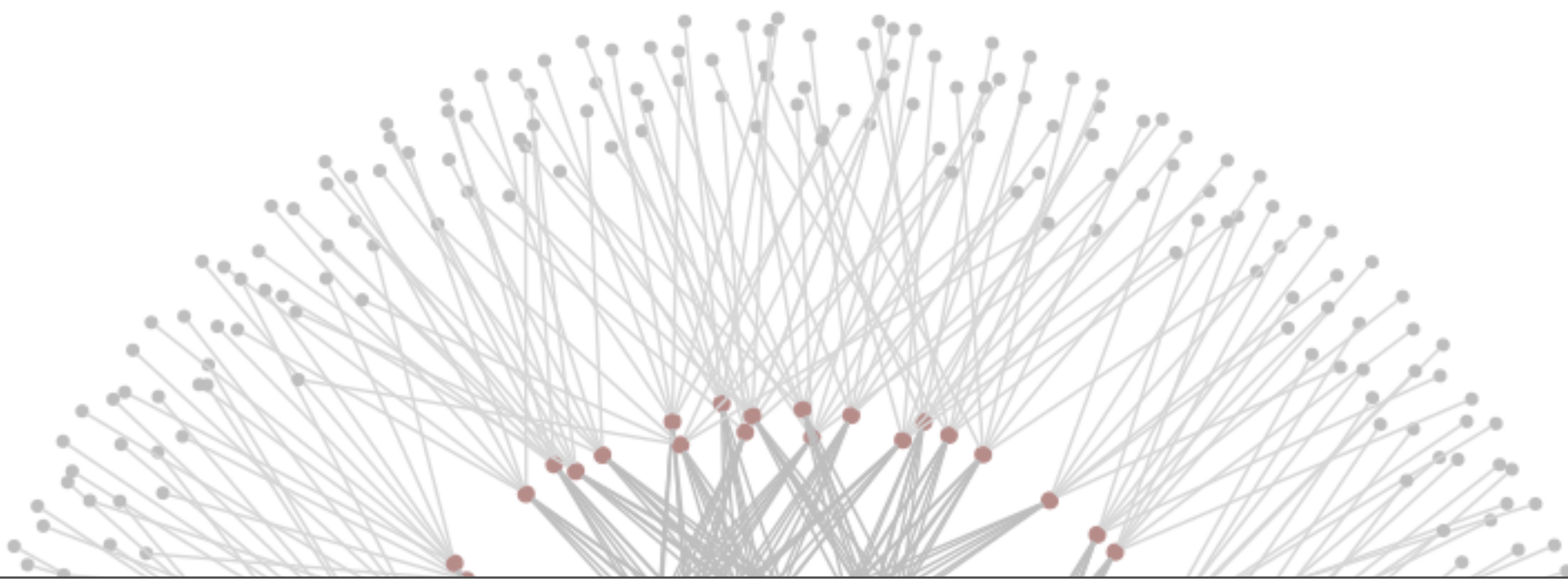
# Computer Security: A Machine Learning Perspective


Phuong Cao

University of Illinois at Urbana Champaign

# Agenda

**Overview of Machine Learning**

**Supervised Learning Framework**

Example: Malicious websites detection

Attack againts supervised learning

**Unsupervised Learning Framework**

Univariate Event Detection

**Future Work**

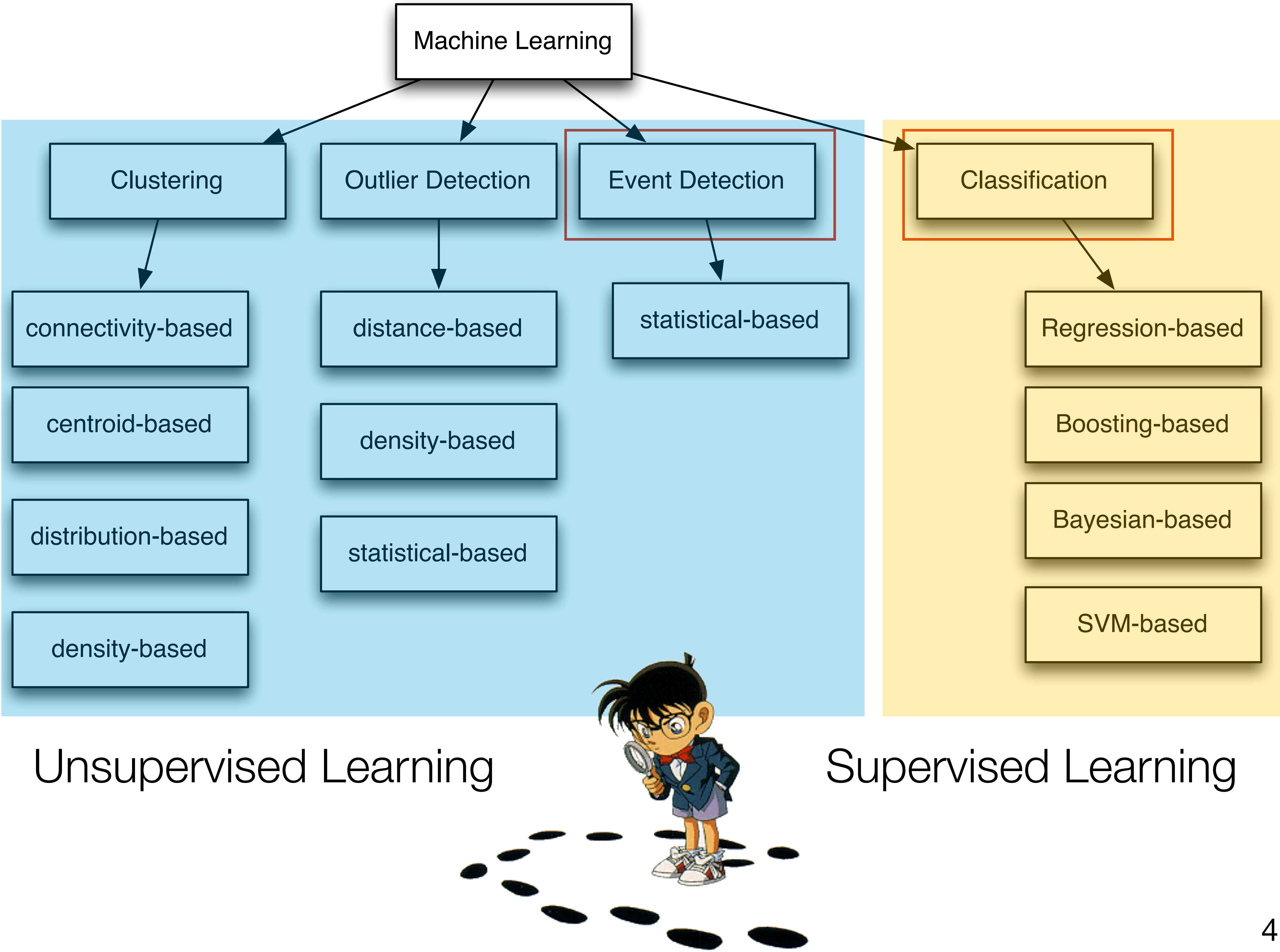
# What is Machine Learning?

"The science of getting computers to act without being explicitly programmed"

Andrew Ng, Associate Professor at Stanford.

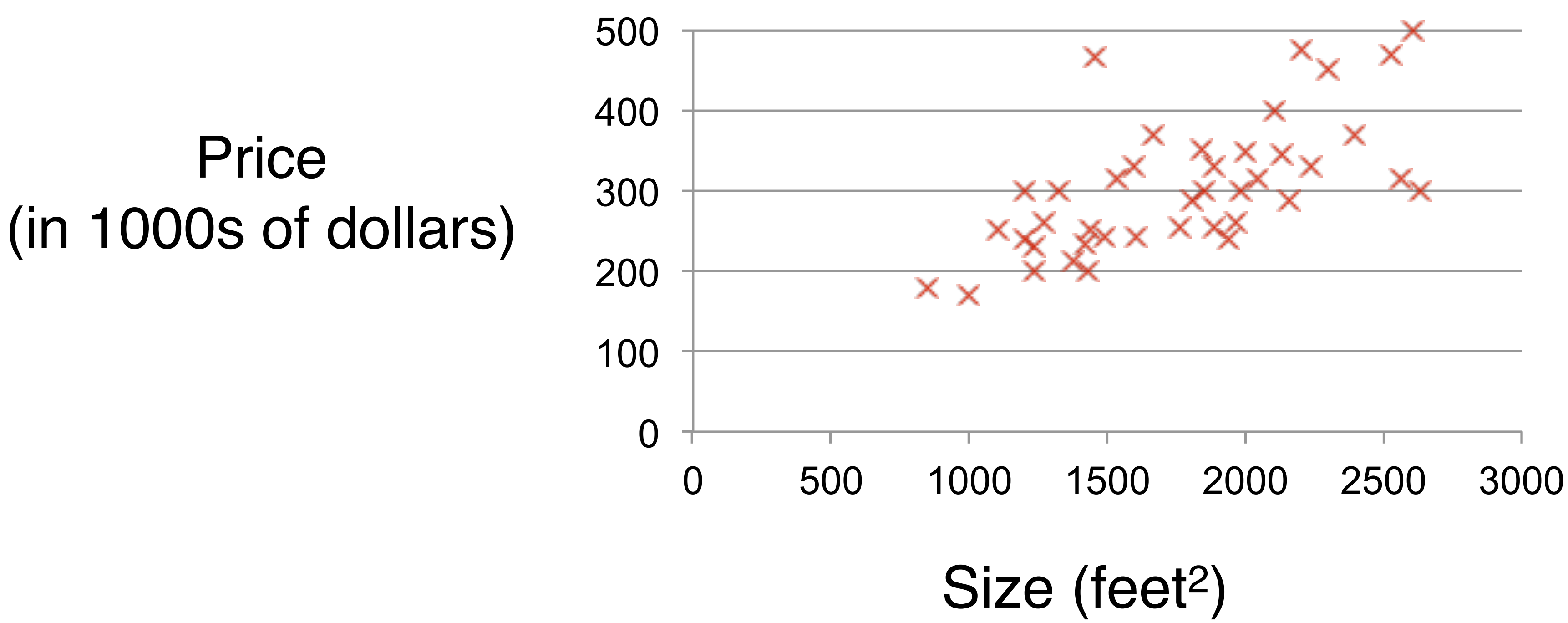
**Examples**

Malicious URLs detection, malware classification, fraud detection, terrorist identification

