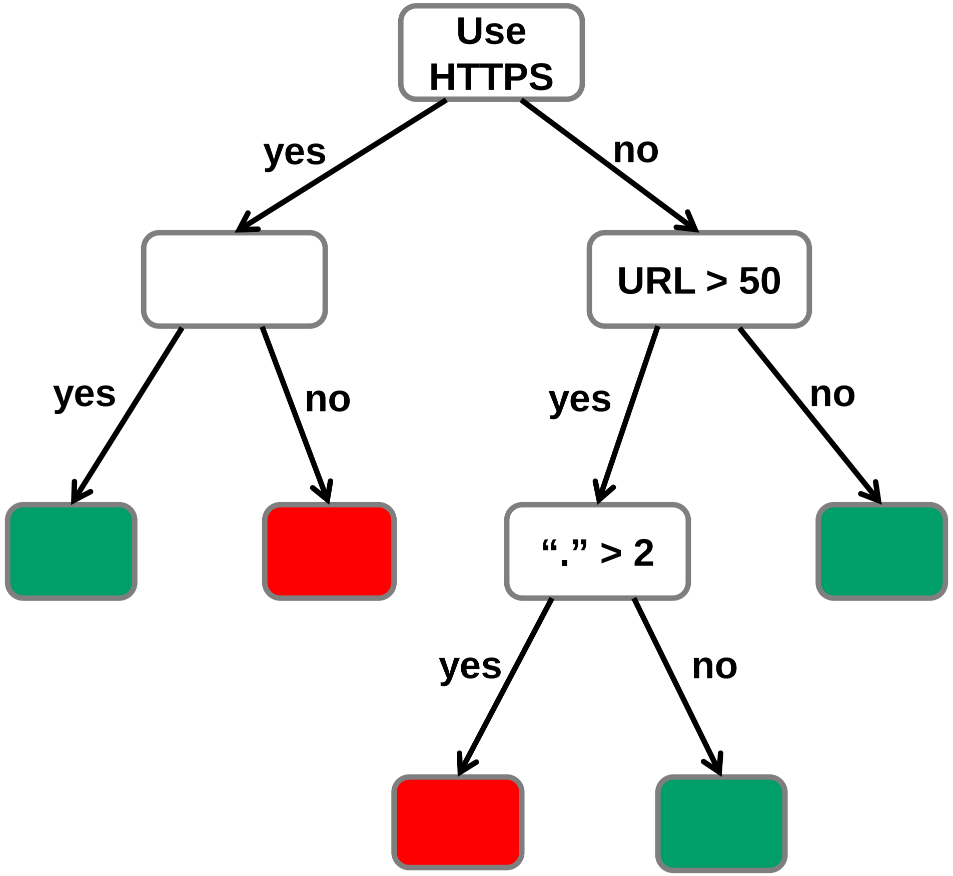
Model learning

Model

Model learning
Example: Decision tree

Group of phishs distinguishable from legitimate websites according to some feature values:

- Do not use HTTPS
- URL length greater than 50
- More than 2 “.” in their path

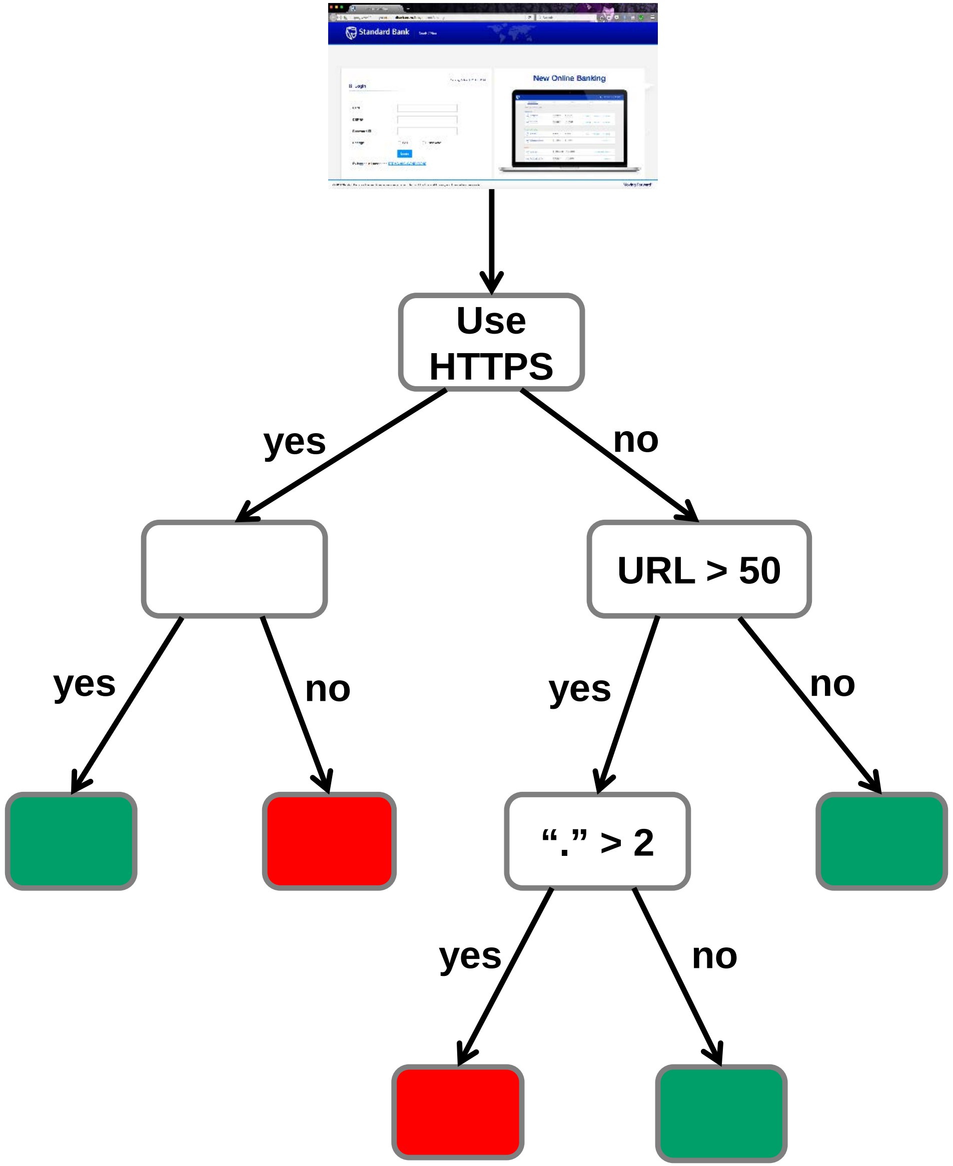


Prediction (new websites)

Features extracted and evaluated against the learned model **(decision tree)**

- Do not use HTTPS
- URL length lower than 50

→ **The website is legitimate**



Off-the Hook: Fast client-side phishing detection

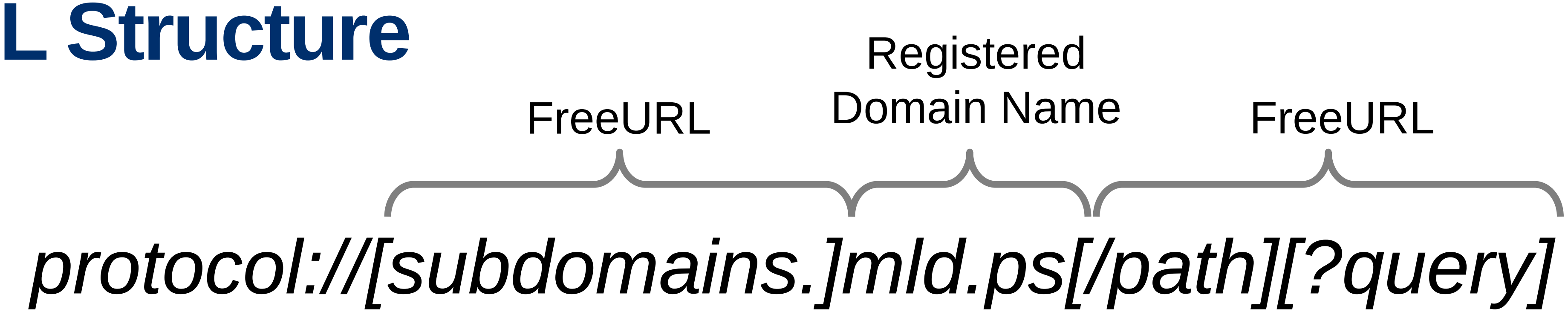
[1] **Marchal et al.**, *Off-the-Hook: An Efficient and Usable Client-Side Phishing Prevention Application*, 2017

Phisher’s control & constraints

Data sources differ in terms of the levels of

- control the phisher has over a source
- constraints placed on the phisher in manipulating that source

URL Structure



protocol://[subdomains.]mld.ps[/path][?query]

https://www.amazon.co.uk/ap/signin?_encoding=UTF8

- Protocol = *https*
- Registered domain name (RDN) = *amazon.co.uk*
- Main level domain (*mld)* = *amazon*
- FreeURL = *{www, /ap/signin?_encoding=UTF8}*

Phisher’s control & constraints

Control:

- **External** loaded content (logged links) and **external** HREF links are *usually* **not controlled** by page owner.

Constraints:

- **Registered domain name** part of URL cannot be freely defined: **constrained** by DNS registration policies.

Conjecture: Improve phish detection by modeling control/constraints

- generalizable, language independent, hard to circumvent

Data sources: control & constraints

	Unconstrained	Constrained
Controlled	Text Title Copyright Internal *FreeURL (2)*	Internal *RDNs* (2)
Uncontrolled	External *FreeURL (2)*	External *RDNs* (2)

Feature selection

A small set (212) of features computed from data sources:

- URL features (106): e.g., # of dots in *FreeURL*
- Consistency features (101)
- Webpage content (5): e.g., # of characters in *Text*

Features not data-driven: e.g., no bag-of-words features

- Conjecture: can lead to language-independence, temporal resilience

Consistency features

Term usage (66)

- strings of 3 or more characters, separated by standard delimiters

Usage of “Main level domain” *(mld)* from starting/landing URLs *(*32)

“Registered domain name” usage *(RDN)* (13)

Phishing detector with 99.9% accuracy

Pitfalls in using ML (for security)

Adversaries will circumvent detection

The ML model is intended to detect/counter attacks

Adversary *will* attempt to circumvent detection:

- poison learning process
- infer detection model
- mislead classifier

In Off-the-Hook:

- Modeling constraints and controls of phishers while training
- Adversary can control External RDNs!

