Statistical Approaches for Network Anomaly Detection

Christian CALLEGARI

Department of Information Engineering
University of Pisa

ICIMP Conference
9, May 2009
Barcelona
Spain

Short Bio

- Post-Doctoral Fellow with the Telecommunication Network research group at the Dept. of Information Engineering of the University of Pisa
- B.E. degree in 2002 from the University of Pisa, discussing a thesis on Network Firewalls
- M.S. degree in 2004 from the University of Pisa, discussing a thesis on Network Simulation
- PhD in 2008 from the University of Pisa, discussing a thesis on Network Anomaly Detection

Contacts

- Dept. of Information Engineering
- Via Caruso 16 - 56122 Pisa - Italy
- christian.callegari@iet.unipi.it

Acknowledgments

I'd like to thank some colleagues of mine for their contribution in support of this work:

- Michele Pagano (Associate Professor)
- Teresa Pepe (PhD Student)
- Loris Gazzarrini (Master Student)

What about you?

Outline

Outline

Why an intrusion detection system?

- Network security mainly means PREVENTION
  - Physical protection for hardware
  - Passwords, access tokens, etc. for *authentication*
  - *Access control list for* authorization
  - Cryptography for *secrecy*
  - *Backups and redundancy for* authenticity
  - . . . and so on

BUT . . .

. . . Absolute security cannot be guaranteed!

What is an Intrusion Detection System?

- Prevention is suitable when
  - Internal users are trusted
  - Limited interaction with other networks
- Need for a system which acts when prevention fails

Intrusion Detection System

An intrusion detection system (IDS) is a software/hardware tool used to detect unauthorized accesses to a computer system or a network

A taxonomy of the intruders

Intruders can be classified as

- **Masquerader**: an individual who is not authorized to use the computer and who penetrates a system's access control to exploit a legitimate user's account
- **Misfeasor**: a legitimate user who accesses data, programs, or resources for which such access is not authorized, or who is authorized for such access, but misuses his/her privileges
- **Clandestine User**: an individual who seizes supervisory control of the system and uses the control to evade auditing and access controls or to suppress audit collection

A taxonomy of the intrusions

Intrusions can be classified as

- **Eavesdropping and Packet Sniffing**: passive interception of network traffic
- **Snooping and Downloading**
- **Tampering and Data Diddling**: unauthorized changes to data or records
- **Spoofing**: impersonating other users
- **Jamming or Flooding**: overwhelming a system's resources
- **Injecting Malicious Code**
- **Exploiting Design or Implementation Flaws** (e.g., buffer overflow)
- **Cracking Passwords and Keys**

IDS Taxonomy

Intrusion Detection Systems are classified on the basis of several criteria:

1. Scope
   - Host IDS (HIDS)
   - Network IDS (NIDS)
2. Architecture
   - Centralized
   - Distributed
3. Analysis Techniques
   - Stateful
   - Stateless
4. Detection Techniques
   - Misuse Based IDS
   - Anomaly Based IDS

Host based vs. Network based

Host based IDS

- Aimed at detecting attacks related to a specific host
- Architecture/Operating System dependent
- Processing of high level information (e.g. system calls)
- Effective in detecting insider misuse

Network based IDS

- Aimed at detecting attacks towards hosts connected to a LAN
- Architecture/Operating System independent
- Processing data at lower level of granularity (packets)
- Effective in detecting attacks from the “outside”

Centralized IDS vs. Distributed IDS

Centralized IDS

- All the operations are performed by the same machine
- More simple to realize
- Only one point of failure

Distributed IDS

- Composed of several components
  - **Sensors** which generate security events
  - **Console** to monitor events and alerts and control the sensors
  - Central **Engine** that records events and generate alarms
- May need to deal with different data formats
- Need of a secure communication protocol (IPFIX)

Stateless IDS vs. Stateful IDS

Stateless IDS

- Treats each event independently of the others
- Simple system design
- High processing speed

Stateful IDS

- Maintains information about past events
- The effect of a certain event depends on its position in the events stream
- More complex system design
- More effective in detecting distributed attacks

Misuse based IDS vs. Anomaly based IDS

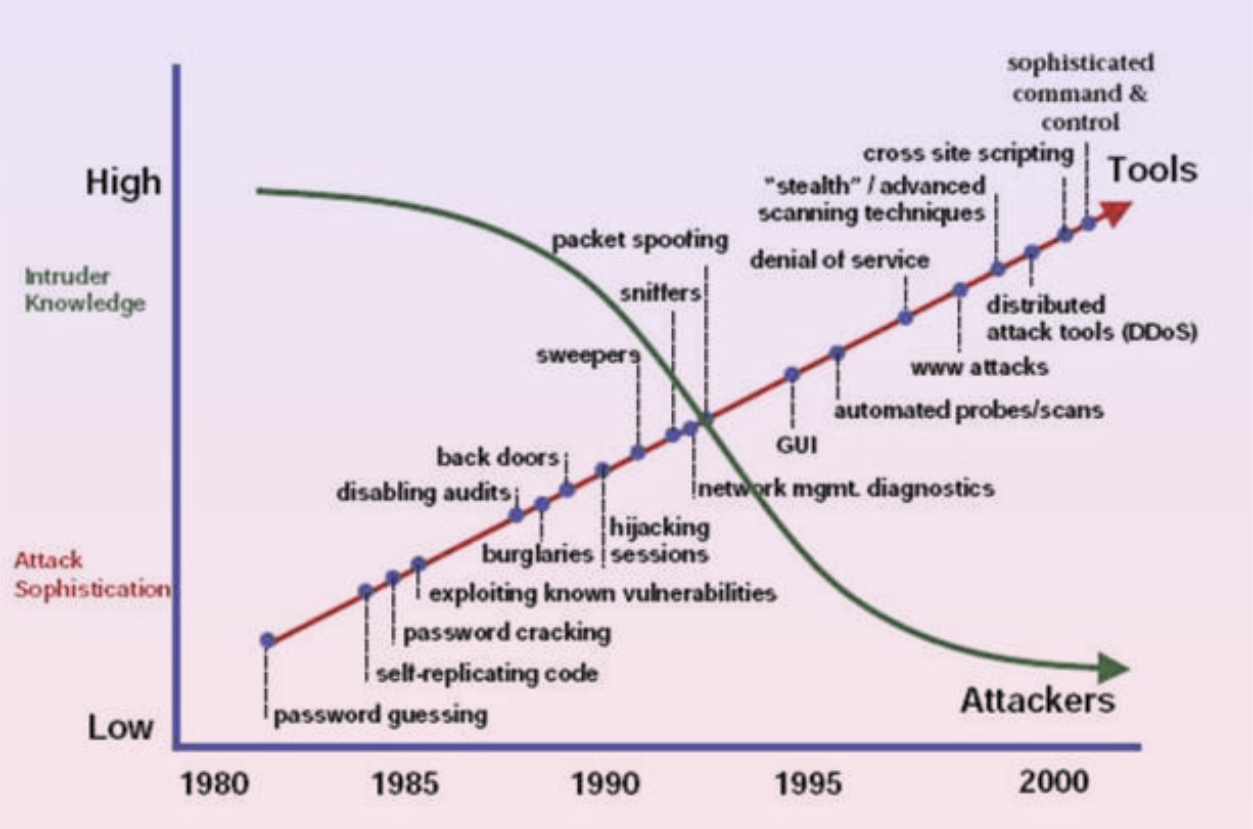
Misuse based IDS

- Identifies intrusion by looking for patterns of traffic or of application data presumed to be malicious
- Pattern of misuses are stored in a database
- Effective in detecting only “known” attacks

Anomaly based IDS

- Identifies intrusions by classifying activity as either anomalous or normal
- Needs a training phase to recognize normal activity
- Able to detect “new” attacks
- Generates more false alarms than a misuse based IDS

Attacks State of the Art



IDS State of the Art

- Focus is on Network based IDSs (The only ones effective in detecting Distributed Denial of Service - DDoS)
- State of the art IDSs are Misuse Based
  - Most attacks are realized by means of software tools available on the Internet
  - Most attacks are “well-known” attacks