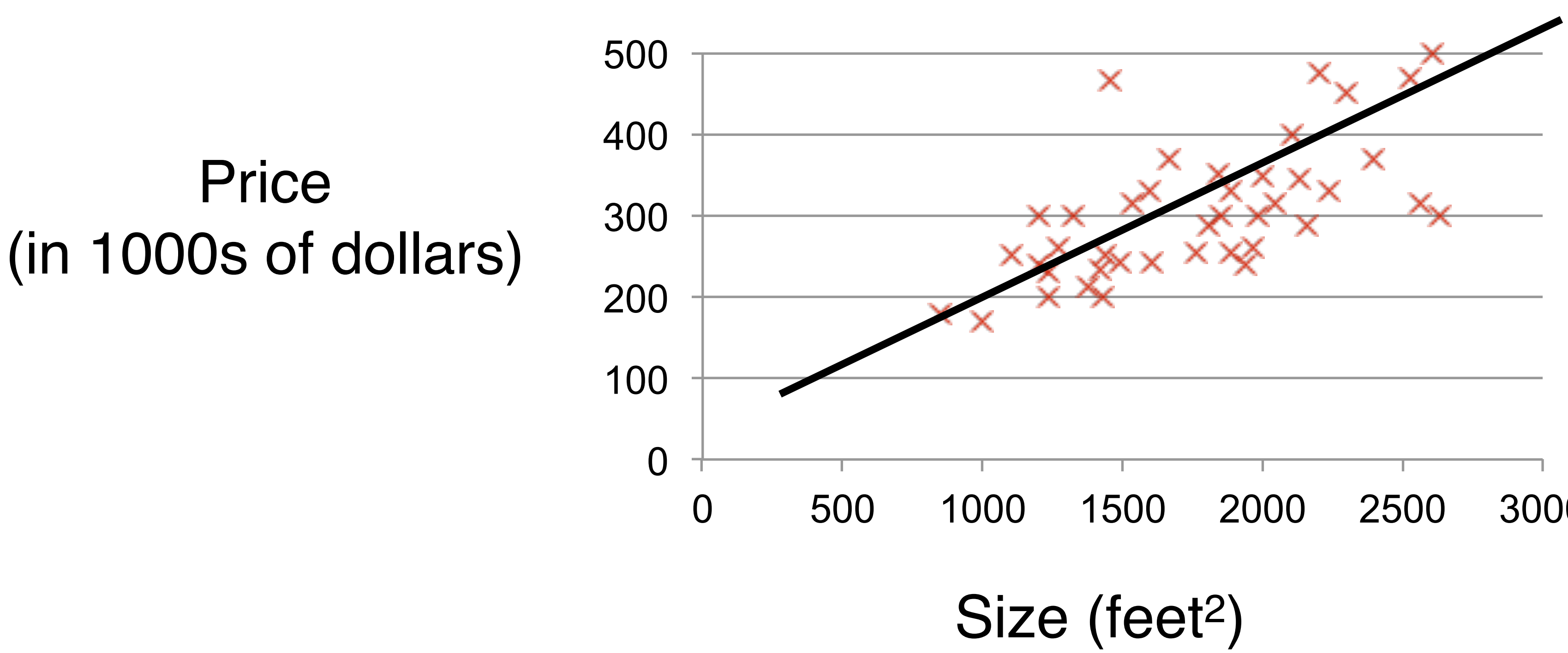

Machine Learning
Clustering
Outlier Detection
Event Detection
Classification
connectivity-based
centroid-based
distribution-based
density-based
distance-based
density-based
statistical-based
statistical-based
Regression-based
Boosting-based
Bayesian-based
SVM-based
Unsupervised Learning
Supervised Learning

# Supervised Learning Overview

# Supervised Learning Overview



# Supervised Learning Overview

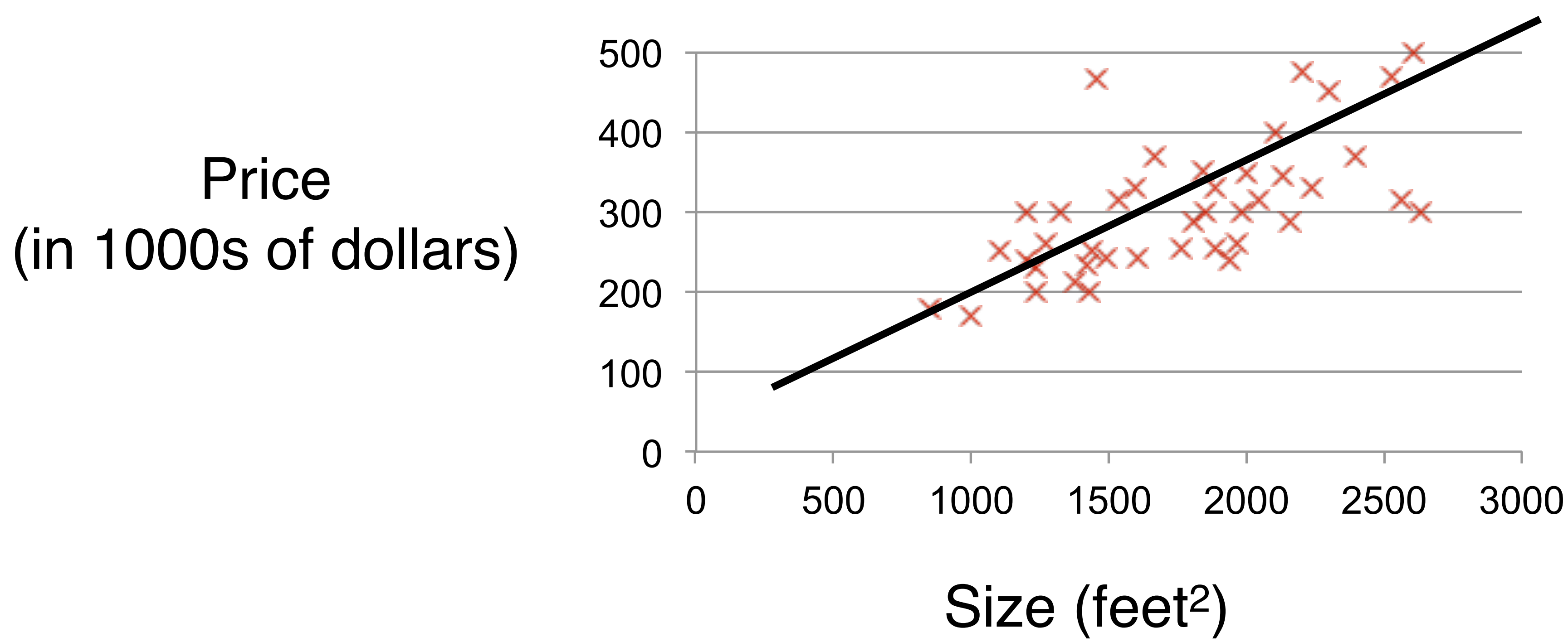


# Supervised Learning Overview

Infers a model from supervised (labeled) training data [Ng11]