→ **Resistance to adversaries**

Privacy concerns are multilateral

Data used for ML may be sensitive

- Sensitive information about users in
  - training data → model inversion, membership inference
  - prediction process → user profiling, e.g., in a cloud setting (ML-as-a-service)

In Off-the-Hook:

- Client-side classifier to avoid disclosure of URLs
- But model stealing may be a concern

➡ **Multilateral privacy guarantees**

Classification landscapes are dynamic

Attacks evolve fast

Prediction instances likely differ from training instances

- E.g., Android malware evolves due to changes in API

In Off-the-Hook:

- Avoidance of data-driven features
- Models that allow inexpensive retraining

➡ **Temporal resilience**

Maintaining labels is expensive

More training data is good; but unbalanced classes typical

Data about malicious behavior difficult to obtain

- Labeling is cumbersome, requires expertise, may be inaccurate or may evolve (e.g. phishing URLs)

In Off-the-Hook:

- Manage with small training sets
- Minimize ratio of training set size to test size

→ **Minimal training data**

Predictions need to be intelligible

Ability of humans to understand why a prediction occurs

- Detection as malicious → forensic analysis
- Explain predictions to users, e.g. why access is prevented
- “Explainability” obligations under privacy regulations like GDPR

In Off-the-Hook:

- Small set of “meaningful” features
- Use of (ensemble of) shallow decision trees

➡ **Transparent decision process**

ML failures can harm user experience

Security is usually a secondary goal

Use of ML must not negatively impact usability

- Decision process should be efficient
- Wrong predictions may have a significant usability cost

In Off-the-Hook:

- Prediction effectiveness and speed
- In phishing detection, one false positive may be one too much!

➡ Lightweight and accurate

Security/privacy applications: desiderata

Circumvention resistance

- Resistance to adversaries

Temporal resilience

- Resilience in dynamic environments

Minimality

- Use of minimal training data

Privacy

- Model privacy, training set privacy, and input/output privacy

Intelligibility

- Transparent decision process

Effectiveness

- Lightweight, accurate models

How to avoid pitfalls ?

Recommendations and good practice

- **ML Algorithm and model**
- **Feature selection**
- **Dataset selection**
  - Training
  - Testing
- **Evaluation**
  - Dataset usage
  - Evaluation metrics to consider

ML Algorithm and Model

Resilience to circumvention (or hardening the task)

- **Keep decision model and features *secret***
  - Remote / local decision process
    - Distant decision can protect model and features from users/attackers
    - Local decision can ensure user privacy
  - Model obfuscation / encryption (also input encryption)

- **Non-linear and complex model to *avoid model inference***
  - Linear regression, decision tree, shallow NN → *easy inference*
  - Ensemble methods, deep NN → *more complex but still doable*

ML Algorithm and Model

- **Algorithm to *learn* and *generalize* from *small amount* of data**
  - Consider low availability of training data
  - E.g. *SVM, deep NN typically* require lots of data for training

- **Model providing *understanding* in *decision process***
  - E.g. *tree-based methods* give understanding of features involved in decision making
  - Features need yet to be significant…
  - Depriving the decision from understanding does not serve as obfucation and *does not increase resilience to attacks (adversarial ML)*

- **Ease for retraining (attack evolution)**

Feature selection

Favor carefully hand-crafted features

- **Sensible to *distinguish* classes e.g. benign / malicious**
  - Can be *automated*: feature extraction / selection

- ***Resilient to manipulation* by attacker or costly to manipulate**
  - Automated feature selection may *discard features hard to manipulate*
  - Any ML algorithm is vulnerable to adversarial samples
  - Needs *domain expertise* and human input

- **Provide understanding for detection and misclassification**
  - Automated feature extraction may *loose semantic*

Feature selection

Features *defining the threat generally* ≠ ephemerally

- **Features *common to any kind of anomalies* to identify**
  - Capability to detect anomalies not already observed

- **Don’t consider features tied to ephemeral trends**
  - Use fixed term, e.g. *“paypal”,* for phishing detection
  - *Irrelevant to detect attacks* targeting an other brand

- **Avoid *data-driven feature* extraction (e.g. Bag-of-Words)**
  - Attacker can easily *infer and manipulate* data-driven features

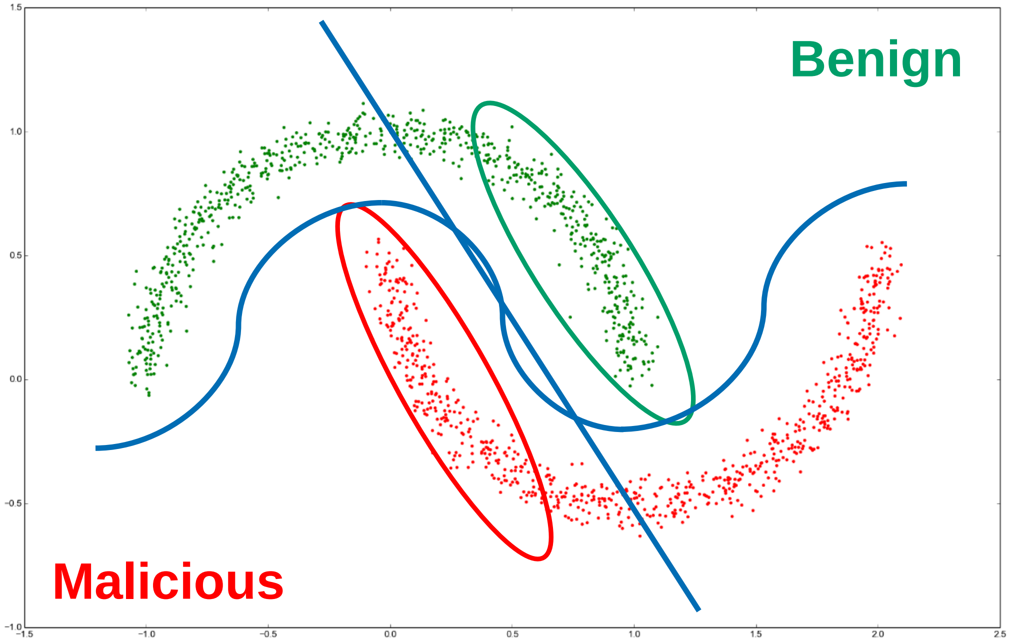
Feature selection

- **Rely on a *few features*:**
  - Reduce space of manipulation for an adversary
  - Limited availability of training data (for some class at least)
  - Good practice to *generalize a phenomenon*: 10x to 100x more training instances than features

Dataset selection

- **Representativity of the data → *unbiased dataset***
  - Coverage of the *whole attack space* / normality space
  - Avoid modeling specificities of subset of malicious / benign instances:
    - *10,000 most popular Android app*
    - *Android malware that contacts malicious domains*

- **Ensure *validity of labels***
  - Malicious samples may have *inaccurate / disagreeing labels*
    - *phishing websites, malware (VirusTotal), etc.*



Evaluation – dataset usage

Represent real-world scenario

- ***Train* the model on *old data* and *test* on *new data***
  - Avoid *cross-validation* → overestimate performance

- **Use larger testing than training set**
  - Ensures *scalability* of the approach
  - Model cannot be trained with more data than it can be applied to

Evaluation – dataset usage

- **Deal with unbalanced class problem for training**
  - Resample the class: under-sampling over-represented class
  - Generate synthetic example for the under-represented class (e.g. SMOTE)
  - Use penalized models (e.g. penalized-SVM)

- **Represent *real-world distribution* for testing**
  - Anomalies << normal instances (e.g. phishs << legitimate websites)
  - Preserve repartition for relevant accuracy results from evaluation

Evaluation – metrics to consider

Imbalanced class consideration

- **Alcohol test balloon**
  - Accuracy = 99%
  - True negative rate = 99%
  - True positive rate = 95%
  - False positive rate = 1%
  - False negative rate = 5%
- **Actual driver repartition:**
  - 1 drunk / 99 sober
  - 1% drive drunk

- **Result from test on 2000 drivers:**
  - 19 positives out of 20 actually drunk (True positives)
  - 20 positives out of 1980 not drunk (False positives)
  - If I got controlled positive, my chance of being actually drunk:

Precision = 19 / (19+20) = **49%**

Evaluation – metrics to consider

- **Considering unbalanced class (e.g. 100/1) some metrics are not sufficient to assess the accuracy of a system**
  - Accuracy, AUC, TPRate, etc. can be high while having a *poor classification system* for security applications

- **Metrics to present in anomaly detection**
  - Recall (TP_{rate}) → detection capability: $Recall = \frac{TP}{TP+FN}$
  - Precision → reliability / unnecessary annoyance: $Precision = \frac{TP}{TP+FP}$
  - F-measure → harmonic mean of previous: $F = 2 \times \frac{precision \times recall}{precision + recall}$

What about Deep Learning ?

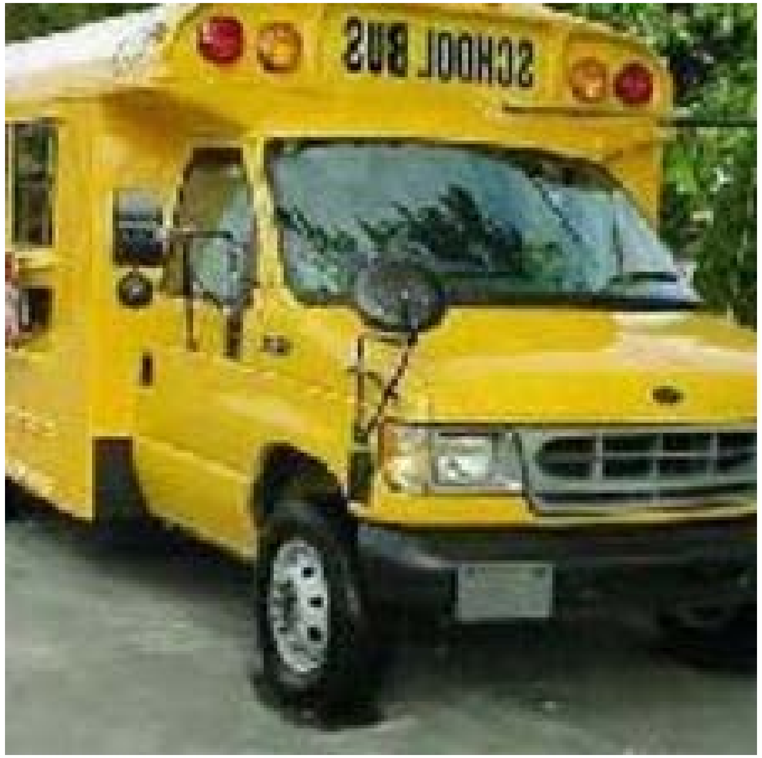
- **Typically require *large amount of data* to be trained accurately**
  - Problem with availability of labeled data