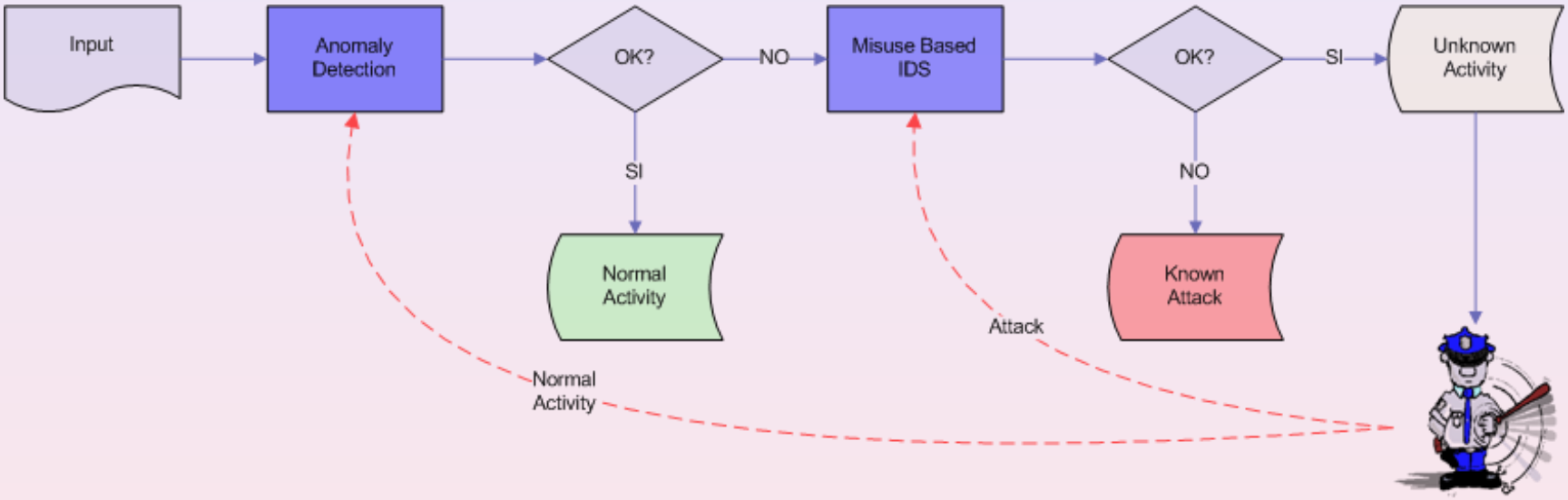
BUT . . .

. . . The most dangerous attacks are those written ad hoc by the intruder!

The best choice?

- Combined use of both
  - HIDS (for insider attacks) & NIDS (for outsider attacks)
  - Misuse IDS (low False Alarm rate) & Anomaly IDS (for "new" attacks)
  - Stateless IDS (fast data process) & Stateful IDS (for "complex" attacks)
- Distributed IDS
  - Not a single point of failure
  - More effective in monitoring large networks

The best choice?

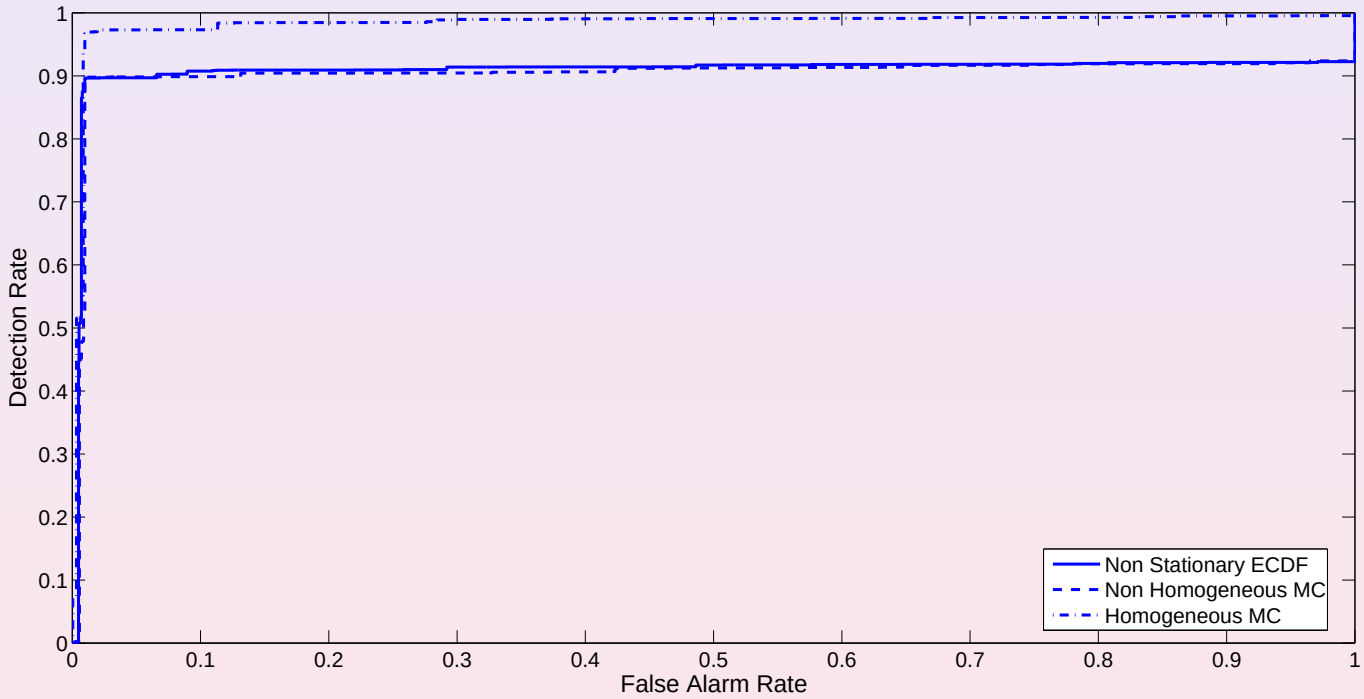


Definitions

- **False Positive (FP)**: the error of rejecting a null hypothesis when it is actually true. In our case it implies the creation of an alarm in correspondence of normal activities
- **False Negative (FN)**: the error of failing to reject a null hypothesis when it is in fact not true. In our case it corresponds to a missed detection

ROC Curve

Plots Detection Rate vs. False Positive Rate



ROC Curve

Results presented by the ROC are often considered incomplete because

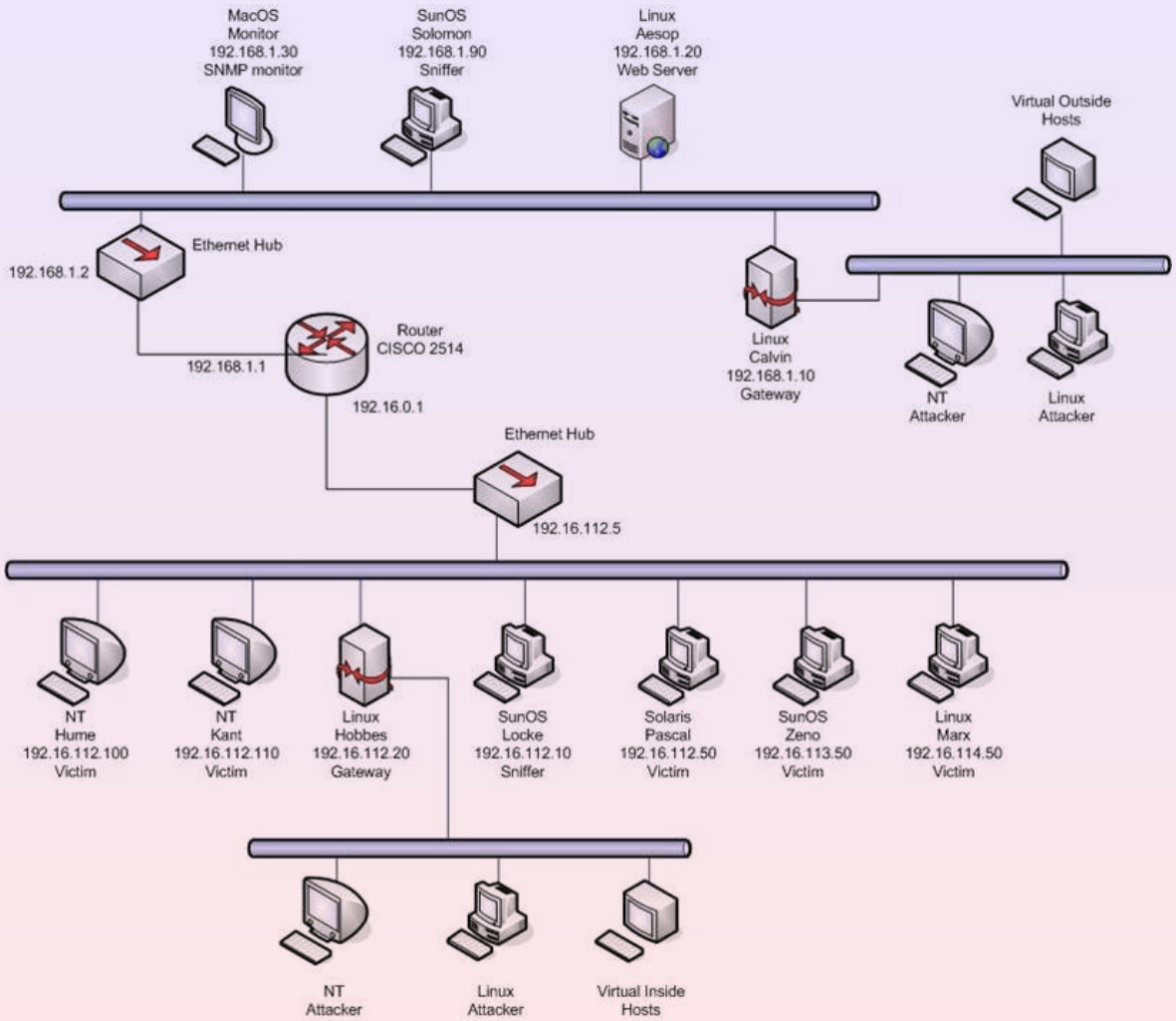
- they do not take into account the cost of missed attacks
- they do not take into account the cost of false alarms
- they do not say if the system itself is resistant to attacks
- ...