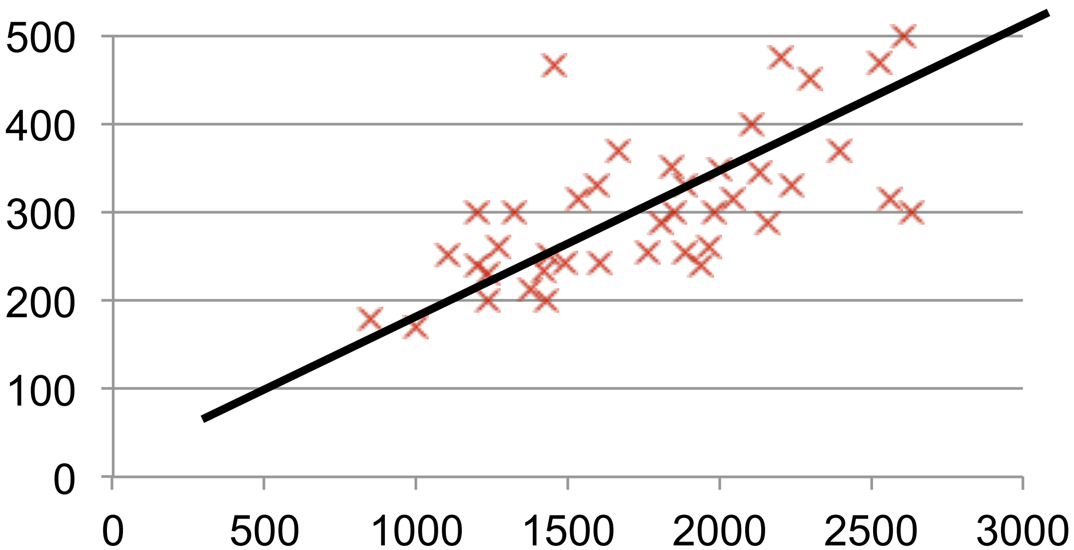
**Input**: labeled data

**Output**: function (linear/non-linear or probabilistic based)



$\theta_0,\theta_1$

# Supervised Learning Model [Ng11]




Hypothesis $h_\theta(x) = \theta_0 + \theta_1 x$

Parameters $\theta_0, \theta_1$

Cost $J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left( h_\theta(x^{(i)}) - y^{(i)} \right)^2$

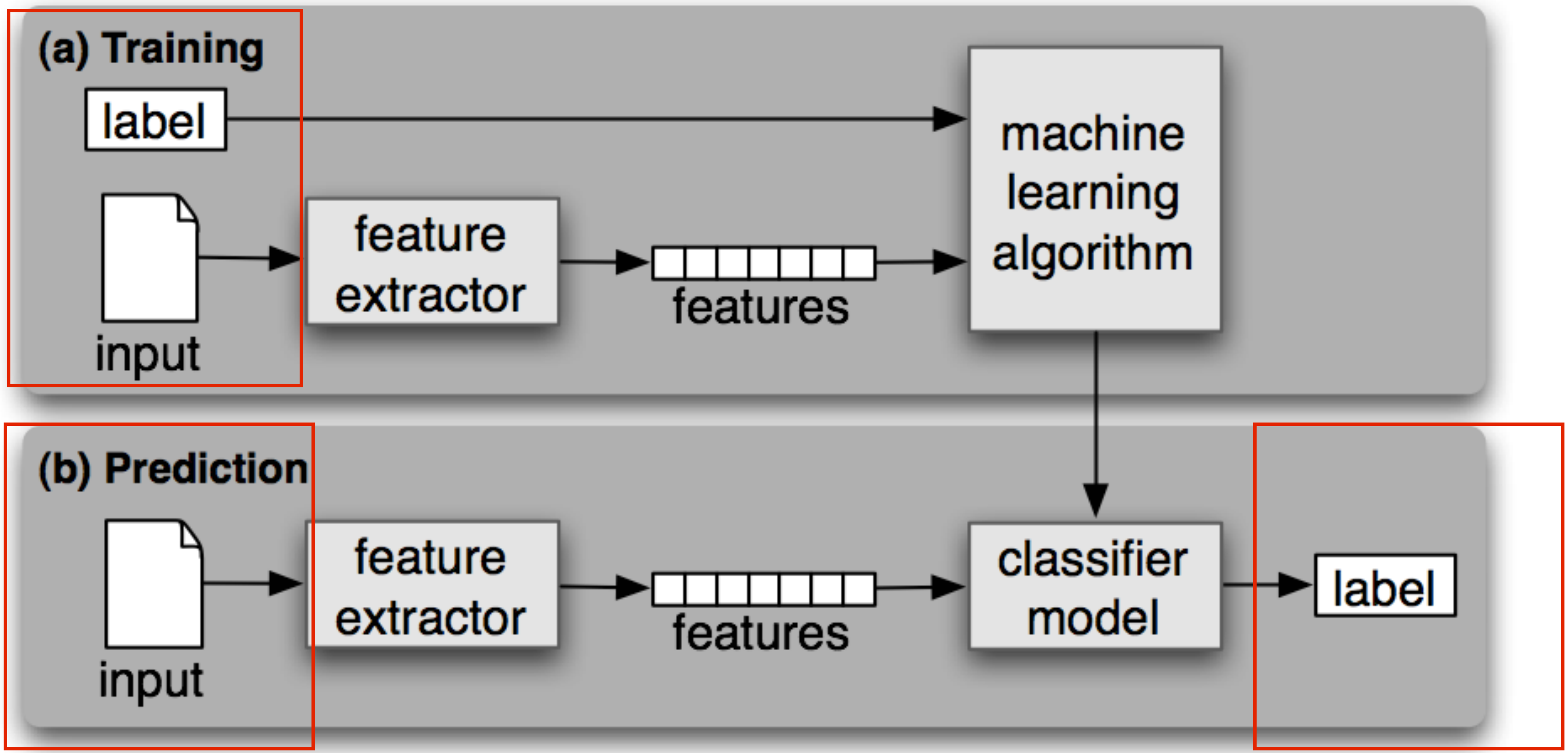
Goal $\underset{\theta_0,\theta_1}{\text{minimize}}\ J(\theta_0, \theta_1)$

1. Propose a hypothesis

2. Define cost function

3. Minimize cost function

# Supervised Learning Framework [Bird09]




http://nltk.googlecode.com/svn/trunk/doc/book/ch06.html

# Example: Malicious URLs [Ma08]

**Problem definition**

Given a website w and a set of labeled malicious/benign websites, identify whether w is malicious or not?

**Training data**

URLs and label of malicious/benign websites

- White and grey list: Alexa top 1M sites
- Blacklist: malicious domains used by botnets, phishing emails, etc.