- **Automated feature extraction/selection**
  - Features *easily manipulated by attackers* → ease for circumvention

- **Complicated decision process**
  - Lack for *accountability in decision*

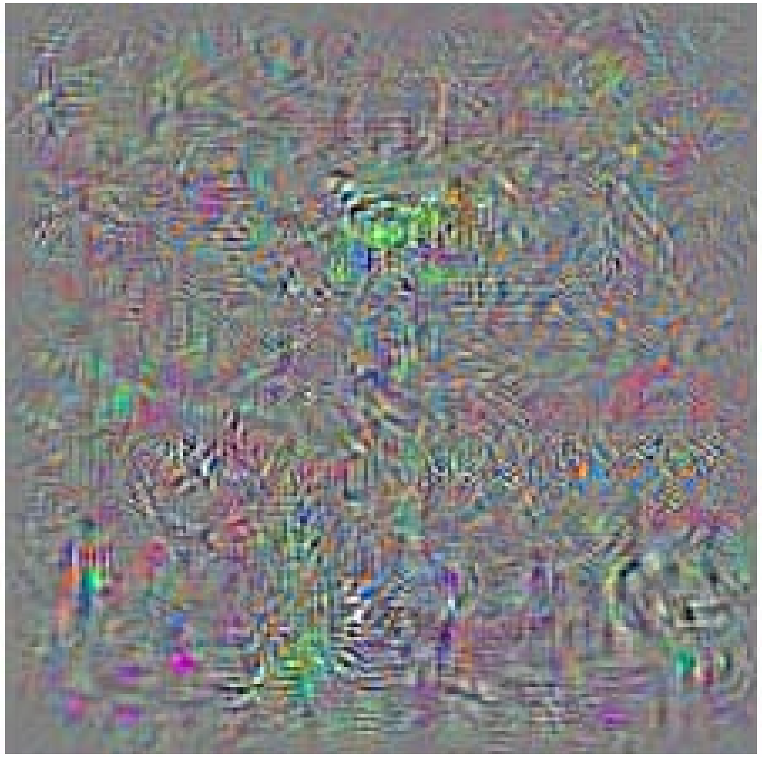
- **Relearning is usually costly**
  - High need considering fast evolution in attacks

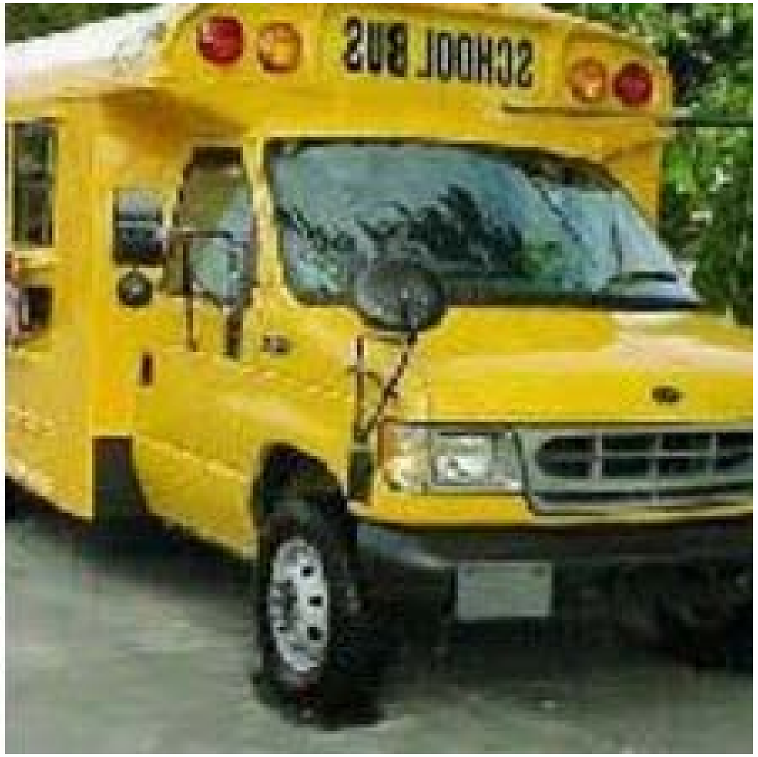
Machine Learning in the presence of adversaries



Which class is this?

School bus

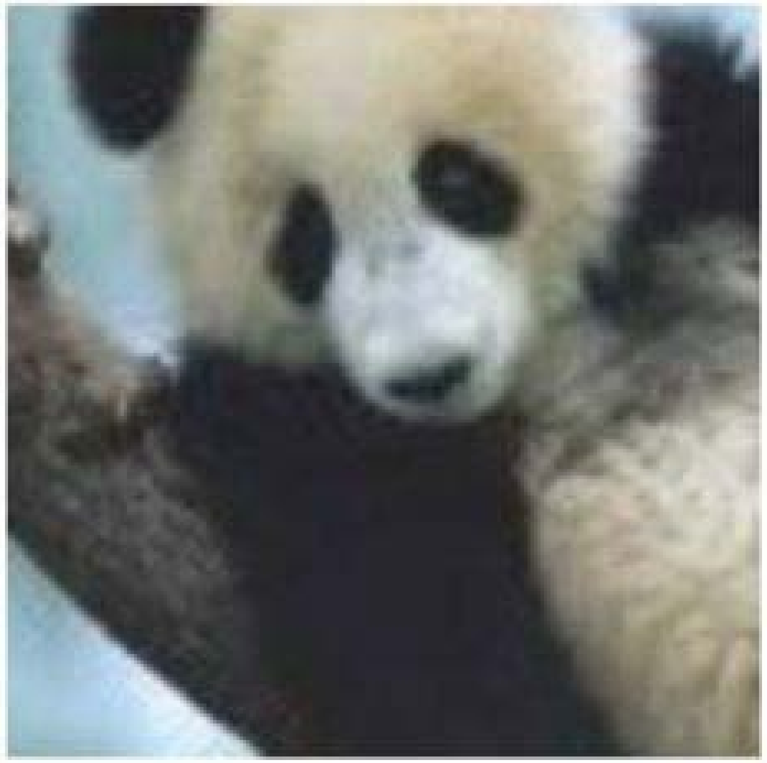




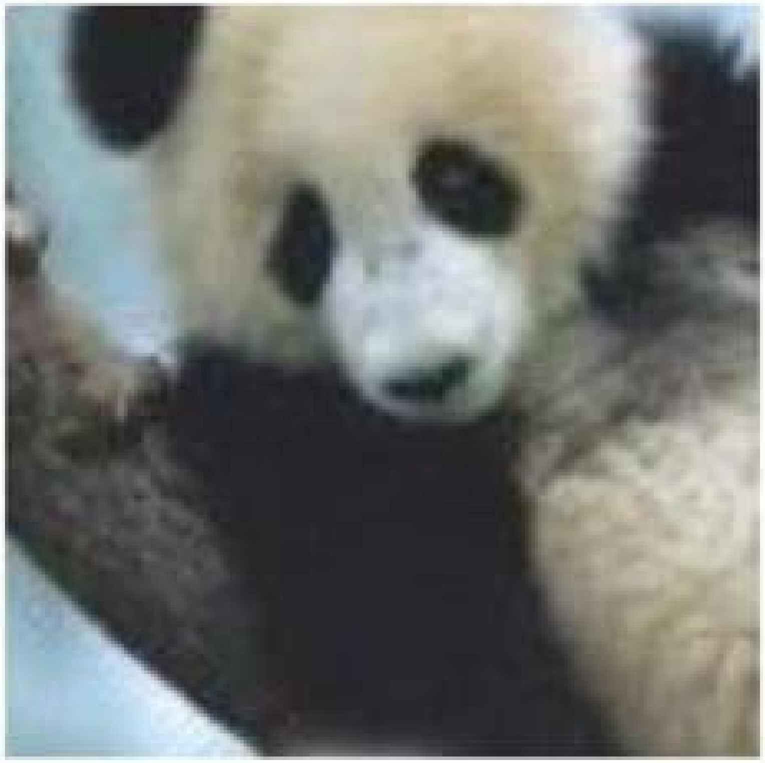
Which class is this?

Ostrich

Szegedy et al. “Intriguing Properties of Neural Networks” 2014
https://arxiv.org/abs/1312.6199v4