Several researchers are working on more complete ways of representing the results

DARPA Evaluation Program

- The 1998/1999 DARPA/MIT IDS evaluation program is the most comprehensive evaluation performed to date
- It provides a corpus of data for the development, improvement, and evaluation of IDSs
- Different kind of data are available:
  - Operating systems logs
  - Network traffic
    - Collected by an “inside” sniffer
    - Collected by an “outside” sniffer
- The data model the network traffic measured between a US Air Force base and the Internet

The DARPA Network



The DARPA Dataset

- 5 weeks data
  - Data from weeks 1 and 3 are attack free and can be used to train the system
  - Data from week 2 contains labeled attacks and can be used to realize the signatures database
  - Data from weeks 4 and 5 contains several attacks and can be used for the detection phase
- An Attack Truth list is provided
- Attacks are categorized as
  - Denial of Service (DoS)
  - User to Root (U2R)
  - Remote to Local (R2L)
  - Data
  - Probe
- 177 instances of 59 different types of attacks

Other Data-sets

The DARPA data-set has many drawbacks:

- simulated environment
- not up-to-date traffic
- the methodology used for generating the traffic has been shown to be inappropriate for simulating actual networks

Other Data-sets:

- several publicly available traffic traces
- e.g. CAIDA, Abilene (Internet2), GEANT, ...
- no ground truth is provided!

References

- *Anderson*, **Computer Security Monitoring and Surveillance**, Tech Rep 98-17, 1980
- *E. Millard* , **Internet attacks increase in number, severity**, Top Tech News, 2005
- *C. Staf* , **Hackers: companies encounter rise of cyber extortion**, vol. 2006, Computer Crime Research Center, 2005.
- *S. Axelsson* , **Intrusion Detection Systems: A Survey and Taxonomy**, Chalmers University, Technical Report 99-15, March 2000
- *W. Stallings* , **Cryptography and Network Security**, Prentice Hall
- *A. Patcha, J.M. Park* , **An overview of anomaly detection techniques: Existing solutions and latest technological trends**, Computer Networks 51, 2008

References

- **MIT, Lincoln laboratory, DARPA evaluation intrusion detection**, http://www.ll.mit.edu/IST/ideval/
- *R. Lippmann, J. Haines, D. Fried, J. Korba, and K. Das* , **The 1999 DARPA off-line intrusion detection evaluation**, Computer Networks 34, 2000
- *J. Haines, R. Lippmann, D. Fried, E. Tran, S. Boswell, and M. Zissman* , **1999 DARPA intrusion detection system evaluation: Design and procedures**, Tech. Rep. 1062, MIT Lincoln Laboratory, 2001
- *J. McHugh*, **Testing Intrusion detection systems: a critique of the 1998 and 1999 DARPA intrusion detection**, ACM Transactions on Information and System Security 3, 2000
- *Christian Callegari, Stefano Giordano, Michele Pagano*, **New Statistical Approaches for Anomaly Detection**, Security and Communication Networks, to appear

Outline

A bit of History

The history of IDSs can be split in three main blocks

1. First Generation IDSs (end of the 1970s)
   - The concept of IDS first appears in the 1970s and early 1980s (Anderson, Computer Security Monitoring and Surveillance, Tech Rep 1980)
   - Focus on audit data of a single machine
   - Post processing of data
2. Second Generation IDSs (1987)
   - Intrusion Detection Expert System (Denning, An intrusion Detection Model, IEEE Trans. on Soft. Eng., 1987)
   - Statistical analysis of data
3. Third Generation IDSs (to come)
   - Focus on the network
   - Real-time detection
   - Real-time reaction
   - Intrusion Prevention System

IDES

Model's components

- **Subjects**: initiators of activity on a target system
- **Objects**: resources managed by the system files, commands, etc.
- **Audit Records**: generated by the target system in response to actions performed or attempted by subjects
- **Profiles**: structures that characterize the behavior of subjects with respect to objects in terms of statistical metrics and models of observed activity
- **Anomaly Records**: generated when abnormal behavior is detected
- **Activity Rules**: actions taken when some condition is satisfied

Subjects and Objects

Subjects

- Initiators of actions on the target system
- It is typically a terminal user
- They can be grouped into different categories
- Users groups may overlap

Objects

- Receptors of subjects' actions
- If a subject is a recipient of actions (e.g. electronic mail), then is also considered to be a object
- Additional structures may be imposed (e.g. records may be grouped in database)
- Objects granularity depends on the environment

Audit Records

{Subject, Action, Object, Exception-Condition, Resource-Usage, Time-stamp}

- **Action**: operation performed by the subject on or with the object
- **Exception-Condition**: denotes which, if any, execution condition is raised on the return
- **Resource-Usage**: list of quantitative elements, where each element gives the amount of some resource
- **Time-stamp**: unique time/date stamp identifying when the action took place

Profiles

- An activity profile characterizes the behavior of a given subject (or set of subjects) with respect to a given object, thereby serving as a signature or description of normal activity for its respective subject and object
- Observed behavior is characterized in terms of a statistical metric and model
- A metric is a random variable x representing a quantitative measure accumulated over a period
- Observations x_i of x obtained from the audit records are used together with a statistical model to determine whether a new observation is abnormal
- The statistical models make no assumptions about the underlying distribution of x; all knowledge about x is obtained from the observations x_i

Metrics and Models

Metrics

- *Event counter*
- *Interval timer*
- *Resource measure*

Statistical models

- *Operational model*: abnormality is decided by comparison of x_n with a fixed threshold
- *Mean and standard deviation model*: abnormality is decided by checking if x_n falls inside the confidence interval
- *Multivariate model*: based on the correlations between two or more metrics
- *Markov process model*: based on the transition probabilities
- *Time series model*: takes into account order and inter-arrival time of the observations

Profile structure

{Variable-name, Action-pattern, Exception-pattern, Resource-usage-pattern, Period, Variable-type, Threshold, Subject-pattern, Object-pattern, Value}

- **Variable-name**
- **Action-pattern**: pattern that matches one or more actions in the audit records (e.g. “login”)
- **Exception-pattern**: pattern that matches on the Exception-condition field of an audit record
- **Resource-usage-pattern**: pattern that matches on the Resource-usage field of an audit record
- **Period**: time interval for measurements
- **Variable-type**: name of abstract data type that defines a particular type of metric and statistical model (e.g. event counter with mean and standard deviation model)
- **Threshold**

Profile structure

{Variable-name, Action-pattern, Exception-pattern, Resource-usage-pattern, Period, Variable-type, Threshold, Subject-pattern, Object-pattern, Value}

- **Subject-pattern**: pattern that matches on the Subject field of an audit record
- **Object-pattern**: pattern that matches on the Object field of an audit record
- **Value**: value of current observation and parameters used by the statistical model to represent distribution of previous values

There also is the possibility of defining profiles for classes

Profile templates

When user accounts and objects can be created dynamically, a mechanism is needed to generate activity profiles for new subjects and objects