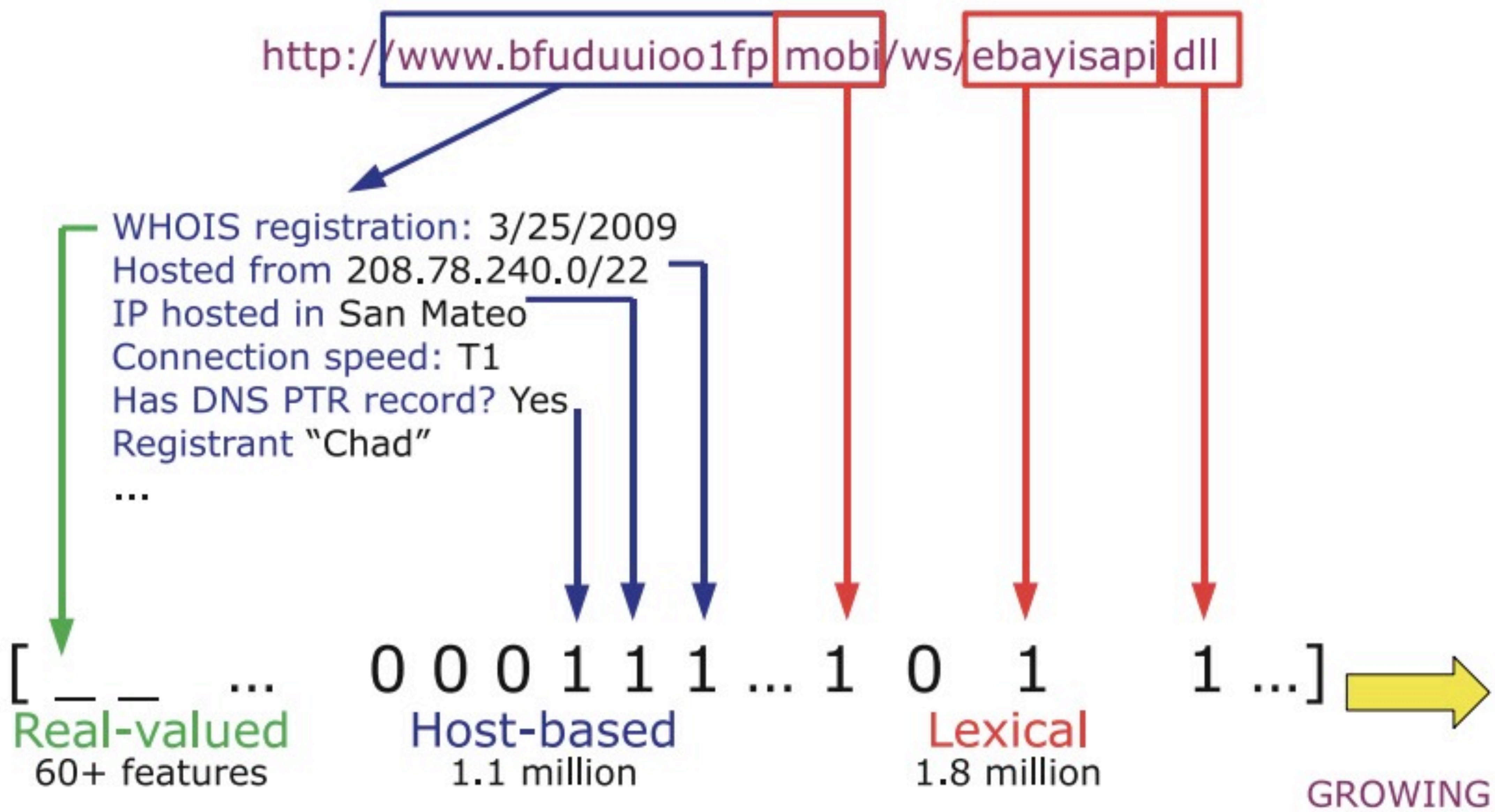
# Feature Extraction from URLs

# Feature Extraction from URLs

http://www.bfuduuioo1fp.mobi/ws/ebayisapi.dll

# Feature Extraction from URLs

http://www.bfuduuioo1fp.mobi/ws/ebayisapi.dll



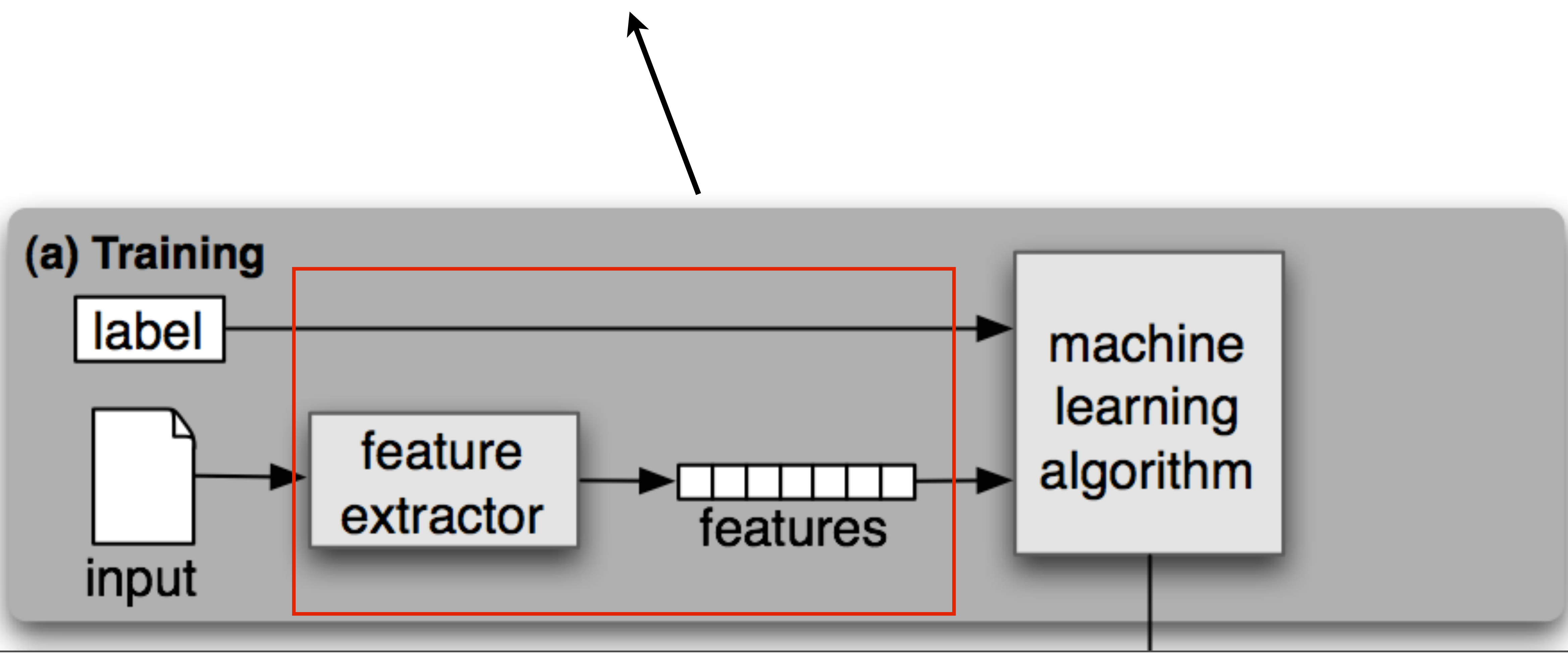
# Feature Extraction from webpage

Website structures (DOM Tree)

Types of advertisings

Types of in/out links

Scale-invariant feature transform (SIFT) of images



# Classification Methods

**Standard Classifiers:** SVM, Naive Bayes, Logistic Regression [Romano07, Hosmer04, McCallum98]

**Results may vary!**

**SVM, Logistic Regression**: Over-fitting of training data.

**Naive Bayes:** Depends on independence assumption

**Boosting:** AdaBoost, Gradient Boosted Decision Tree [Collins02]

**Key idea:** combine multiple weak classifier for a strong classifier

**Advantages:**

High classification accuracy

Noise-tolerant

# Classification Methods in Practices

**Example**: Google still uses rule-based approach for search ranking.

Quora Search

**Edmond Lau, Ex-Google Search Quality Engineer**

**331** votes by Anne K. Halsall, Adam D'Angelo, Mark Cao, (more)

From what I gathered while I was there, Amit Singhal, who heads Google's core ranking team, has a philosophical bias against using machine learning in search ranking. My understanding for the two main reasons behind this philosophy is:

1. In a machine learning system, it's hard to explain and ascertain why a particular search result ranks more highly than another result for a given query. The explainability of a certain decision can be fairly elusive; most machine learning algorithms tend to be black boxes that at best expose weights and models that can only paint a coarse picture of why a certain decision was made.
2. Even in situations where someone succeeds in identifying the signals that factored into why one result was ranked more highly than other, it's difficult to directly tweak a machine learning-based system to boost the importance of certain signals over others in isolated contexts. The signals and features that feed into a machine learning system tend to only indirectly affect the output through layers of weights, and this lack of direct control means that even if a human can explain why one web page is better than another for a given query, it can be difficult to embed that human intuition into a system based on machine learning.

Rule-based scoring metrics, while still complex, provide a greater opportunity for engineers to directly tweak weights in specific situations. From Google's

# Classification Methods in Practices

**In practice:**

Prefers simple and interpretable models, e.g., decision trees, Bayes network

A ML application must scale

Data is the first class citizen

# Supervised Learning Attacks

# Supervised Learning Attacks

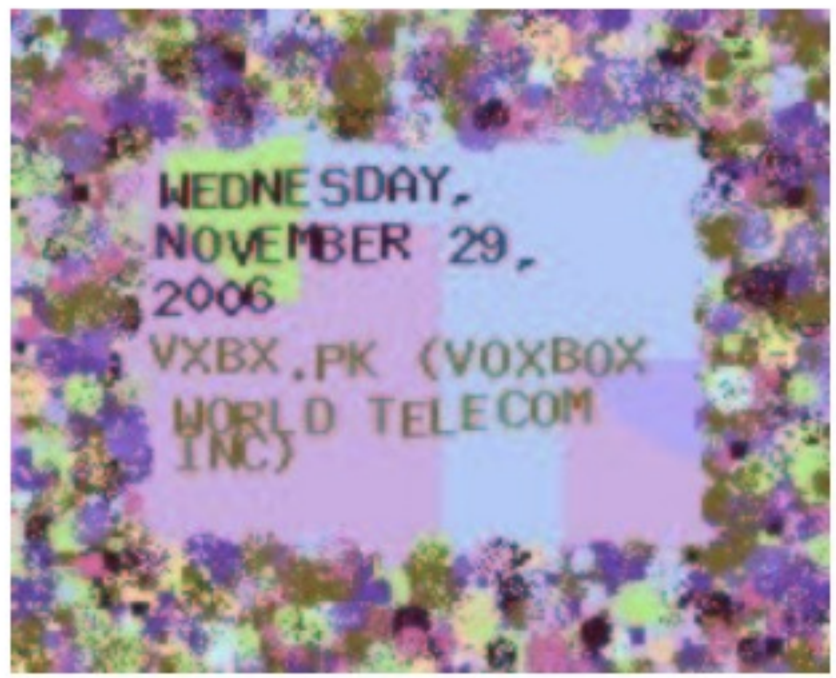
How would you attack a spam filter?

# Supervised Learning Attacks

How would you attack a spam filter?

# Supervised Learning Attacks

How would you attack a spam filter?