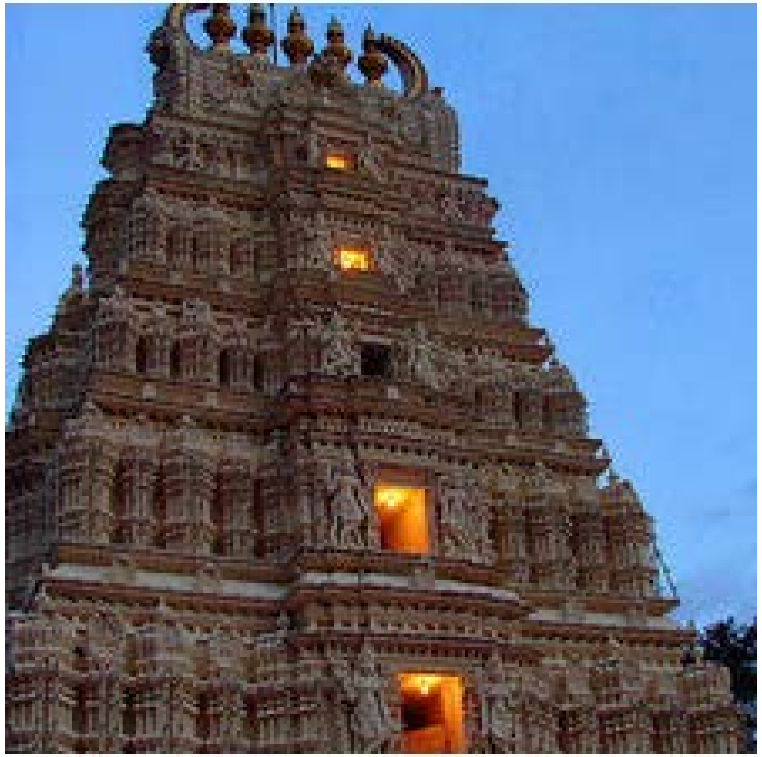


Which class is this?
Panda

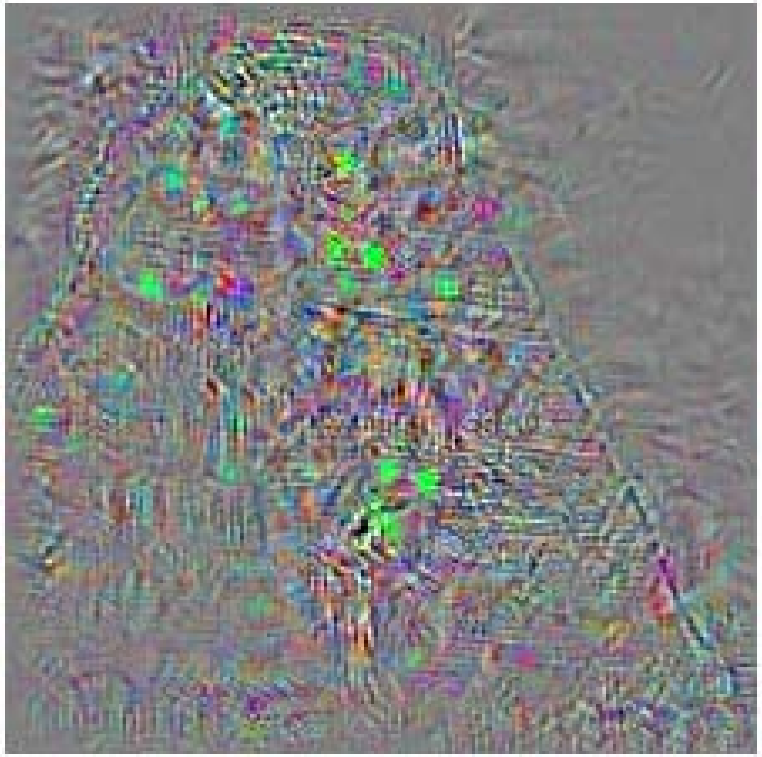


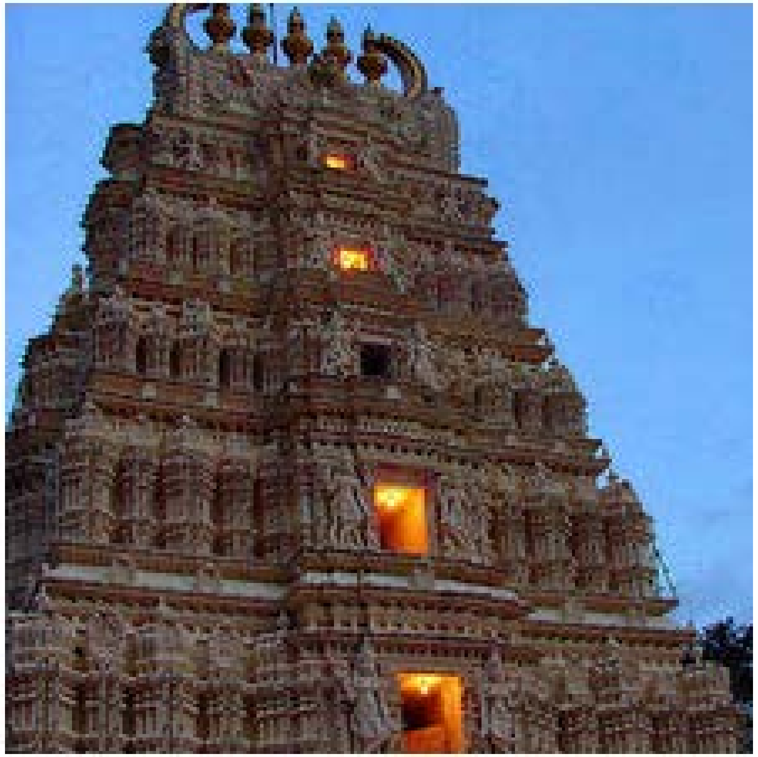
Which class is this?
Gibbon

Goodfellow et al. “Explaining and Harnessing Adversarial Examples” 2015
https://blog.openai.com/robust-adversarial-inputs/



Which class is this?
Building





Which class is this?
Ostrich

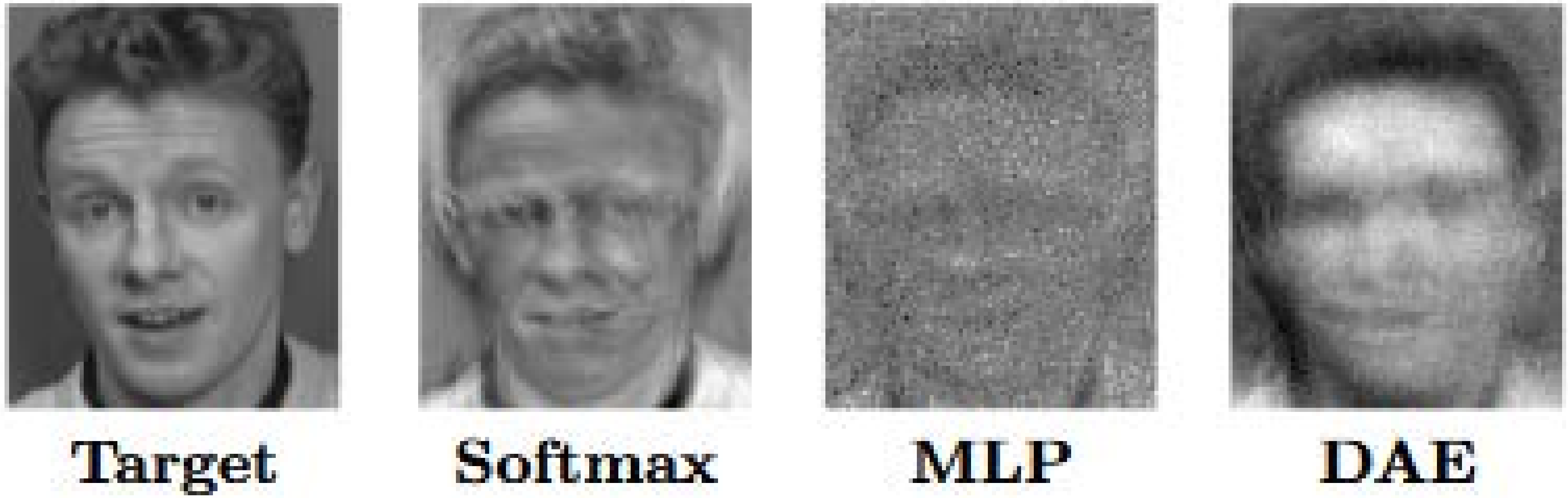
Szegedy et al. “Intriguing Properties of Neural Networks” 2014
https://arxiv.org/abs/1312.6199v4

Which class is this?

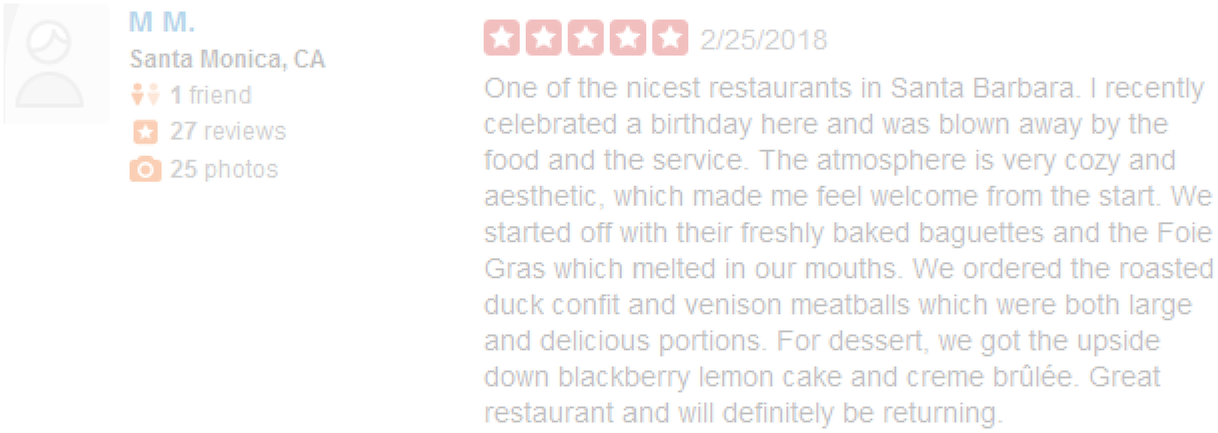
Cat

Athalye et al. “Synthesizing Robust Adve
https://blog.openai.com/robust-adversari

Zhang et al, “DolphinAttack: Inaudible Voice Commands”, ACM CCS 2017 https://arxiv.org/abs/1708.09537



Fredrikson et al. ”Model Inversion Attacks that Exploit Confidence Information and Basic Countermeasures”, CCS’15.
https://www.cs.cmu.edu/~mfredrik/papers/fjr2015ccs.pdf

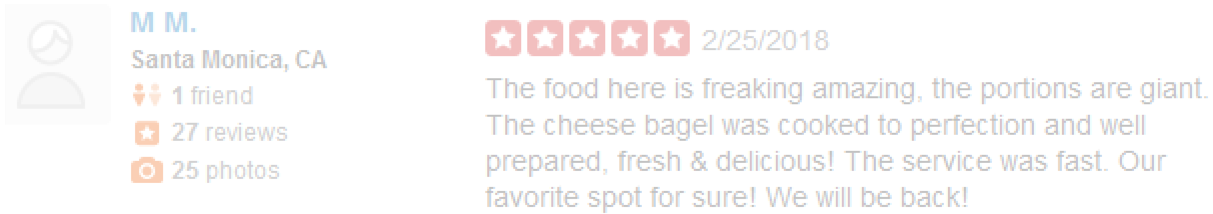


M M.
Santa Monica, CA
1 friend
27 reviews
25 photos

★★★★★ 2/25/2018

One of the nicest restaurants in Santa Barbara. I recently celebrated a birthday here and was blown away by the food and the service. The atmosphere is very cozy and aesthetic, which made me feel welcome from the start. We started off with their freshly baked baguettes and the Foie Gras which melted in our mouths. We ordered the roasted duck confit and venison meatballs which were both large and delicious portions. For dessert, we got the upside down blackberry lemon cake and creme brûlée. Great restaurant and will definitely be returning.