- **Manual create**: the security officer explicitly creates all profiles
- **Automatic explicit create**: all profiles for a new user are generated in response to a “create” record in the audit trail
- **First use**: a profile is automatically generated when a subject (new or old) first uses an object (new or old)

Anomaly Records

{Event, Time-stamp, Profile}

- **Event**: indicates the event giving rise to the abnormality and is either “audit”, meaning the data in an audit record was found abnormal, or “period”, meaning the data accumulated over the current period was found abnormal
- **Time-stamp**: either the Time-stamp in the audit trail or interval stop time
- **Profile**: activity profile with respect to which the abnormality was detected

Activity Rules

A *condition that*, when satisfied, causes the rule to be fired, and a *body*, which specified the action to be taken

- **Audit-record rule**: triggered by a match between a new audit record and an activity profile, updates the profiles and checks for anomalous behavior
- **Periodic-activity-update rule**: triggered by the end of an interval matching the period component of an activity profile, updates the profiles and checks for anomalous behavior
- **Anomaly-record rule**: triggered by the generation of an anomaly record, brings the anomaly to the immediate attention of the security officer
- **Periodic-anomaly-analysis rule**: triggered by the end of an interval, generates summary reports of the anomalies during the current period

References

- *D. Denning* , **An intrusion detection model**, IEEE Transactions Software Engineering, vol. SE-13, no.2, 1987

Outline

Statistical Approach: Traffic Descriptors

The goal is to identify some traffic parameters, which can be used to describe the network traffic and that vary significantly from the normal behavior to the anomalous one

Some examples

- Packet length
- Inter-arrival time
- Flow size
- Number of packets per flow
- . . . and so on

Choice of the Traffic Descriptors

For each parameter we can consider

- Mean Value
- Variance and higher order moments
- Distribution function
- Quantiles
- . . . and so on

The number of potential traffic descriptors is huge (some papers identify up to 200 descriptors)

GOAL

To identify as few "**attack invariant**" descriptors as possible to classify traffic with an *acceptable* error rate

Outline

Clustering

- Clustering is the assignment of a set of observations into subsets (called clusters) so that observations in the same cluster are similar in some sense
- Clustering is a method of unsupervised learning
- The clusters are computed on the basis of a distance measure, which will determine how the similarity of two elements is calculated
- Common distances are:
  - Euclidean distance
  - Manhattan distance
  - Mahalanobis distance
  - ...

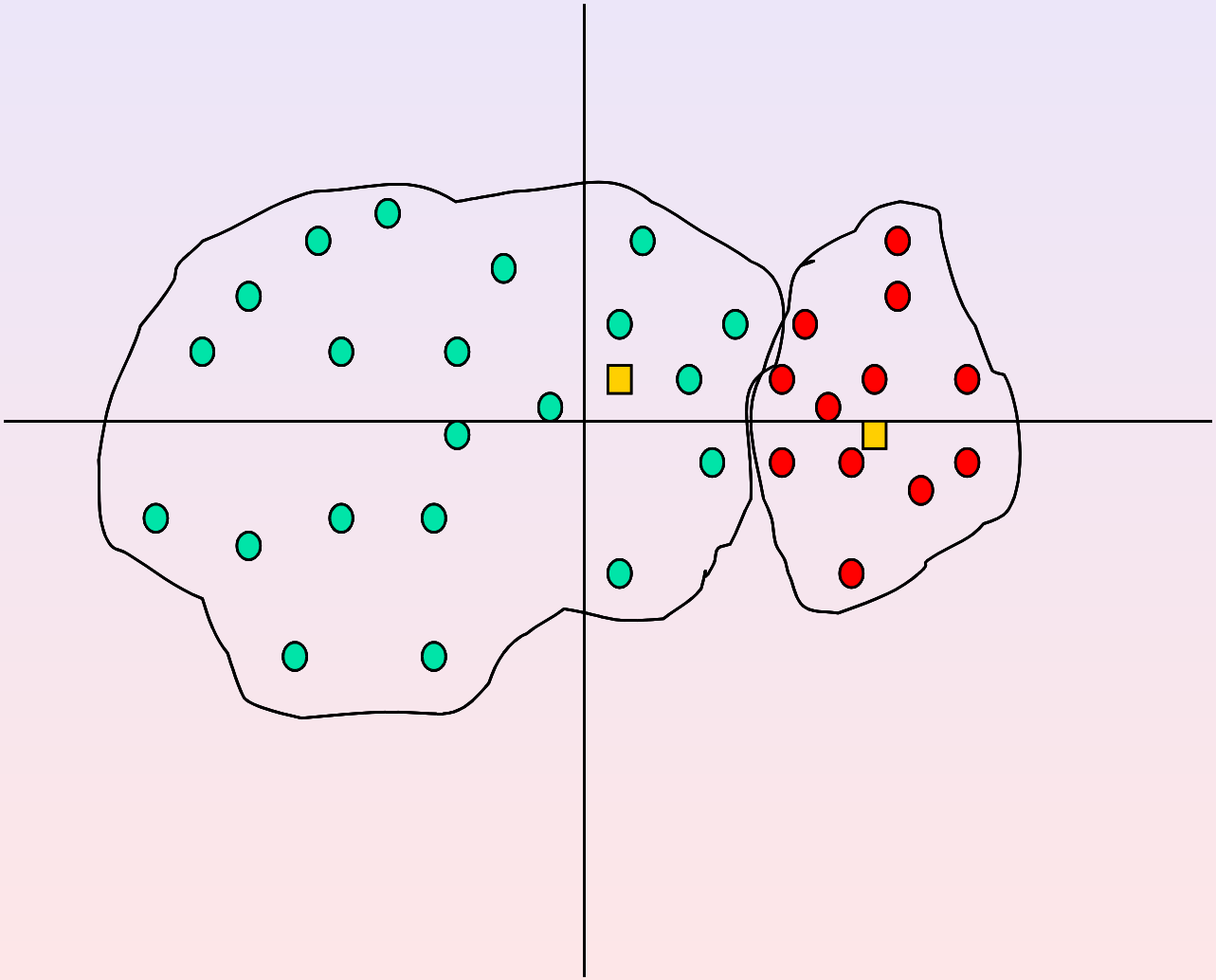
K-Means Algorithm

The k-means algorithm assigns each point to the cluster whose center (also called centroid) is the nearest

1. Choose the number of clusters, k
2. Randomly generate k clusters and determine the cluster centers, or directly generate k random points as cluster centers
3. Assign each point to the nearest cluster center
4. Recompute the new cluster centers
5. Repeat the two previous steps until some convergence criterion is met (e.g., the assignment hasn't changed)