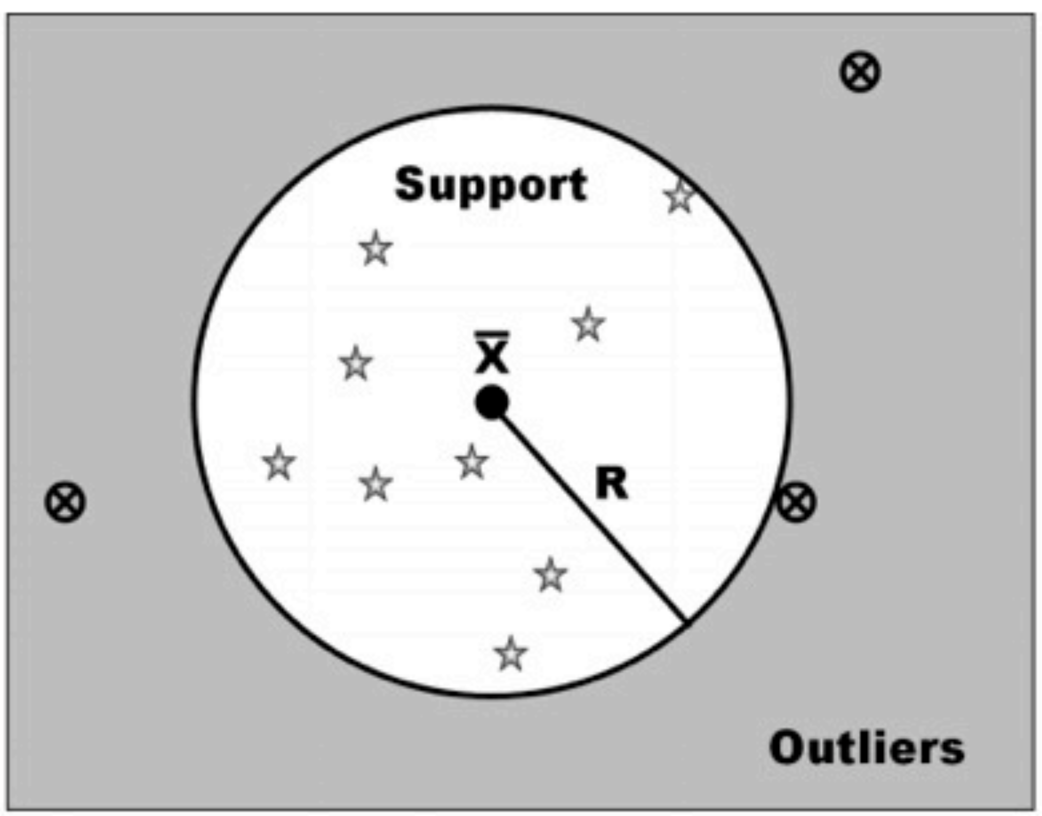

Adversarial spam image designed to defeat OCR text extraction [Chan06]

# Supervised Learning Attacks

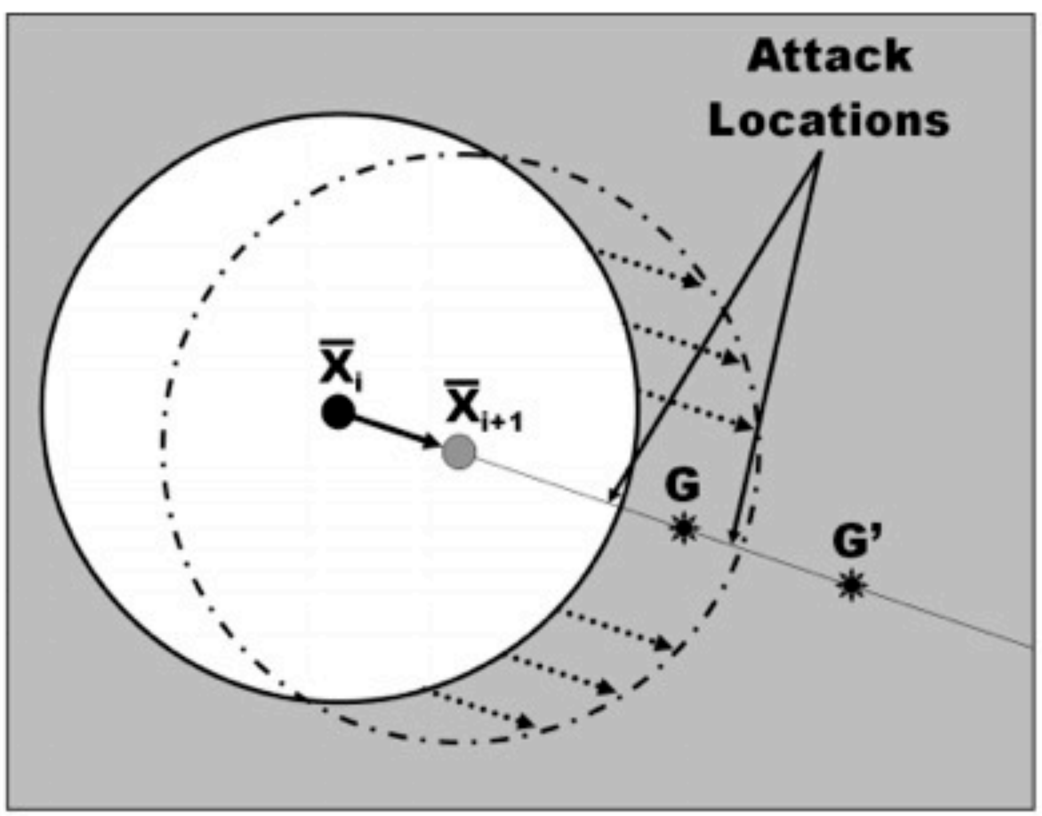
Security Violations [Barreno06]

**Integrity:** Intrusion points classified as normal (false negatives)

**Availability:** Enough classification errors that learner becomes unusable




(a) Hypersphere Outlier Detection




(b) Attack on a Hypersphere Outlier Detector

# Supervised Learning Summary

**Process**

Collect data labels, Extract features, Evaluate classifiers

**Problems**

Over-fitting, sensitive to noise, susceptible to security violation attacks.

# Agenda

**Overview of Machine Learning**

**Supervised Learning Framework**

- Example: Malicious websites detection
- Attack againts supervised learning

**Unsupervised Learning Framework**

- Event Detection in Spatial stream

**Future Work**

# Event Detection

Identify events from spatio-temporal data stream

**Input**: spatio-temporal data stream

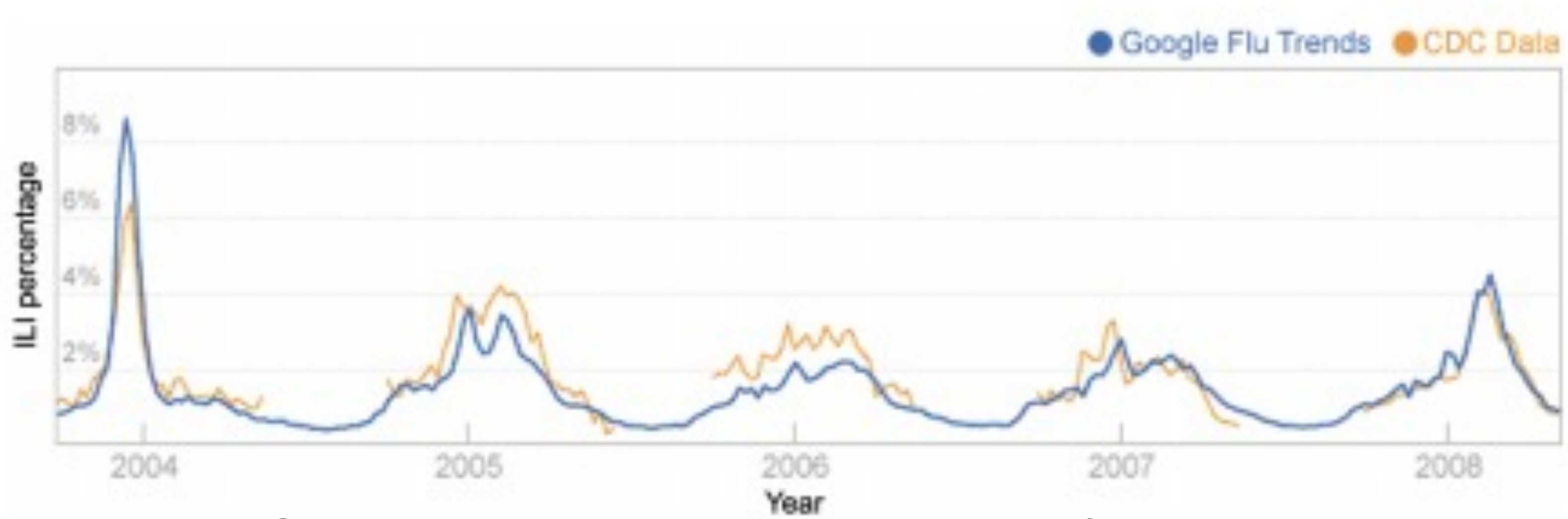
**Output**: spatio-temporal location of events