Human

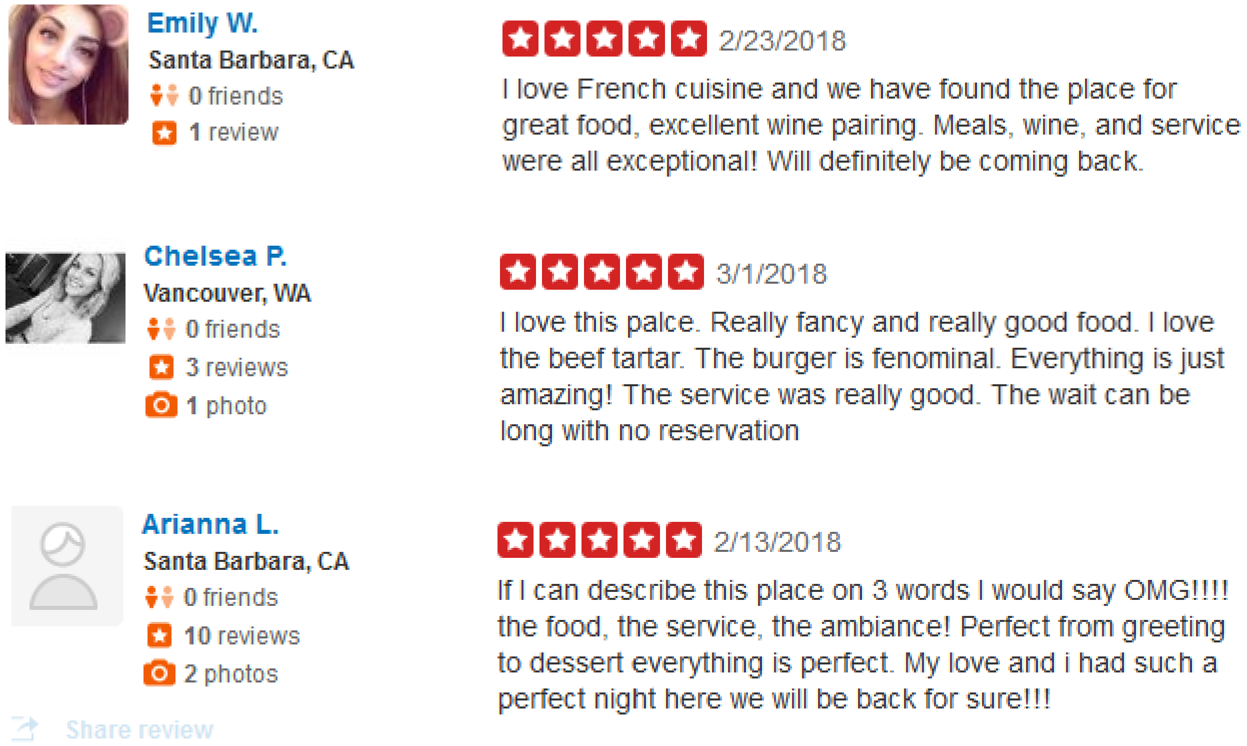


M M.
Santa Monica, CA
1 friend
27 reviews
25 photos

★★★★★ 2/25/2018

The food here is freaking amazing, the portions are giant. The cheese bagel was cooked to perfection and well prepared, fresh & delicious! The service was fast. Our favorite spot for sure! We will be back!

Machine-generated



Emily W.
Santa Barbara, CA
0 friends
1 review

★★★★★ 2/23/2018

I love French cuisine and we have found the place for great food, excellent wine pairing. Meals, wine, and service were all exceptional! Will definitely be coming back.

Chelsea P.
Vancouver, WA
0 friends
3 reviews
1 photo

★★★★★ 3/1/2018

I love this palce. Really fancy and really good food. I love the beef tartar. The burger is fenominal. Everything is just amazing! The service was really good. The wait can be long with no reservation

Arianna L.
Santa Barbara, CA
0 friends
10 reviews
2 photos

Share review

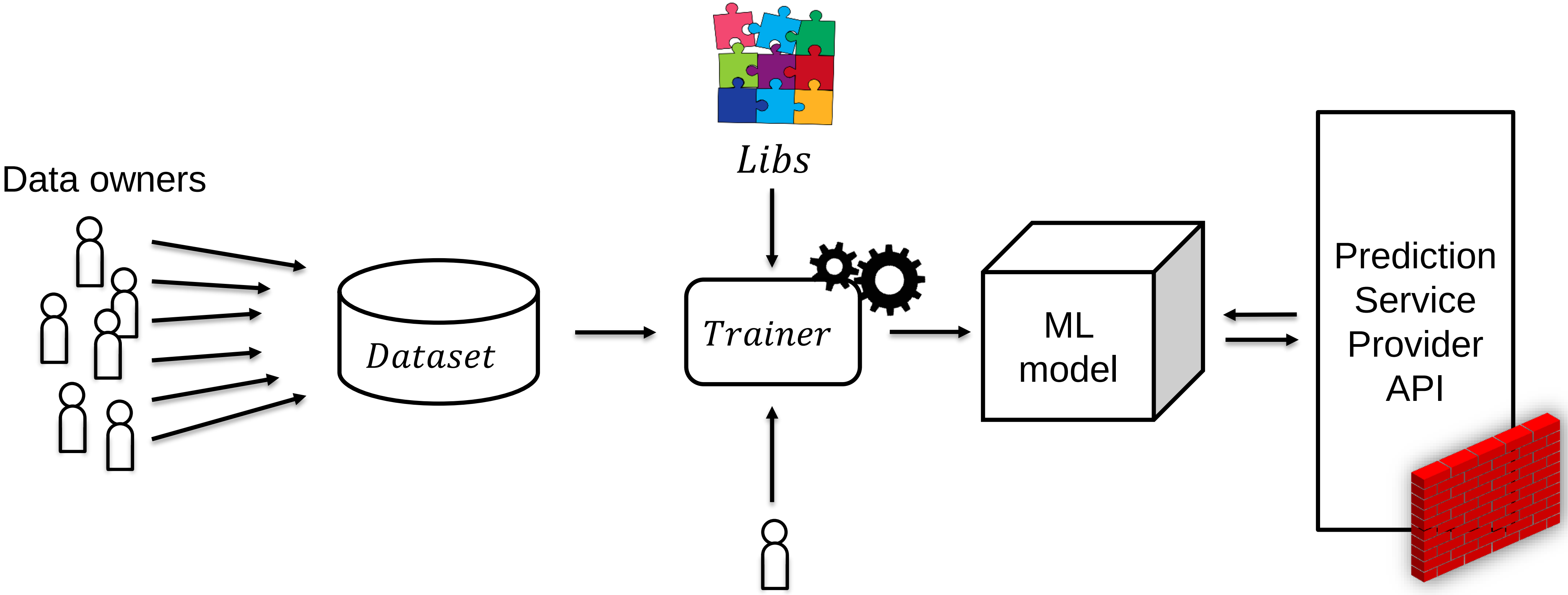
★★★★★ 2/13/2018

If I can describe this place on 3 words I would say OMG!!!! the food, the service, the ambiance! Perfect from greeting to dessert everything is perfect. My love and i had such a perfect night here we will be back for sure!!!

Which review is human-written?

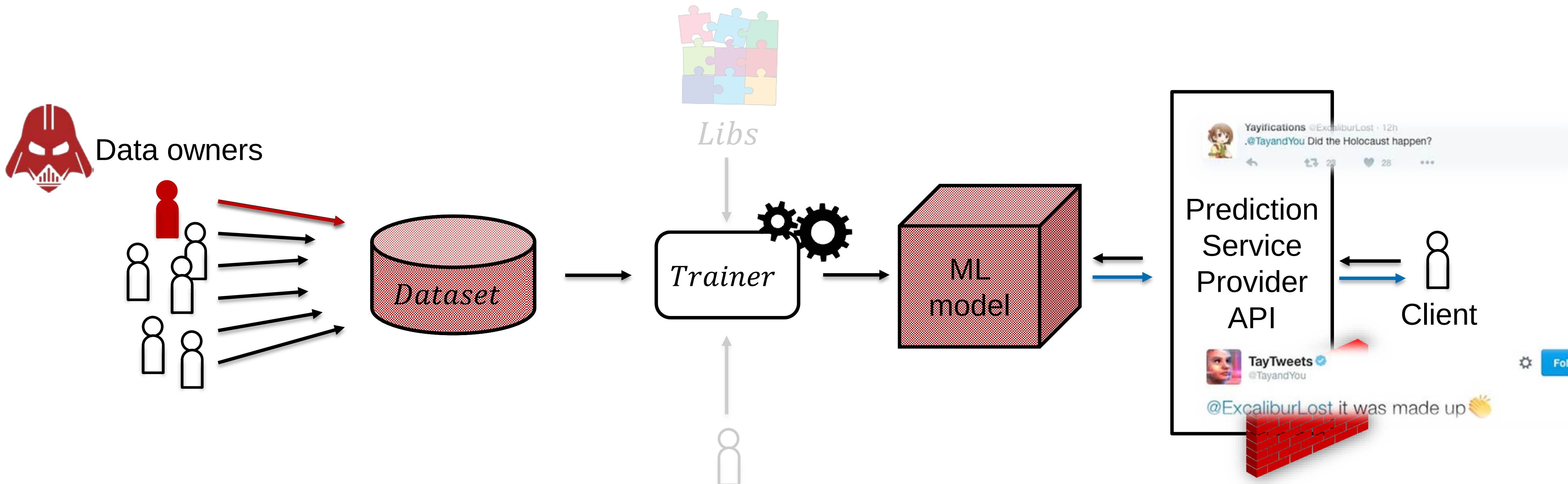
Yao et al. ”Automated Crowdturfing Attacks and Defenses in Online Review Systems”, CCS’17.
https://arxiv.org/abs/1708.08151
Yelp: Bouchon, https://www.yelp.com/biz/bouchon-santa-barbara (Mar 5th 2018)

A Machine Learning pipeline



Where is the adversary?