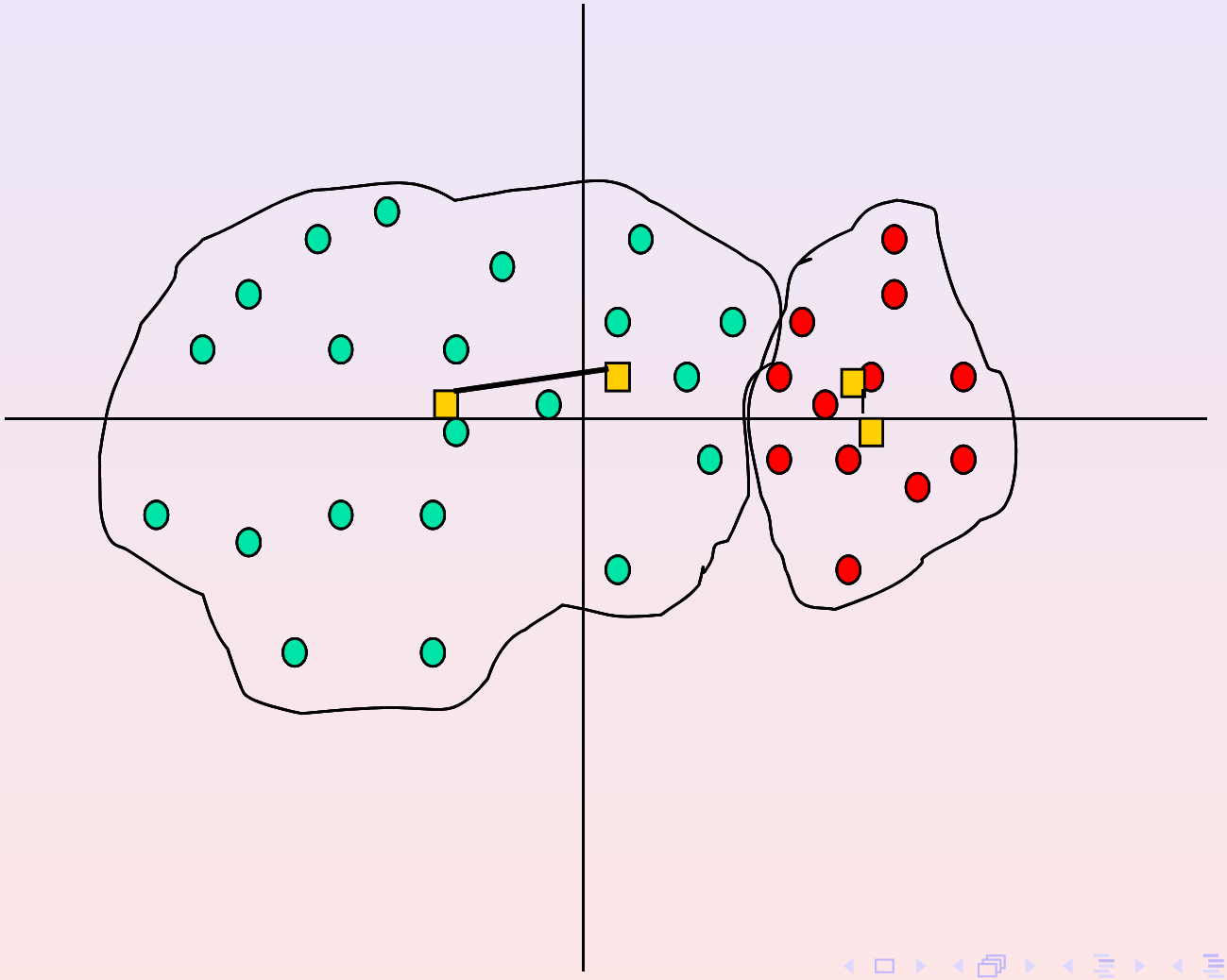
K-Means Algorithm - An example

Consider $k = 2$, choose 2 points (centroids), build 2 clusters



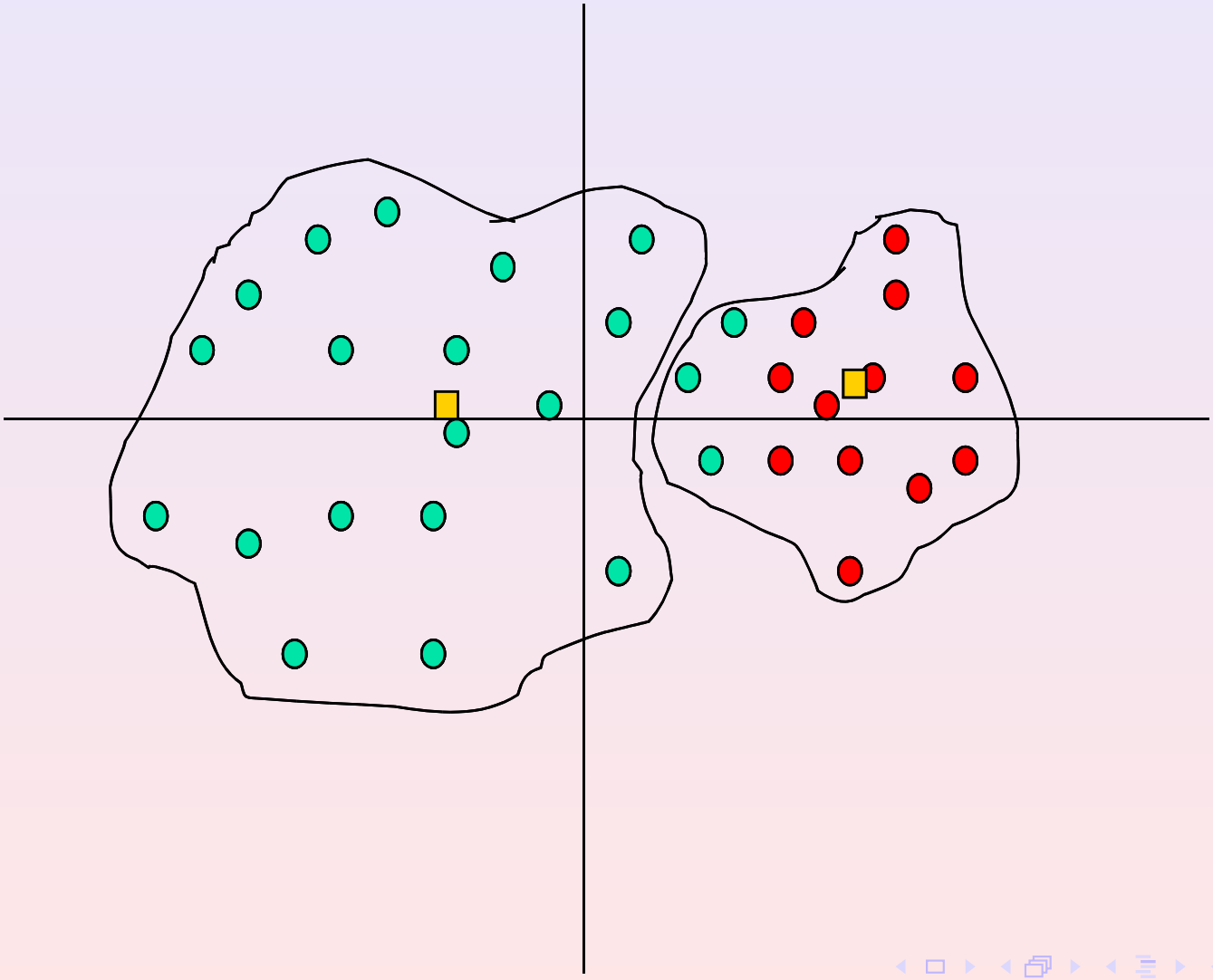
K-Means Algorithm - An example

Compute the new centroids



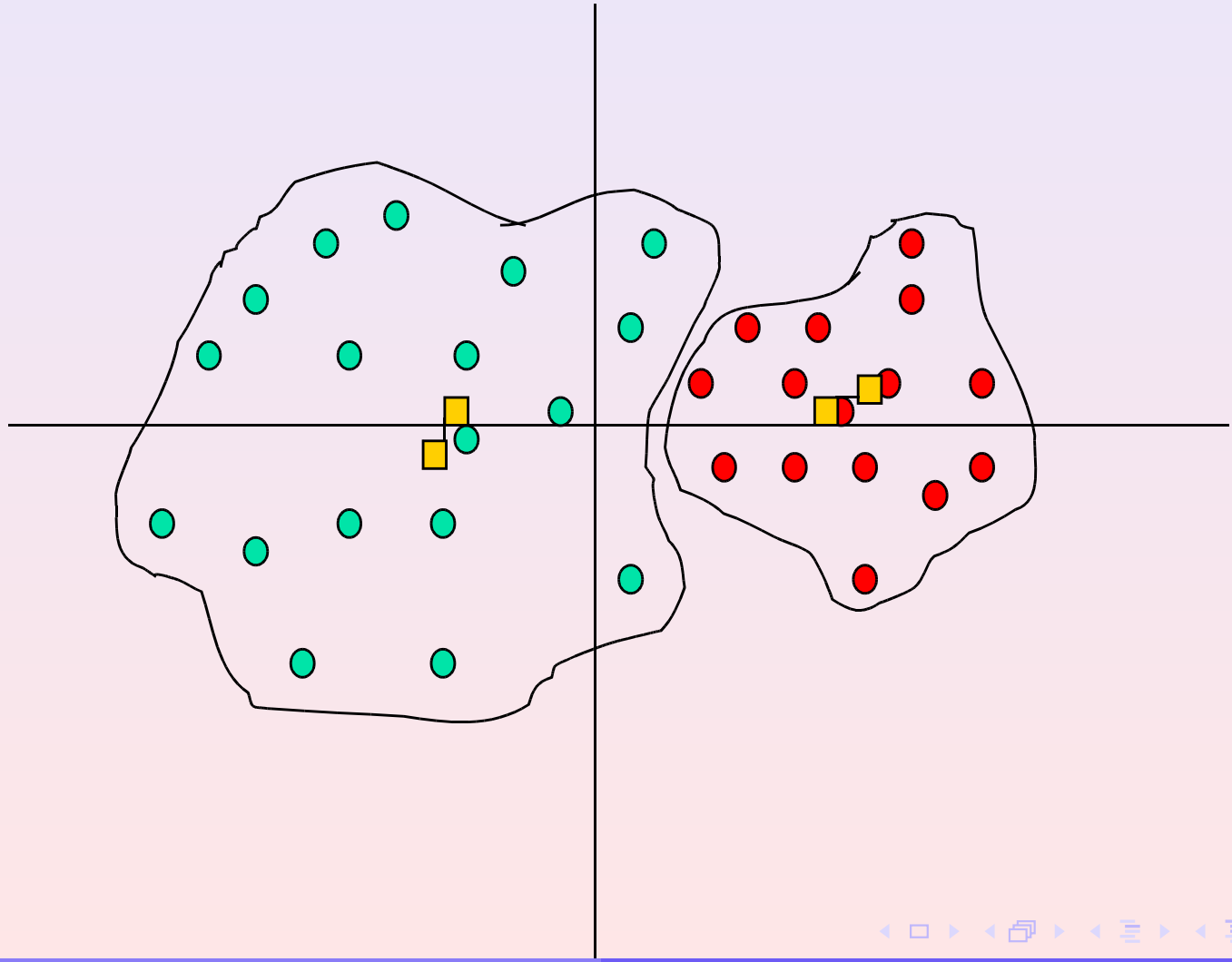
K-Means Algorithm - An example

Build the new clusters



K-Means Algorithm - An example

Repeat last 2 steps, until a assignments don't change



Outliers

- In statistics, an outlier is an observation that is numerically distant from the rest of the data
- Detection based on the full dimensional distances between the points as well as the densities of local neighborhoods
- There exist at least two approaches
  - the anomaly detection model is trained using unlabeled data that consist of both normal as well as attack traffic
  - the model is trained using only normal data and a profile of normal activity is created

Outliers Detection - Method 1

- The idea behind the first approach is that anomalous or attack data form a small percentage of the total data
- Anomalies and attacks can be detected based on cluster sizes
  - large clusters correspond to normal data
  - the rest of the data points, which are outliers, correspond to attacks

References

- *L. Portnoy, E. Eskin, S.J. Stolfo* , **Intrusion detection with unlabeled data using clustering**, ACM Workshop on Data Mining Applied to Security, 2001
- *S. Ramaswamy, R. Rastogi, K. Shim* , **Efficient algorithms for mining outliers from large data sets**, ACM SIGMOD International Conference on Management of Data, 2000
- *K. Sequeira, M. Zaki* , **ADMIT: Anomaly-based data mining for intrusions**, ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2002
- *V. Barnett, T. Lewis* , **Outliers in Statistical Data**, Wiley, 1994

References