# Event Detection

Identify events from spatio-temporal data stream

**Input**: spatio-temporal data stream

**Output**: spatio-temporal location of events



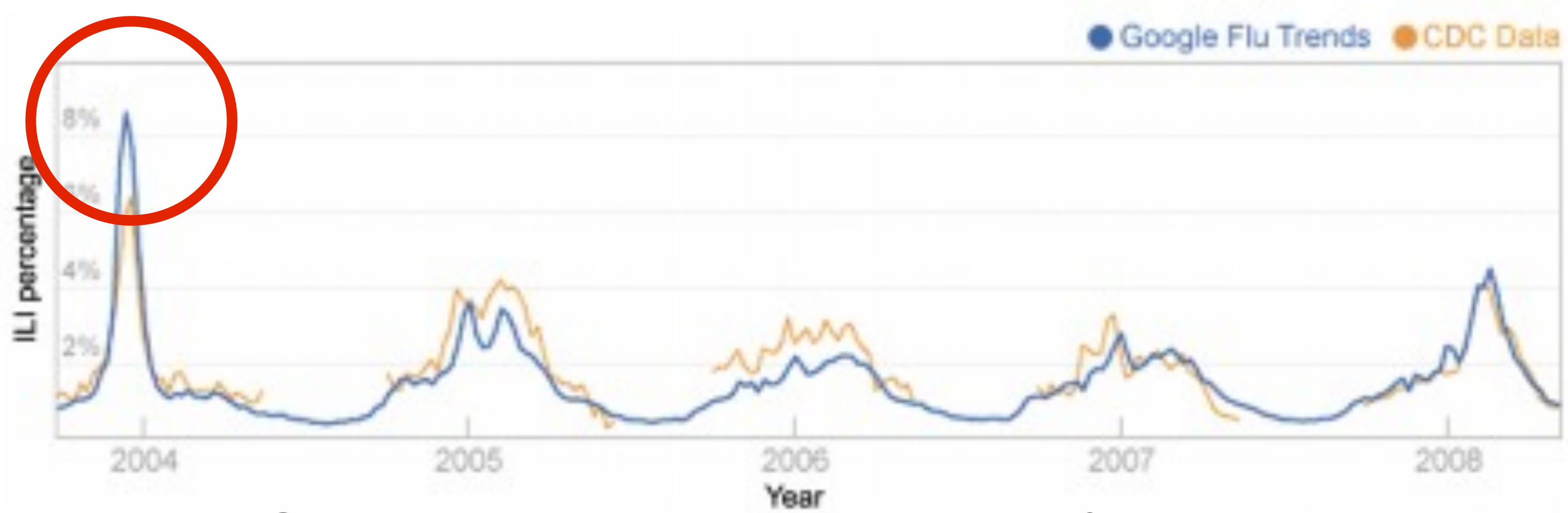

Correlation between reporting flu sickness
Google Flu Trends and Center for Disease Control (CDC), 2004-2008

# Event Detection

Identify events from spatio-temporal data stream

**Input**: spatio-temporal data stream

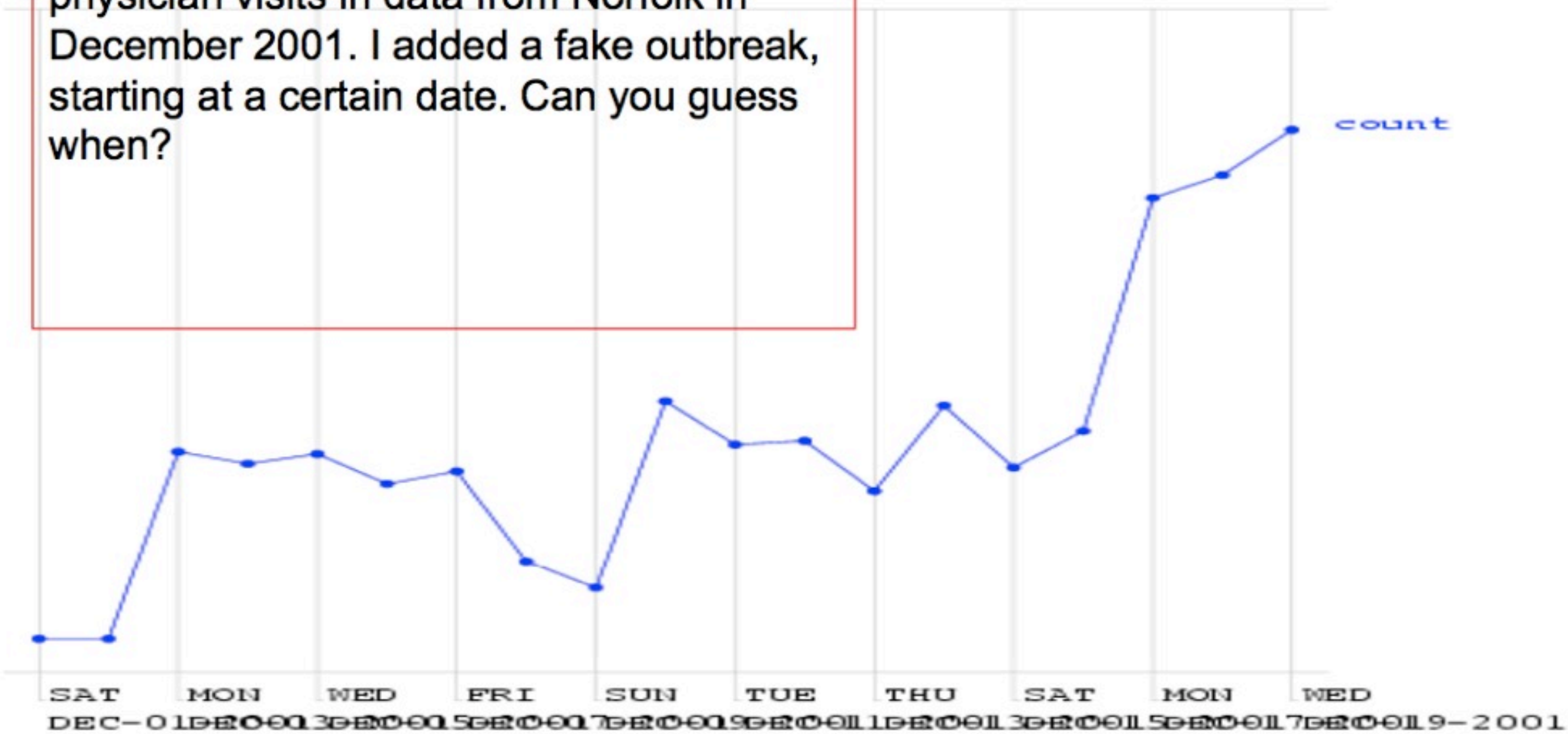
**Output**: spatio-temporal location of events




Correlation between reporting flu sickness
Google Flu Trends and Center for Disease Control (CDC), 2004-2008

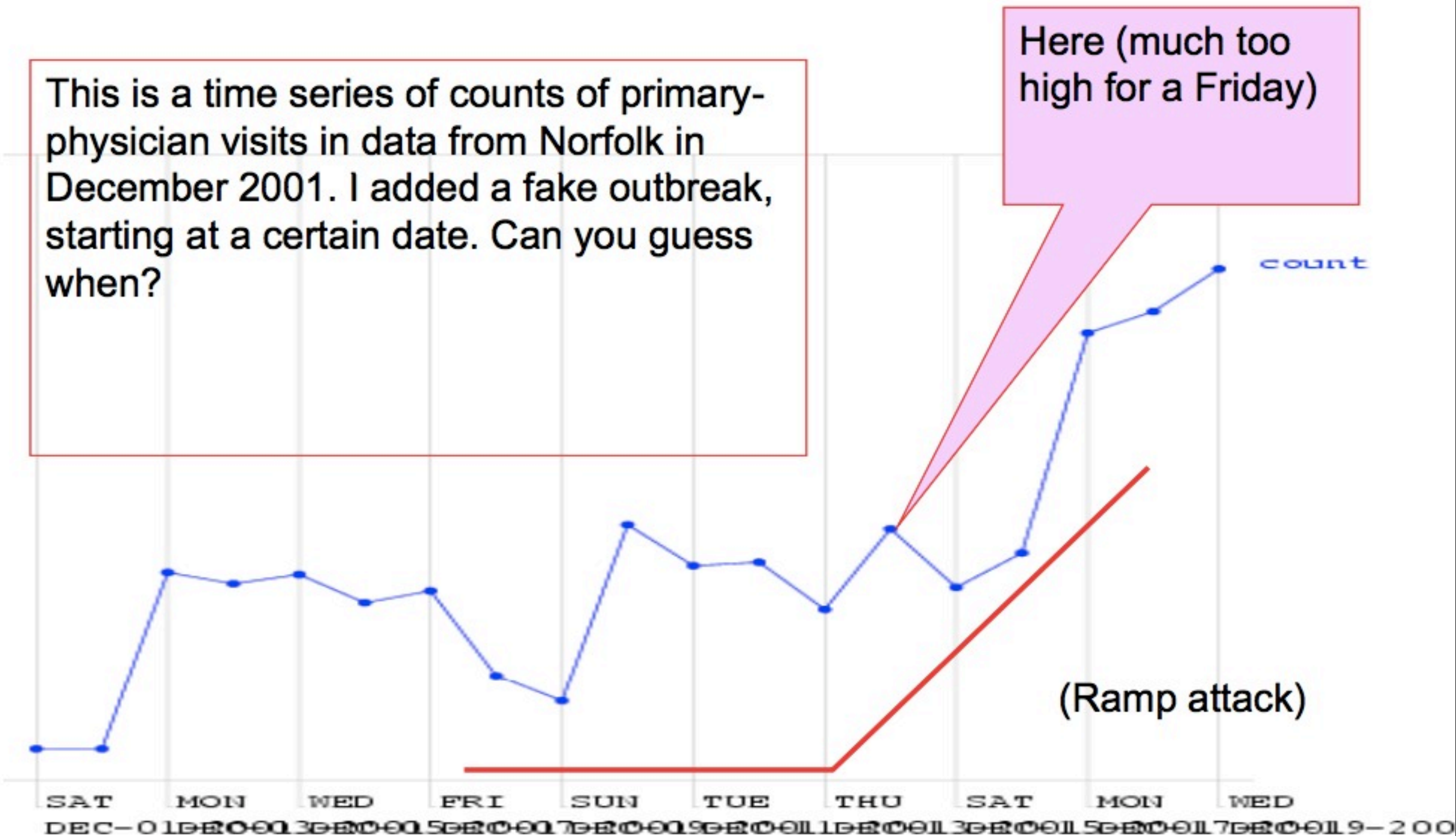
# Univariate Event Detection [Neill09]

This is a time series of counts of primary-physician visits in data from Norfolk in December 2001. I added a fake outbreak, starting at a certain date. Can you guess when?