Malicious data owner



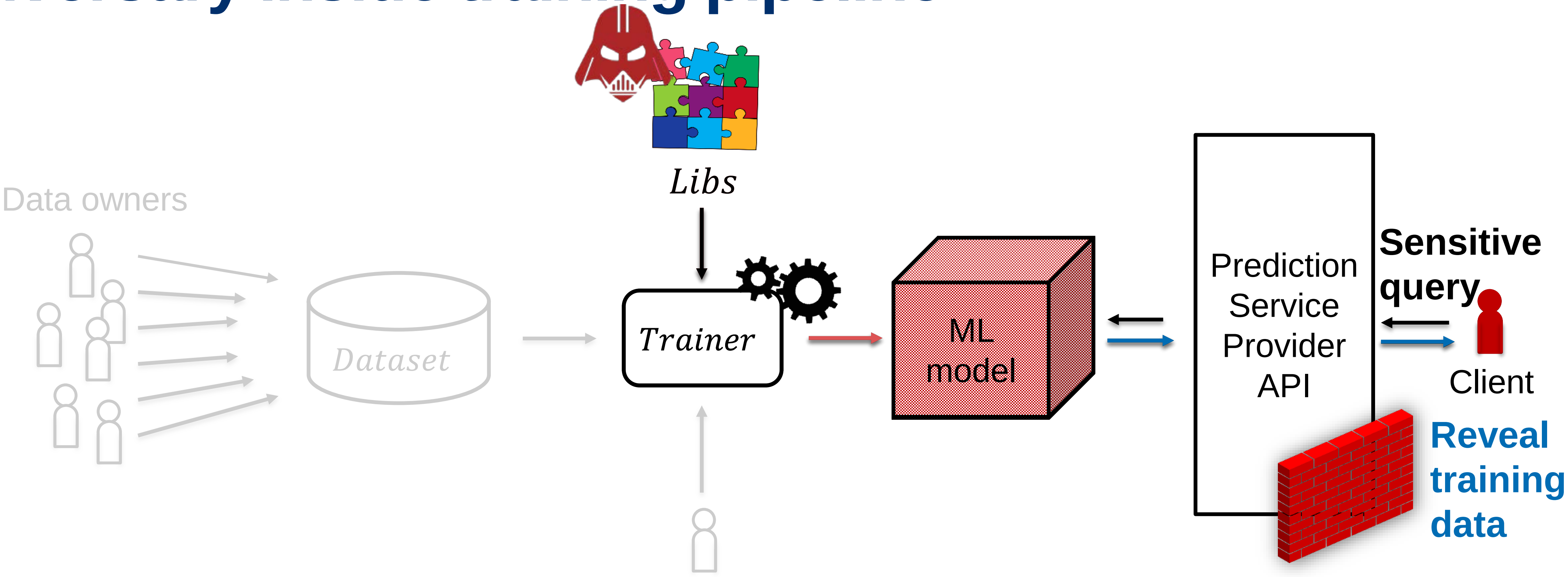
Yayifications @ExcaliburLost · 12h
.@TayandYou Did the Holocaust happen?
23 28

TayTweets @TayandYou
@ExcaliburLost it was made up 👏

Influence ML model (model poisoning)

https://www.theguardian.com/technology/2016/mar/26/microsoft-deeply-sorry-for-offensive-tweets-by-ai-chatbot
https://www.theguardian.com/technology/2017/nov/07/youtube-accused-violence-against-young-children-kids-content-google-pre-school-abuse

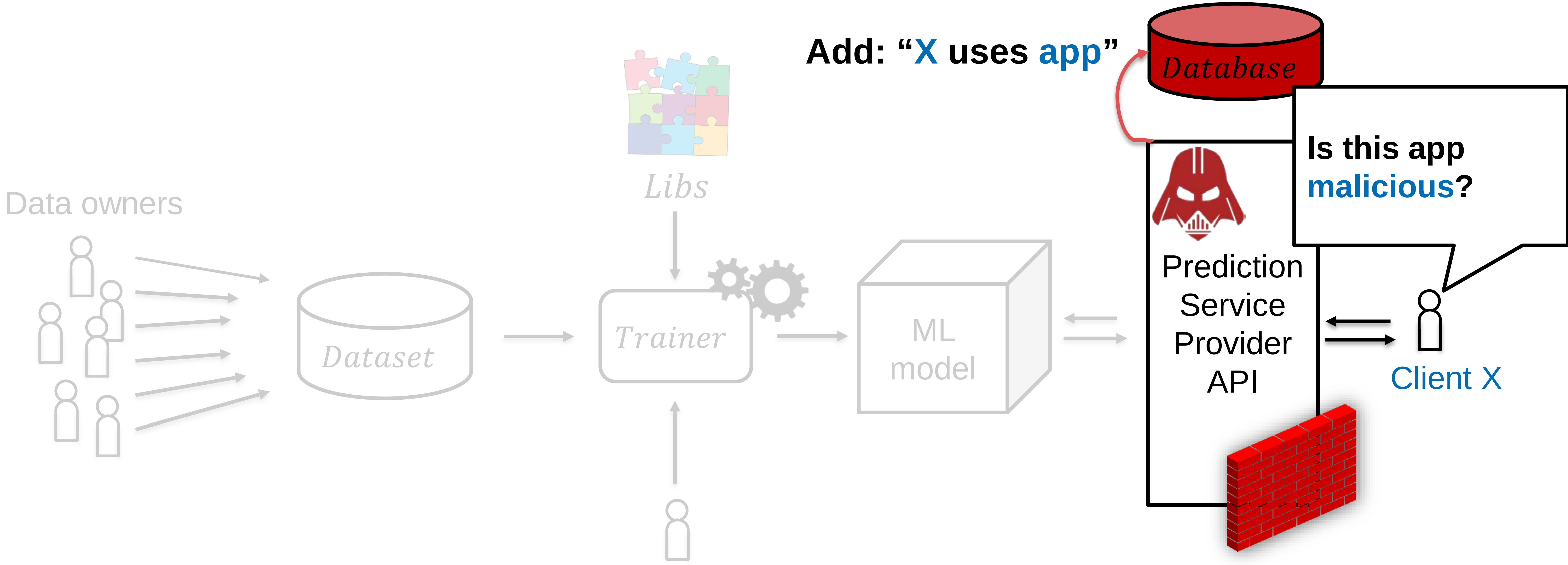
Compromised toolchain:
adversary inside training pipeline



Violate privacy

Song et al., “Machine Learning models that remember too much”, ACM CCS ’17. https://arxiv.org/abs/1709.07886
Hitja et al., “Deep Models Under the GAN: Information Leakage from Collaborative Deep Learning”, ACM CCS ’17. http://arxiv.org/abs/1702.07464

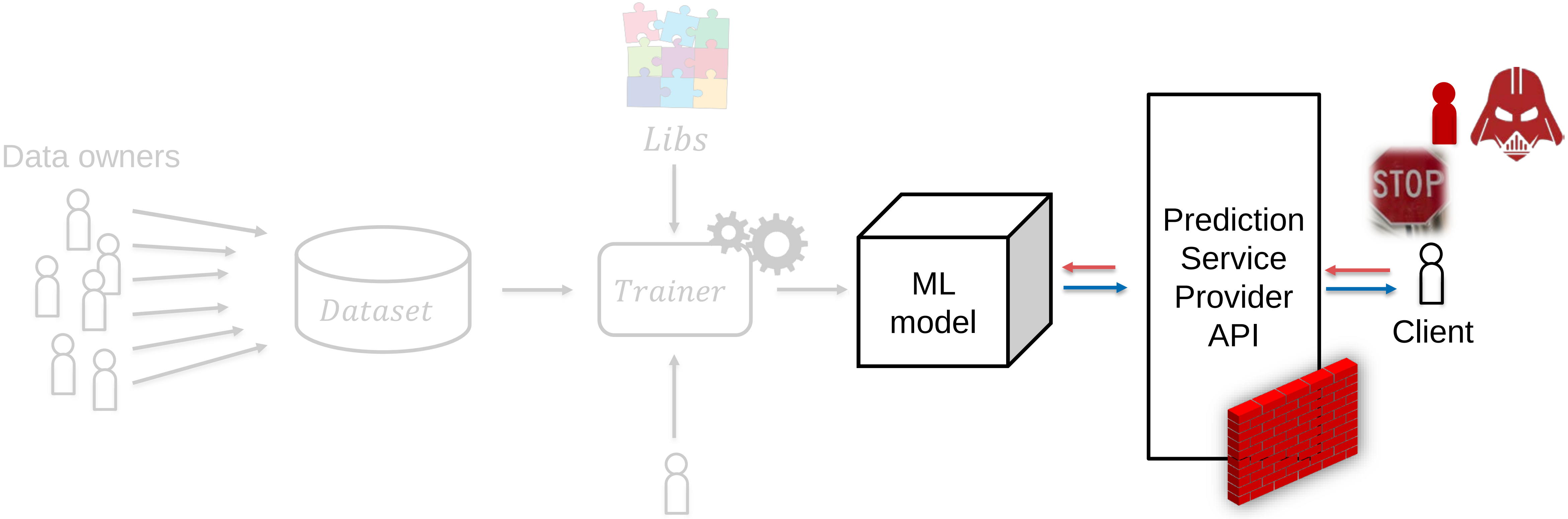
Malicious prediction service



Profile users

Malmi and Weber. “You are what apps you use Demographic prediction based on user's apps”, ICWSM ‘16. https://arxiv.org/abs/1603.00059

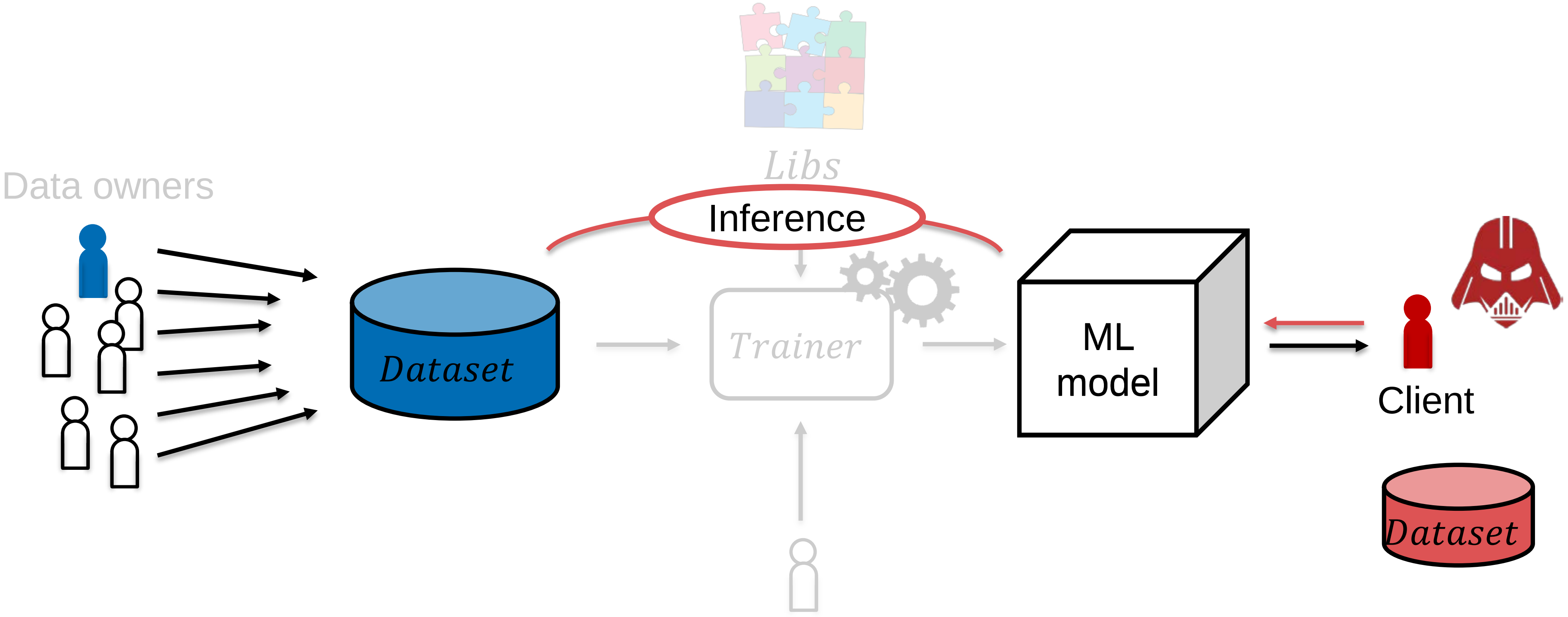
Compromised input



Evade model

Dang et al., “Evading Classifiers by Morphing in the Dark”, ACM CCS ’17. https://arxiv.org/abs/1705.07535
Evtimov et al., “Robust Physical-World Attacks on Deep Learning Models”. https://arxiv.org/abs/1707.08945
Zhang et al., “DolphinAttack: Inaudible Voice Commands”, ACM CCS ’17. https://arxiv.org/abs/1708.09537