- *C.C. Aggarwal, P.S. Yu* , **Outlier detection for high dimensional data**, ACM SIGMOD International Conference on Management of Data, 2001
- *M. Breunig, H.-P. Kriegel, R.T. Ng, J. Sander* , **LOF: identifying density-based local outliers**, ACM SIGMOD International Conference on Management of Data, 2000
- *E.M. Knorr, R.T. Ng* , **Algorithms for mining distance-based outliers in large datasets**, International Conference on Very Large Data Bases, 2008
- *P.C. Mahalanobis* , **On tests and measures of groups divergence**, Journal of the Asiatic Society of Bengal 26, 1930

Outline

State Transition Analysis

- The approach was first proposed by Denning and developed in the 1990s.
- Mainly used in two distinct environment
  - **HIDS**: to model the sequence of system commands used by a user
  - **NIDS**: to model the sequence of some specific fields of the packet (e.g. the sequence of the flags values in a TCP connection)
- The most classical approach: **Markov chains**

Markov Chains and TCP

- Idea: Model TCP connections by means of Markov chains
- The IP addresses and the TCP port numbers are used to identify a connection
- State space is defined by the possible values of the TCP flags
- The value of the flags is used to identify the chain transitions
- A value S_p is associated to each packet according to the rule

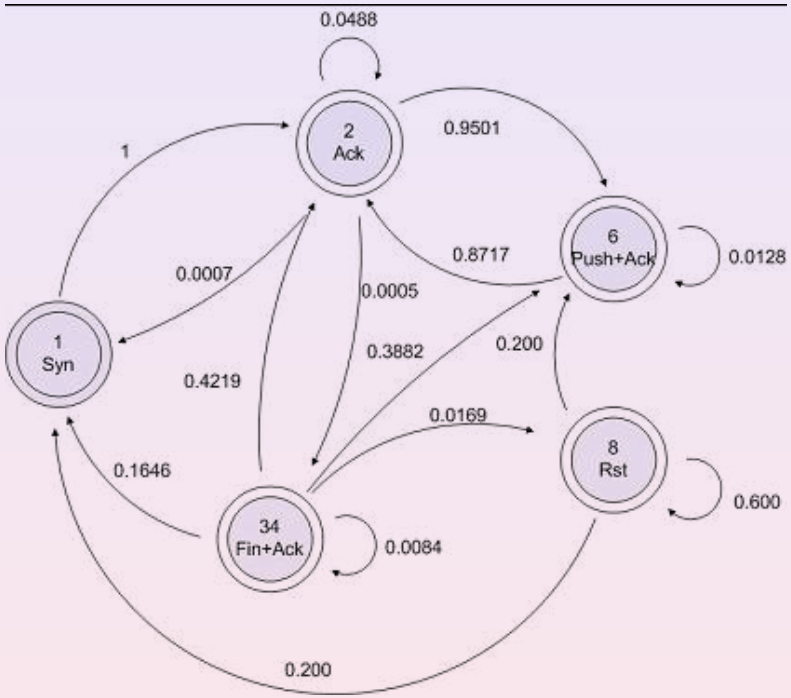
$$S_p = syn + 2 \cdot ack + 4 \cdot psh + 8 \cdot rst + 16 \cdot urg + 32 \cdot fin$$

Markov Chain and TCP - Training phase

Calculate the transition probabilities

$$a_{ij} = P[q_{t+1} = j | q_t = i] = \frac{P[q_t = i, q_{t+1} = j]}{P[q_t = i]}$$

- Server side
- 3-way handshake
- psh flag
- closing



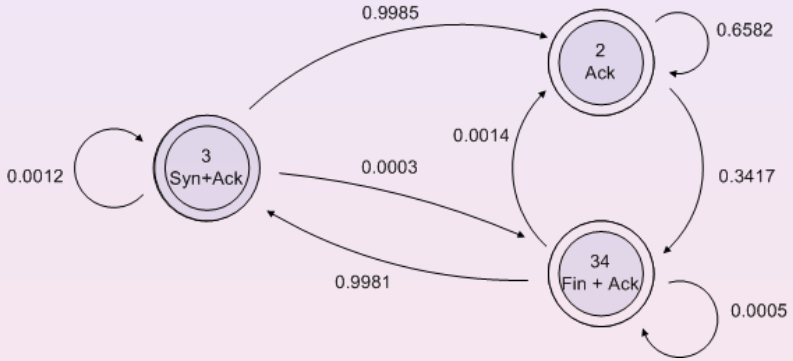
SSH Markov Chain

Markov Chain and TCP - Training phase

Calculate the transition probabilities

$$a_{ij} = P[q_{t+1} = j | q_t = i] = \frac{P[q_t = i, q_{t+1} = j]}{P[q_t = i]}$$