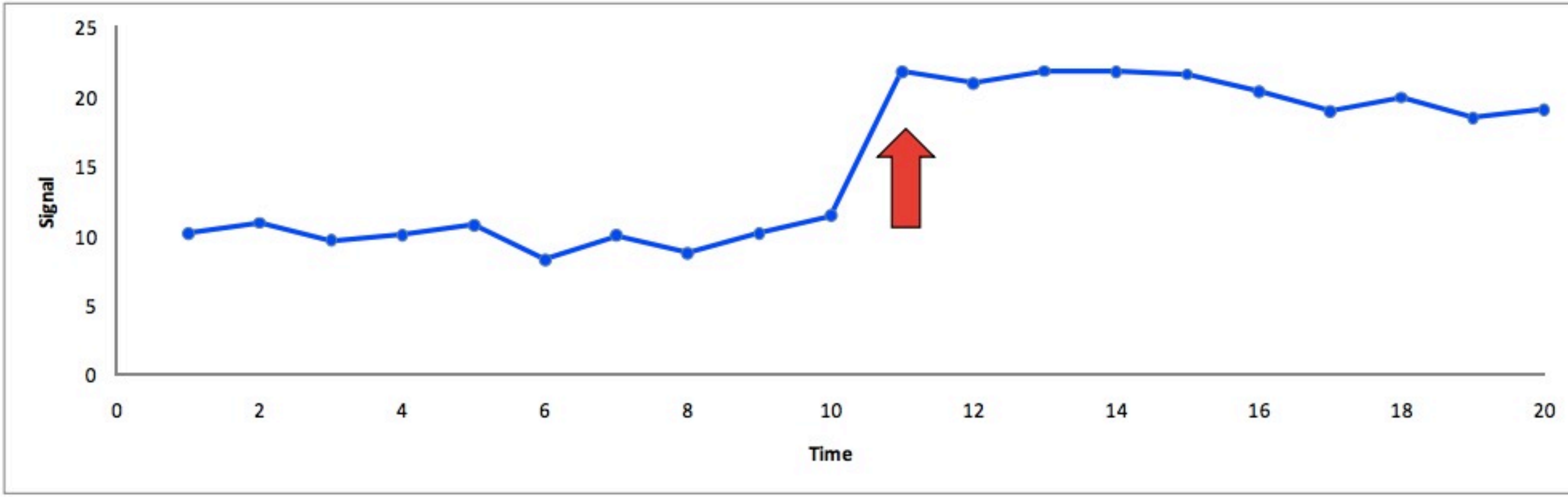


# Univariate Event Detection [Neill09]

This is a time series of counts of primary-physician visits in data from Norfolk in December 2001. I added a fake outbreak, starting at a certain date. Can you guess when?

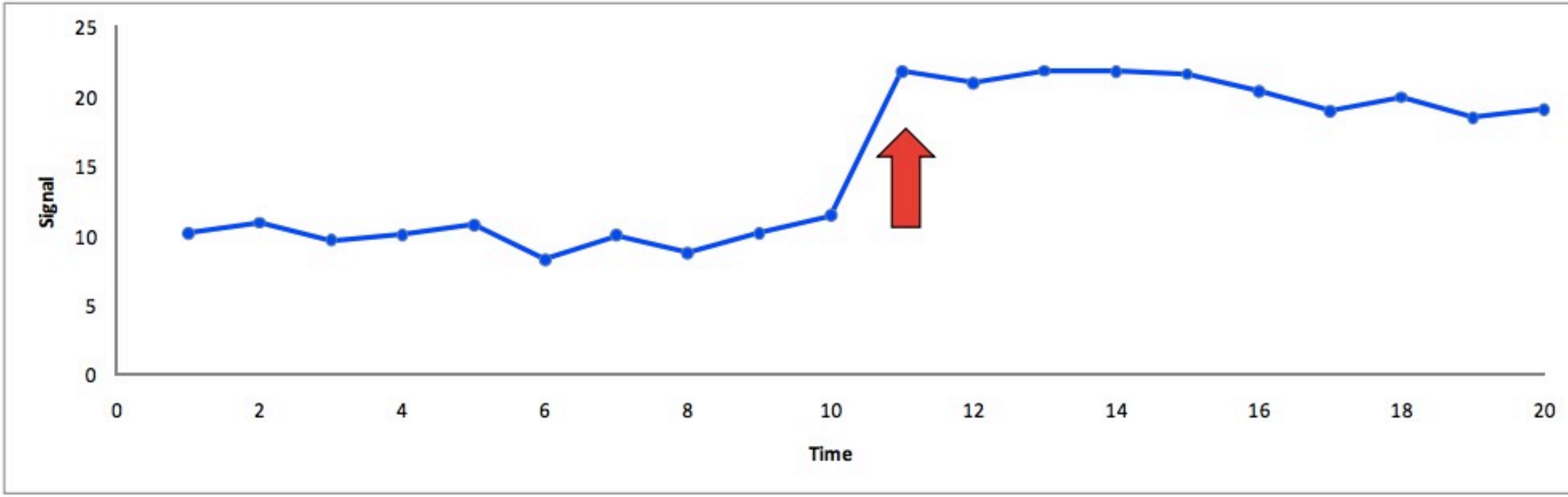


# Univariate Event Detection [Neill09]



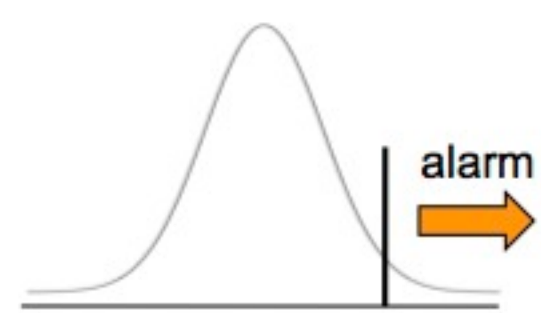

When does an event happen?

# Univariate Event Detection [Neill09]

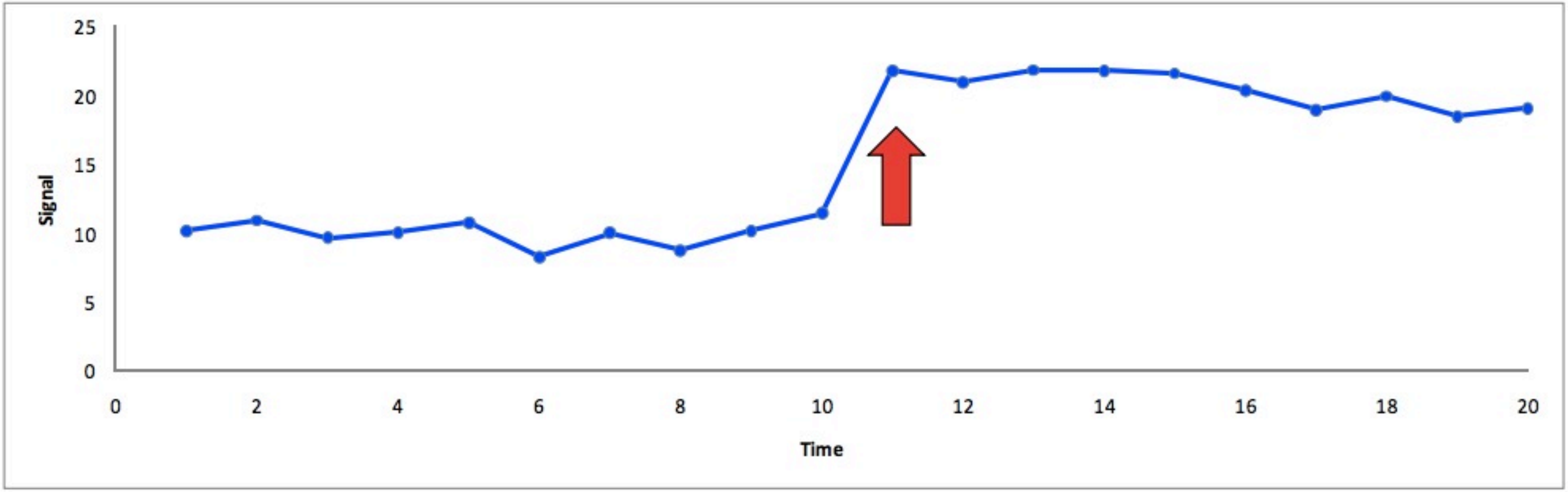



**Event Detection Framework**

1. Learn model to predict expected signal value
2. Measure difference between actual and expected
3. Define alert threshold