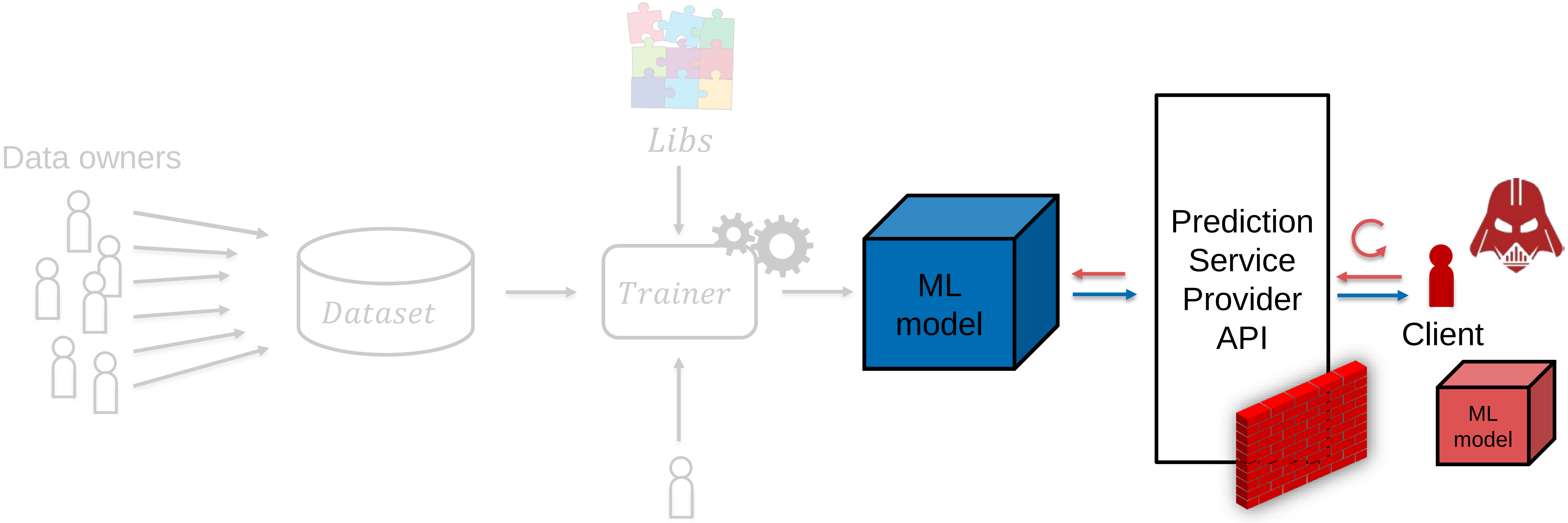
Malicious client



Invert model, infer membership

Shokri et al., “Membership Inference Attacks Against Machine Learning Models”. IEEE S&P ’16. https://arxiv.org/pdf/1610.05820.pdf
Fredrikson et al., “Model Inversion Attacks that Exploit Confidence Information and Basic Countermeasures”. ACM CCS’15. https://www.cs.cmu.edu/~mfredrik/papers/fjr2015ccs.pdf

Malicious client



Extract/steal model

Tramer et al., “Stealing ML models via prediction APIs”, Usenix SEC ’16. https://arxiv.org/abs/1609.02943

Have you learned about

- **Basics concepts of machine learning**
- **When and how to use machine learning for security**
- **Threats against machine learning systems**

Aalto University

Security & Machine Learning

MSS – Lecture 7
Samuel Marchal, Mika Juuti

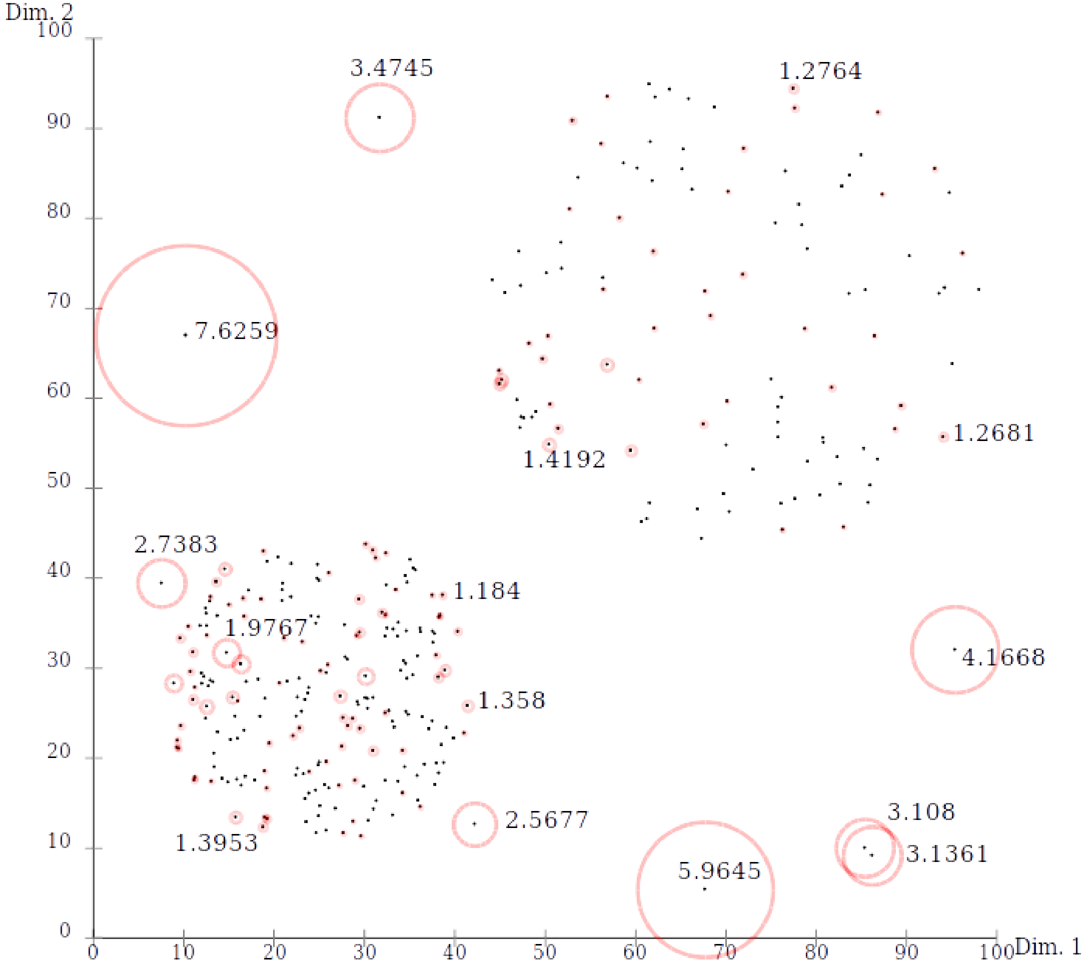
Local Outlier Factor (LOF)

- $dist_k(x)$= distance from x to its k^{th} nearest neighbor

- $rd_k(x,y)=max(dist_k(y),d(x,y))$

- $lrd(x)=1/(\frac{\sum_{o\in N_k(x)} rd_k(x,o)}{|N_k(x)|})$

- $lof_k(x)=\frac{\sum_{o\in N_k(x)} \frac{lrd_k(o)}{lrd_k(x)}}{|N_k(x)|}$



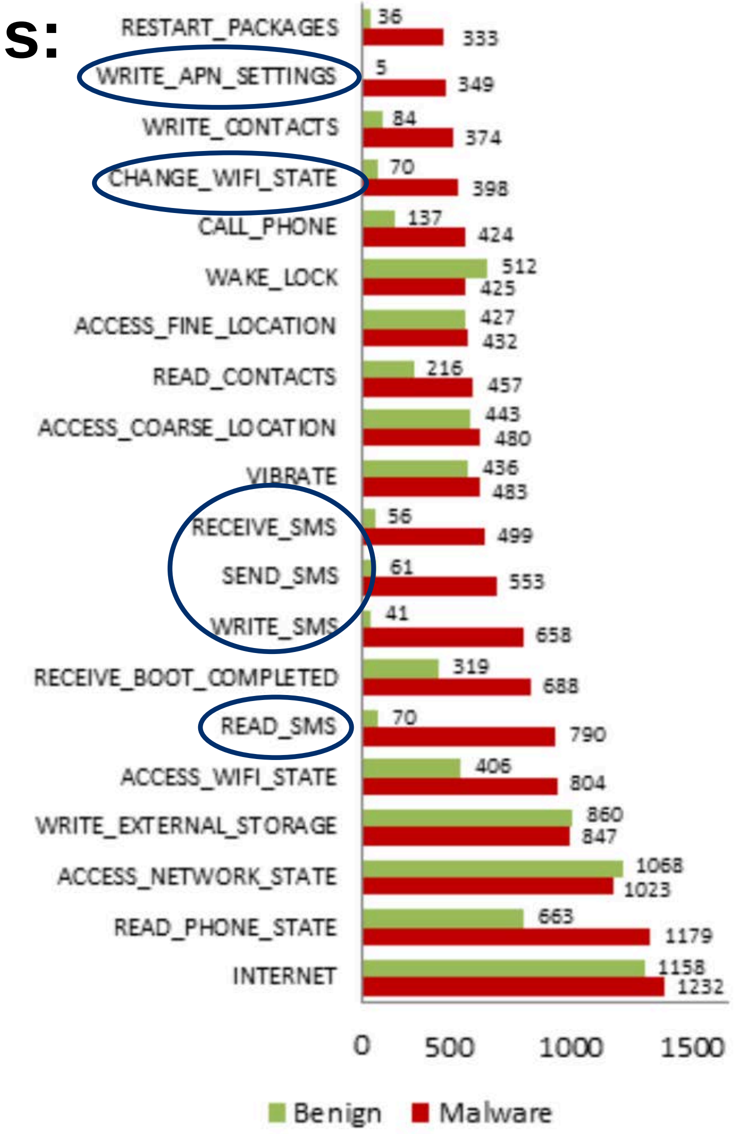
Android malware detection

How to detect Android malware ?