- Client side
- 3-way handshake
- ack flag
- closing

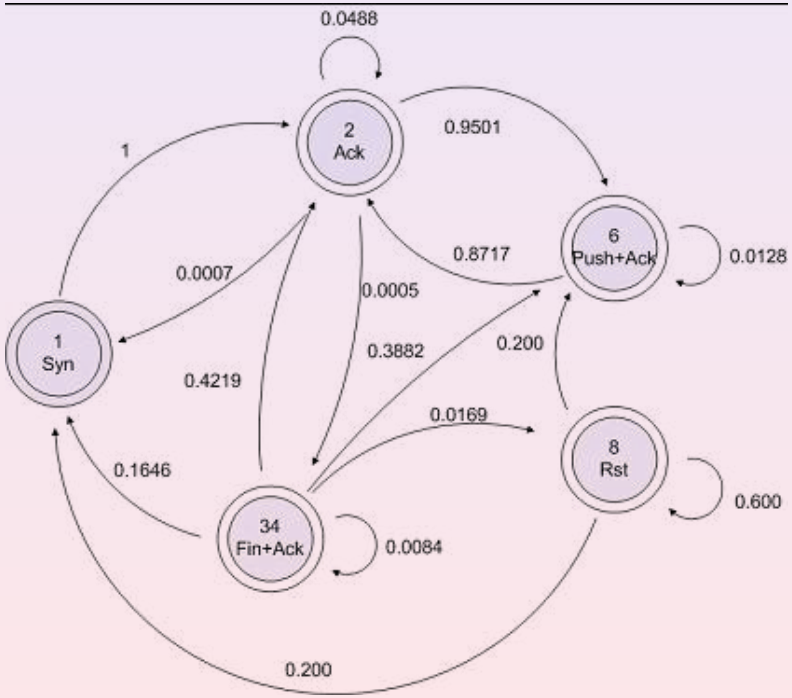


FTP Markov Chain

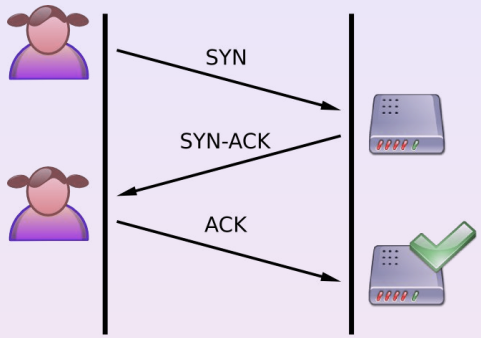
Markov Chain and TCP - Training phase

Calculate the transition probabilities

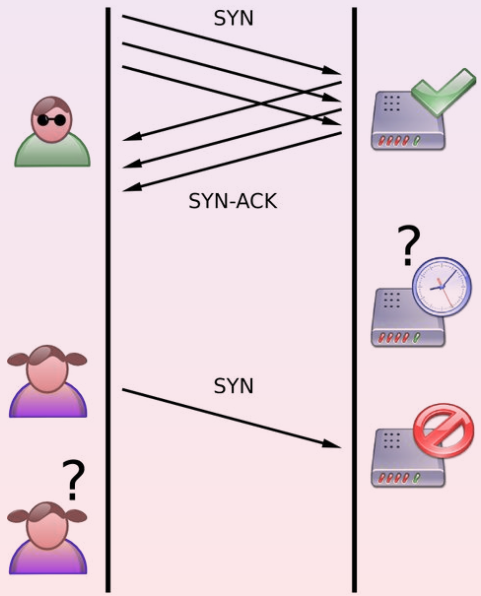
$$a_{ij} = P[q_{t+1} = j | q_t = i] = \frac{P[q_t = i, q_{t+1} = j]}{P[q_t = i]}$$



SSH Markov Chain



3-Way Handshake



Syn Flood Attack

Markov Chain and TCP - Detection phase

- Given the observation $(S_1, S_2, \cdots, S_T)$
- The system has to decide between two hypothesis

$$\begin{array}{c} H_0 : \textit{ normal behaviour} \\ H_1 : \textit{ anomaly} \end{array} \tag{1}$$

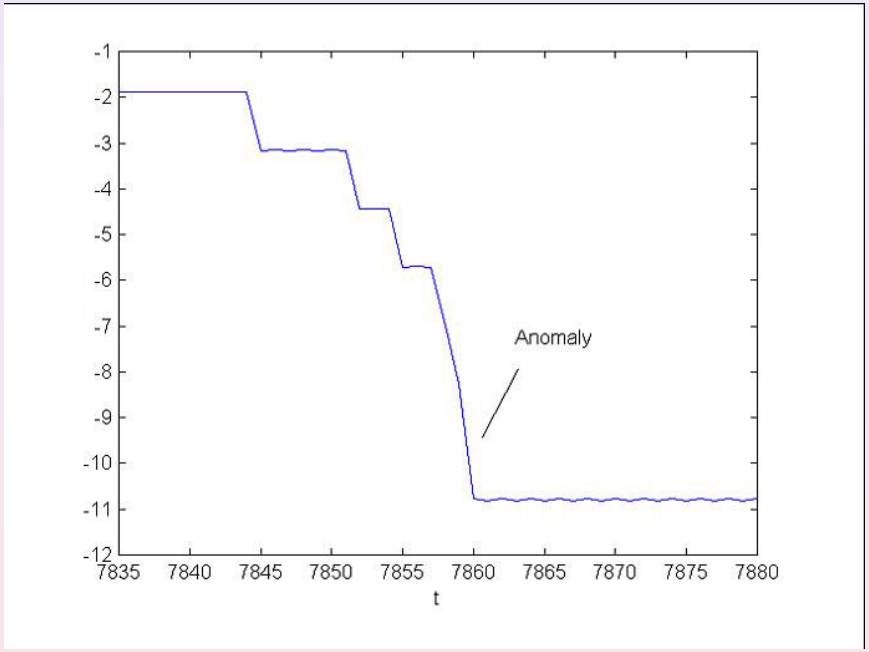
- A possible statistic is given by the logarithm of the Likelihood Function

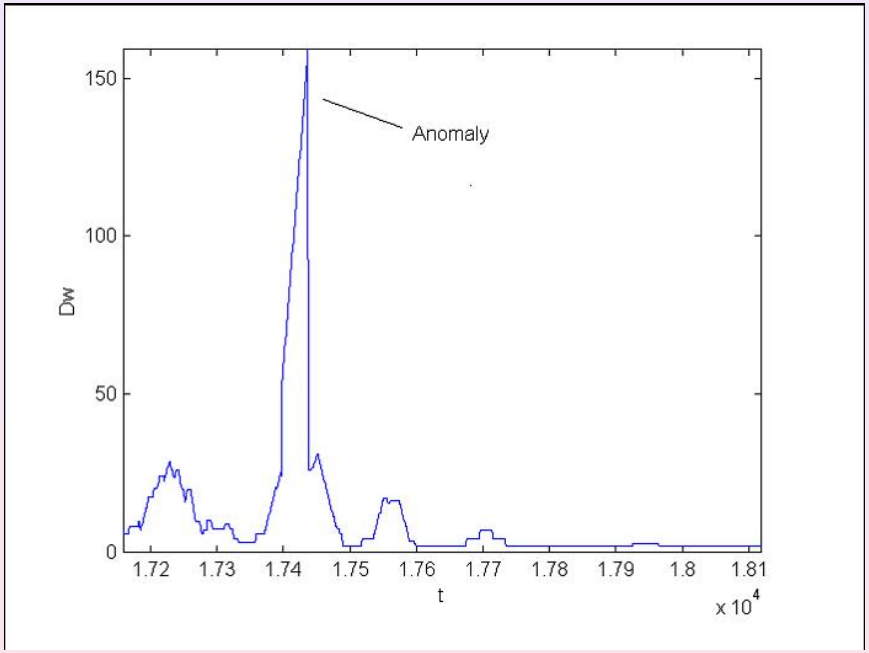
$$LogLF(t) = \sum_{t=R+1}^{T+R} Log(a_{S_t S_{t+1}})$$

- Or by its temporal "derivative"

$$D_w(t) = \left| LogLF(t) - \frac{1}{W} \sum_{i=1}^{W} LogLF(t-i) \right|$$

Markov Chain and TCP - Detection phase





Non Homogeneous Markov Chain

- First order homogeneous Markov chain

$$P(C_t = s_{i_0} | C_{t-1} = s_{i_1}, C_{t-2} = s_{i_2}, C_{t-3} = s_{i_3}, \cdots) =$$
$$P(C_t = s_{i_0} | C_{t-1} = s_{i_1}) = P(C_0 = s_{i_0} | C_{-1} = s_{i_1}) =$$
$$P(s_{i_0} | s_{i_1})$$

- First order non-homogeneous Markov chain

$$P(C_t = s_{i_0} | C_{t-1} = s_{i_1}, C_{t-2} = s_{i_2}, C_{t-3} = s_{i_3}, \cdots) =$$
$$P(C_t = s_{i_0} | C_{t-1} = s_{i_1}) =$$
$$P_t(s_{i_0} | s_{i_1})$$

- We build a distinct Markov Chain for each connection step (first 10 steps)
- The model should better characterizes the setup and the release phases

High order Markov Chain

- First order homogeneous Markov chain

$$P(C_t = s_{i_0} | C_{t-1} = s_{i_1}, C_{t-2} = s_{i_2}, C_{t-3} = s_{i_3}, \cdots) =$$
$$P(C_t = s_{i_0} | C_{t-1} = s_{i_1}) = P(C_0 = s_{i_0} | C_{-1} = s_{i_1}) =$$
$$P(s_{i_0} | s_{i_1})$$