# Univariate Event Detection [Neill09]




**Event Detection Techniques**

1. Control Charts [Shewhart31]
2. **Moving Average** [Roberts59]
3. CUSUM [Page54]
4. **Regression** [Montgomery01]

# Univariate Event Detection: Moving Average

Let W be the window size

A moving average window predicts the following:

$$X_{t+1} = \frac{1}{W}(X_t + X_{t-1} + \ldots + X_{t-W-1})$$

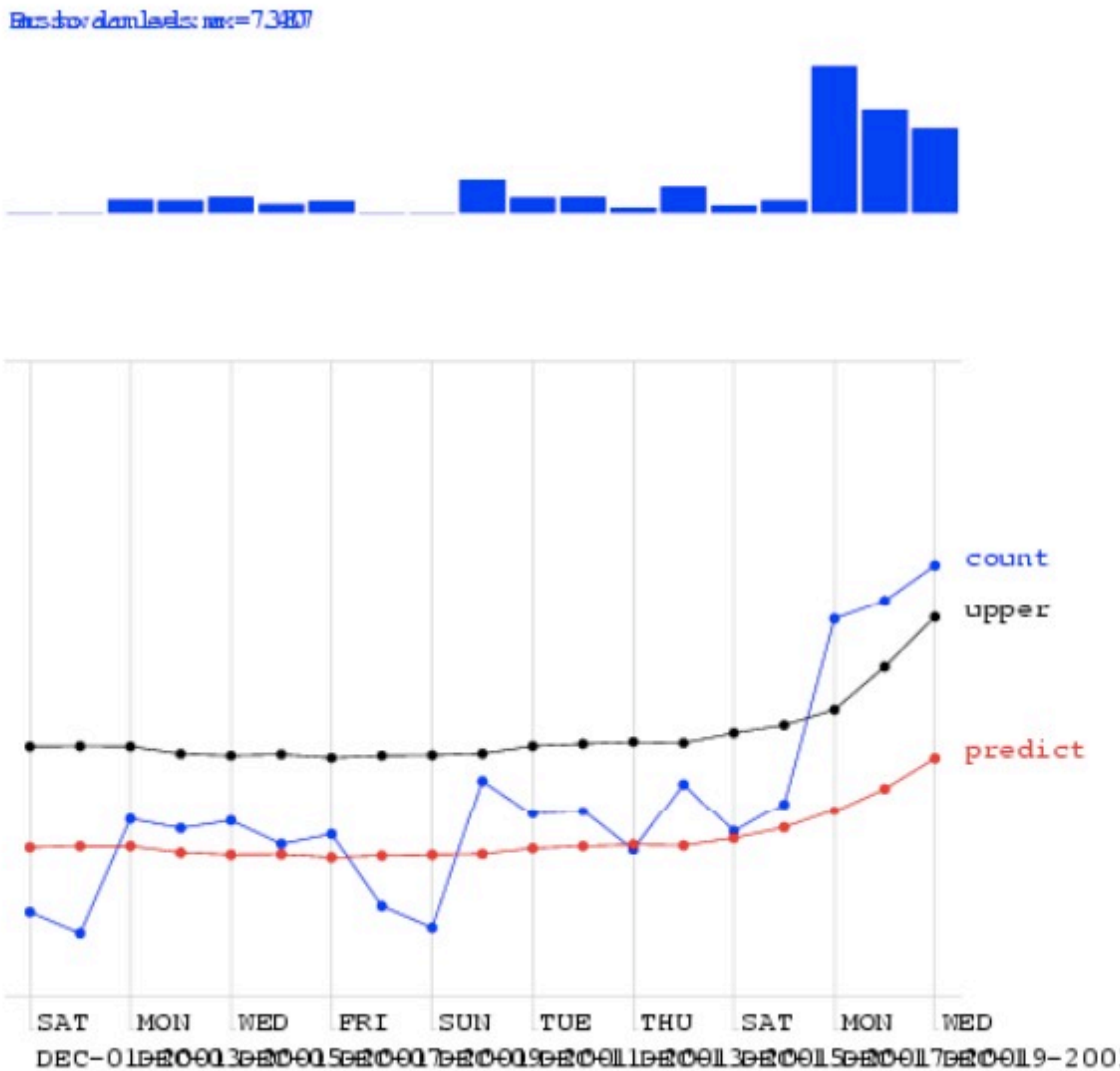
Setting the alarm value:

Fit a Gaussian to the W observations within the window ie. estimate $\hat{\mu}$ and $\hat{\sigma}$

Calculate the alarm level as before

$$\text{Alarm level} = \Phi\left(\frac{\max(0, X_i - \hat{\mu})}{\hat{\sigma}}\right) \quad \text{where } \Phi = \text{CDF for N}(0,1)$$

# Univariate Event Detection: Moving Average



# Problems?

Data often contains trends

- Seasonal effect
- Holiday effect
- Day-night effect
- Day-of-week effect

**Regression methods** address this problem.

# Univariate Event Detection: Regression

Regression example to model seasonal effects and Monday effects:

$$Y_i = \beta_0 + \beta_1(HoursOfDaylight_i) + \beta_2(IsMonday_i) + \varepsilon_i$$

Could be defined as:

$$\sin\left(\frac{2\pi(\text{num days since July 31})}{365.25} - \frac{\pi}{2}\right)$$

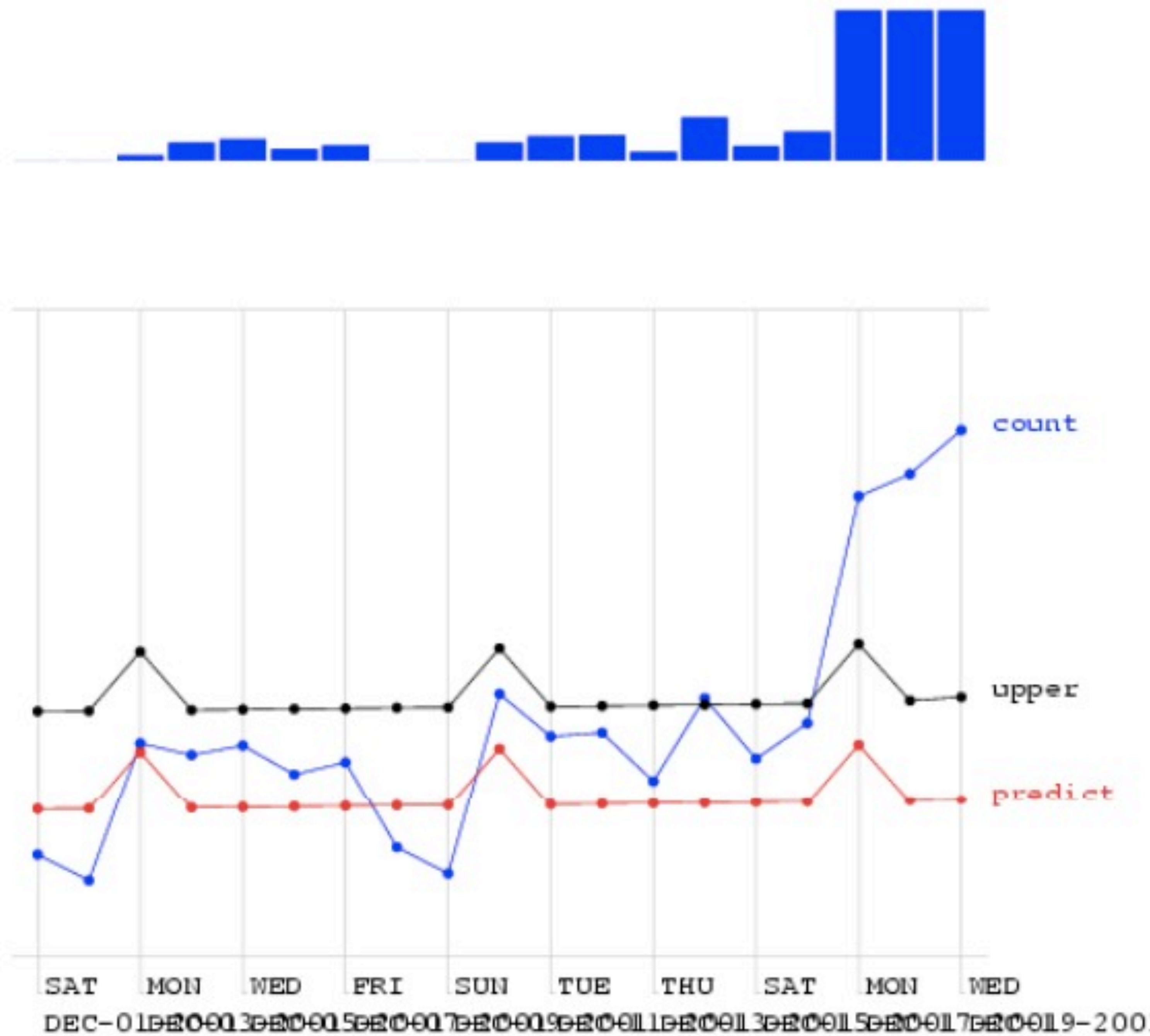
Boolean feature – adds a “bump” to the value of Y if it is a Monday

Normally distributed noise with mean 0, known variance $\sigma^2$

Regression learns the **β** parameters from data to minimize the residual sum of squares

# Univariate Event Detection: Regression

Regression applied to Norfolk data using *HoursOfDaylight* and *IsMonday* terms



# Event Detection Summary

**Process**

Learn data generation model, predict expected signal value, set alert threshold.

**Problems**

Complex data structures support, e.g., multivariate features, spatio-temporal data, graph data, etc.

# Conclusion

Data-driven security is an exciting research direction, given a large amount of operational data available

**Challenges**

**Robust learning under attack**

Tolerant with noises from attackers

**Over-fitting**

Predicting future attacks

**Online learning**

Learn as stream of data coming in

**Unsupervised learning**

Learn with no expert guidance