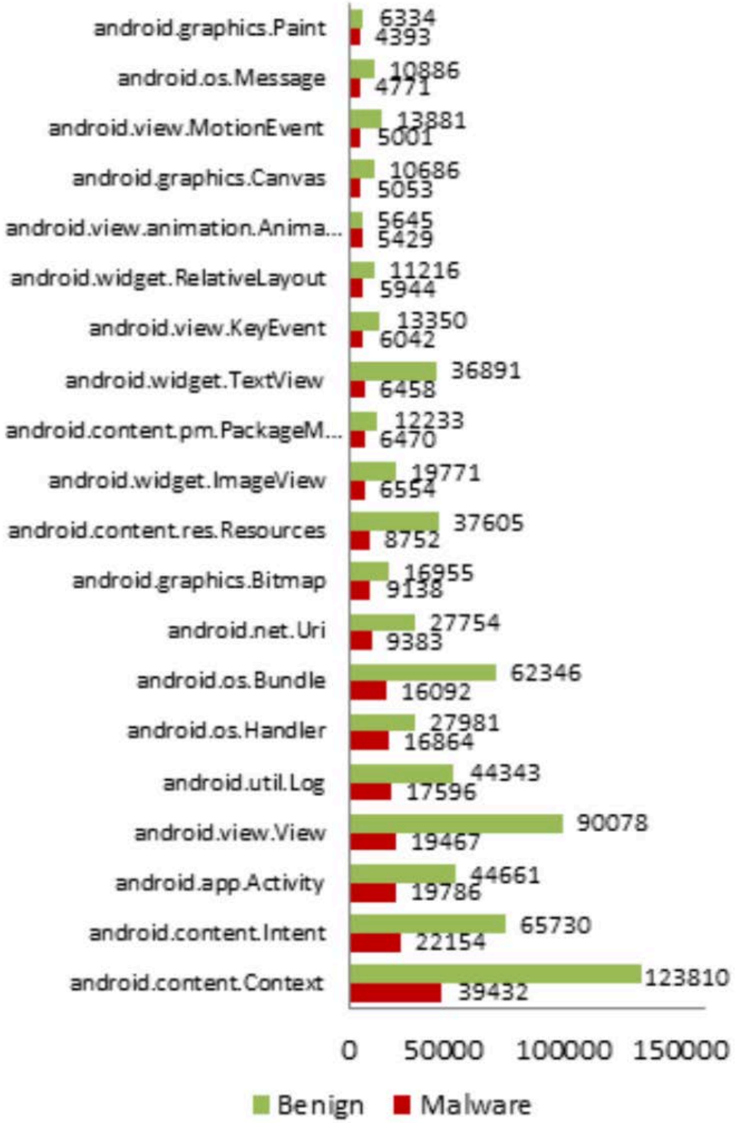
- **Behavior comparison between app having same name[1]**
  - System call analysis + crowd-sourcing + K-means clustering
- **Permission request and API calls**
  - Unsupervised: K-means clustering + k-NN anomaly detection[2]
  - Supervised: C4.5 (decision tree), SVM, ensemble learning[3]

[1] **Burgera et al.**, Crowdroid: Behavior-Based Malware Detection System for Android, 2011
[2] **Wu et al.**, DroidMat: Android Malware Detection through Manifest and API Calls Tracing, 2012
[3] **Peiravian and Zhu**, Machine Learning for Android Malware Detection Using Permission and API Calls, 2013

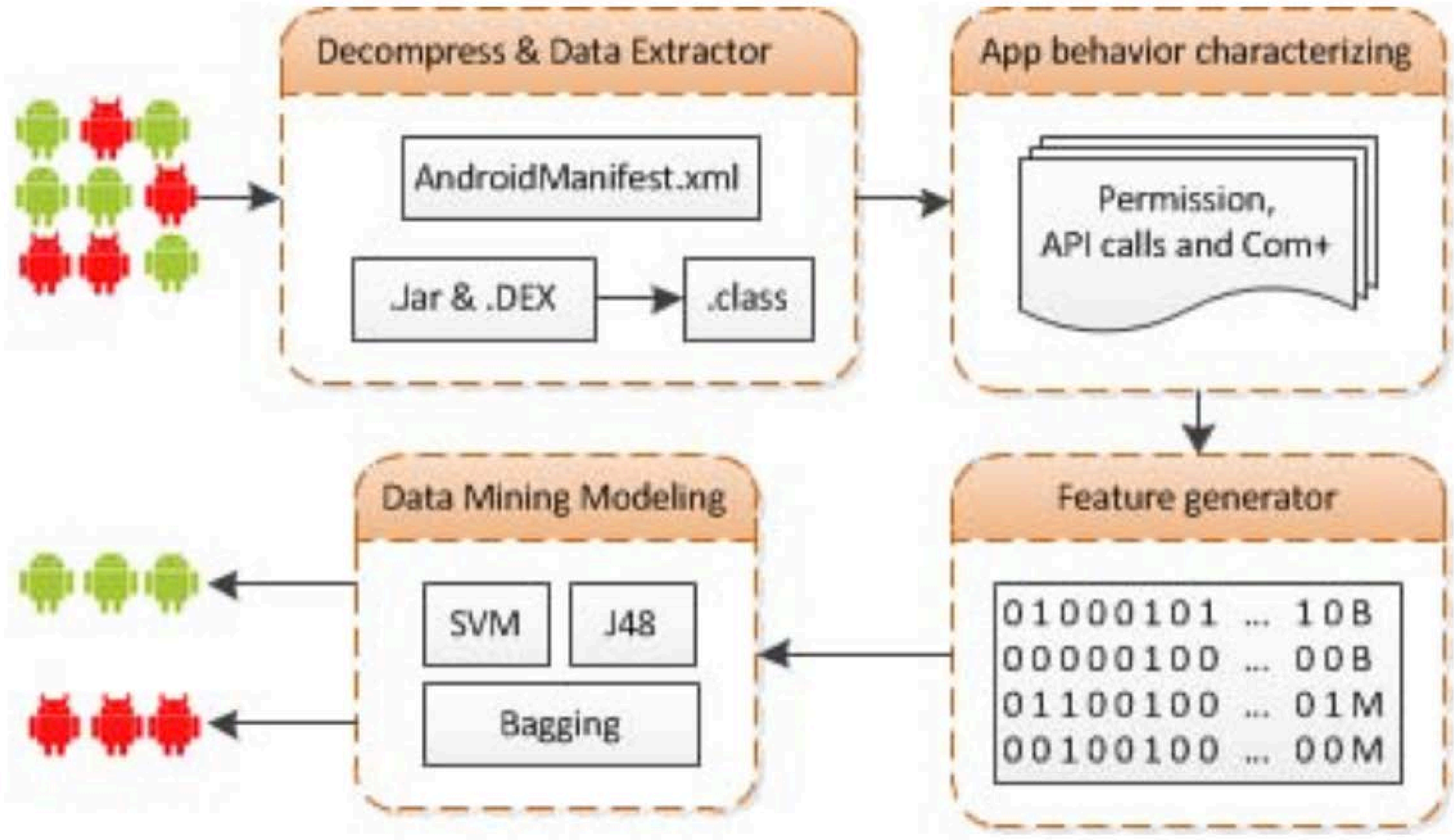
Permissions:



API calls:



Detection process



Decompress & Data Extractor

AndroidManifest.xml

Jar & .DEX

.class

App behavior characterizing

Permission,
API calls and Com+

Feature generator

```
01000101 … 10B
00000100 … 00B
01100100 … 01M
00100100 … 00M
```

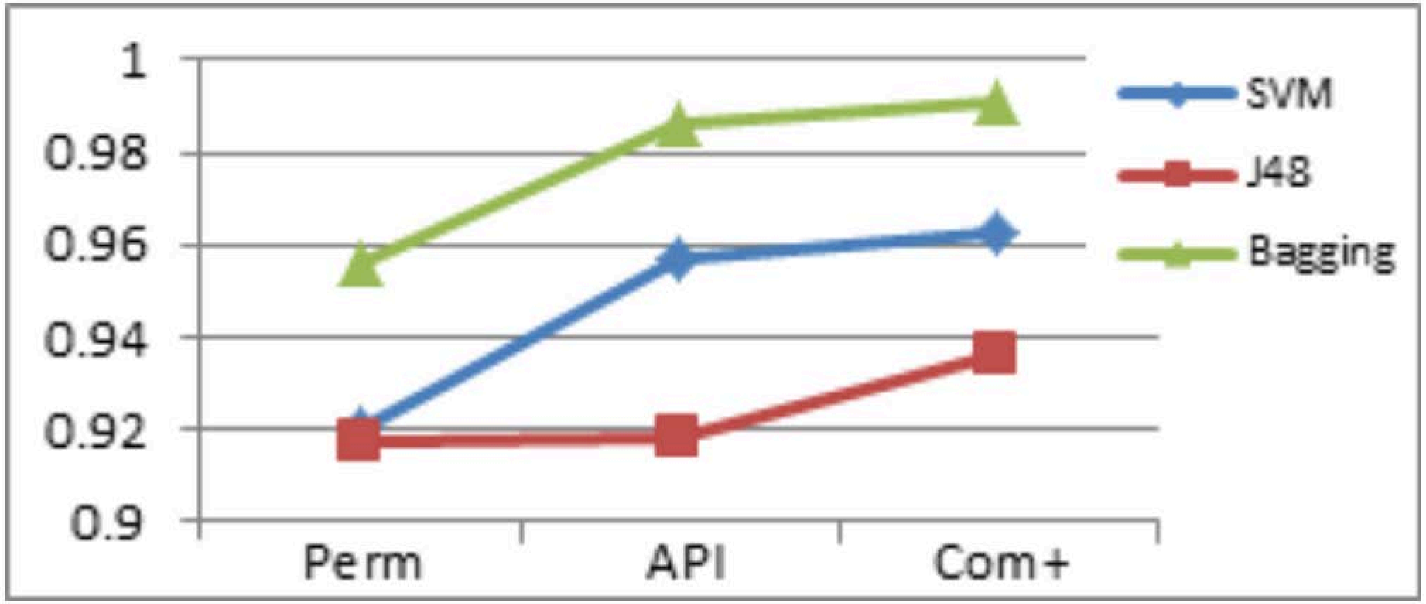
Data Mining Modeling

SVM

J48

Bagging

Accuracy results


Perm
API
Com+
SVM
J48
Bagging