- l^{th} order homogeneous Markov chain

$$P(C_t = s_{i_0} | C_{t-1} = s_{i_1}, C_{t-2} = s_{i_2}, C_{t-3} = s_{i_3}, \cdots) =$$
$$P(C_t = s_{i_0} | C_{t-1} = s_{i_1}, C_{t-2} = s_{i_2}, \cdots, C_{t-l} = s_{i_l}) =$$
$$P(C_0 = s_{i_0} | C_{-1} = s_{i_1}, C_{-2} = s_{i_2}, \cdots, C_{-l} = s_{i_l}) =$$
$$P(s_{i_0} | s_{i_1}, s_{i_2}, \cdots, s_{i_l})$$

- Some connection phases have dependences, between packets, of order bigger than 1

Mixture Transition Distribution

- We have an explosion of the number of the chain parameters, which grows exponentially with the order $(K^l(K-1))$
- Parsimonious representation of the transition probabilities
- Mixture Transition Distribution (MTD) model $(K(K-1)+l-1)$

$$P(C_t = s_{i_0} | C_{t-1} = s_{i_1}, C_{t-2} = s_{i_2}, \cdots, C_{t-l} = s_{i_l}) = \sum_{j=1}^{l} \lambda_j r(s_{i_0} | s_{i_j})$$

  - where the quantities $\boldsymbol{R} = \{r(s_i|s_j);\ i,j = 1,2,\cdots,K\}$ and $\boldsymbol{\Lambda} = \{\lambda_j;\ j = 1,2,\cdots,l\}$
  - satisfy the constraints

$$r(s_i|s_j) \geq 0;\ i,j = 1,2,\cdots,K \text{ and } \sum_{s_i=1}^{K} r(s_i|s_j) = 1\ \forall j = 1,2,\cdots,K$$

$$\lambda_j \geq 0;\ j = 1,2,\cdots,l \sum_{j=1}^{l} \lambda_j = 1$$

State Space Reduction

- We only consider the states observed during the training phase
- We add a *rare state* to take into account all the other possible states
- We fix the following quantities:

$$r(rare|s_i) = \epsilon \qquad \forall i = 1, 2, \cdots, K$$

with ϵ small (in our case $\epsilon = 10^{-6}$)

$$r(s_i|rare) = (1 - \epsilon)/(K - 1)$$
$$\forall i = 1, 2, \cdots, K - 1 \tag{2}$$

Parameters Estimation

- We need to estimate the parameters of the Markov chain (Maximum Likelihood Estimation - MLE)
- According to the MTD model, the log-likelihood of a sequence $(c_1, c_2, \cdots, c_T)$ of length T is:

$$LL(c_1, c_2, \cdots, c_T) = \sum_{i_0=1}^{K} \cdots \sum_{i_l=1}^{K} N(s_{i_0}, s_{i_1}, \cdots, s_{i_l}) \cdot \log\left(\sum_{j=1}^{l} \lambda_j r(s_{i_0} | s_{i_j}) \right)$$

  where $N(s_{i_0}, s_{i_1}, \cdots, s_{i_l})$ represents the number of times the transition $s_{i_l} \rightarrow s_{i_{l-1}} \rightarrow \cdots \rightarrow s_{i_0}$ is observed
- We have to maximize the right hand side of the equation, with respect to R and Λ, taking into account the given constraints

Parameters Estimation

Estimation Steps

- We apply an alternate maximization with respect to R and to Λ
- In the first step (estimation of Λ) we use the sequential quadratic programming
- The second step (estimation of R) is a linear inverse problem with positivity constraints (LININPOS) that we solve applying the Expectation Maximization (EM) algorithm

Global Maximum

This process leads to a global maximum, since LL is concave in $\boldsymbol{R}$ and $\boldsymbol{\Lambda}$.

Markov Chains - Detection Phase

- Choose between a single hypothesis H_0 (estimated stochastic model), and the composite hypothesis H_1 (all the other possibilities)

$$H_0 : \{(c_1, c_2, \cdots, c_T) \sim \textit{computed model } MC_0\}$$

$$H_1 : \{\textit{anomaly}\}$$

- No optimal result is presented in the literature
- The best solution is represented by the use of the Generalized Likelihood Ratio (GLR) test:

$$X = \left(\frac{Max_{v \neq u} L(c_1, c_2, \cdots, c_T | \Lambda_v, R_v)}{L(c_1, c_2, \cdots, c_T | \Lambda_u, R_u)} \right)^{\frac{1}{T}} \underset{H_1}{\overset{H_0}{\lessgtr}} \xi$$

Markov Chains - Detection Phase

- Equivalent to decide on the basis of the Kullback-Leibler divergence between the model associated to H_0 (MC_0) and the one computed for the observed sequence (MC_s)
- The Kullback-Leibler divergence, for first order Markov chains, is defined as:

$$KL(\mathrm{MC}_0, \mathrm{MC}_s) = \sum_i \sum_j \pi_0(s_i) P_0(s_j|s_i) \log \frac{P_0(s_j|s_i)}{P_s(s_j|s_i)}$$

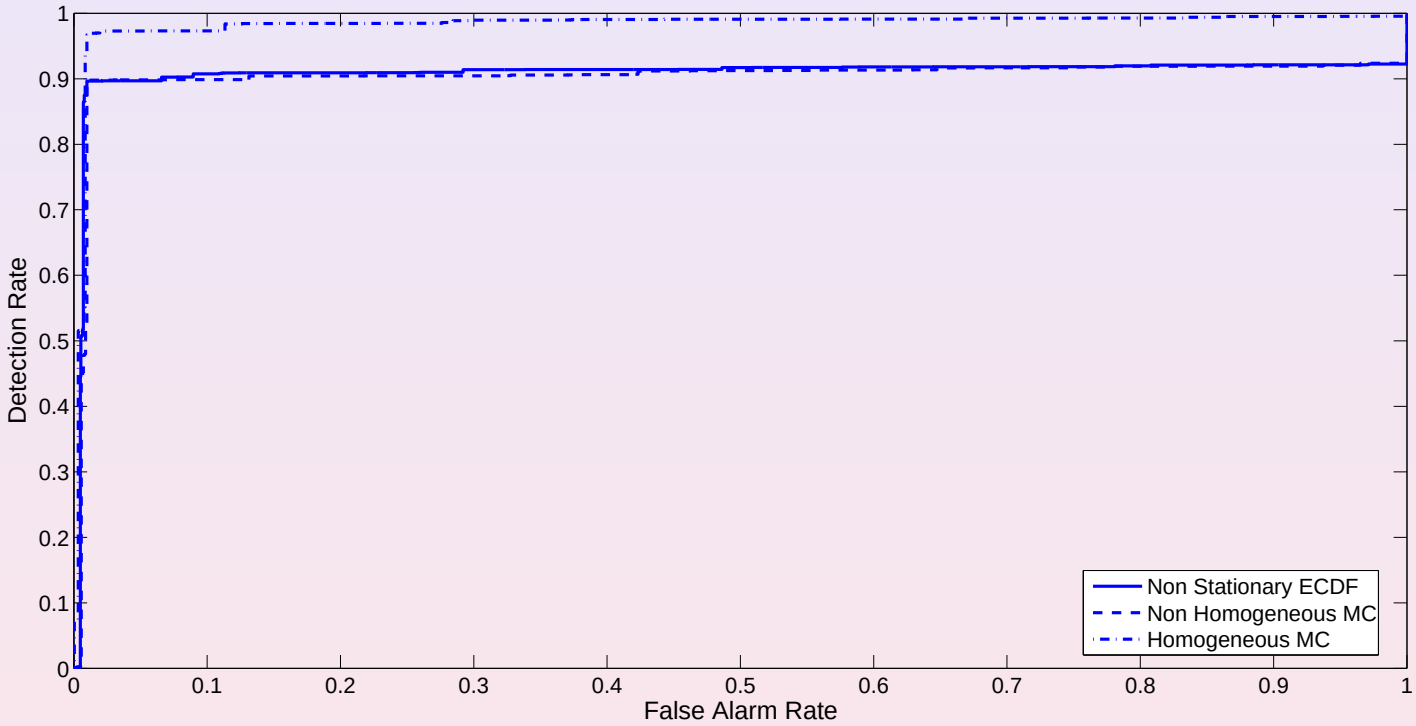
where $\pi_0(s_i)$ is the stationary distribution of MC_0 and $P_k(s_j|s_i)$ is the (single step) transition probability from state $C_{t-1} = s_i$ to state $C_t = s_j$

Extension to Markovian models of order l

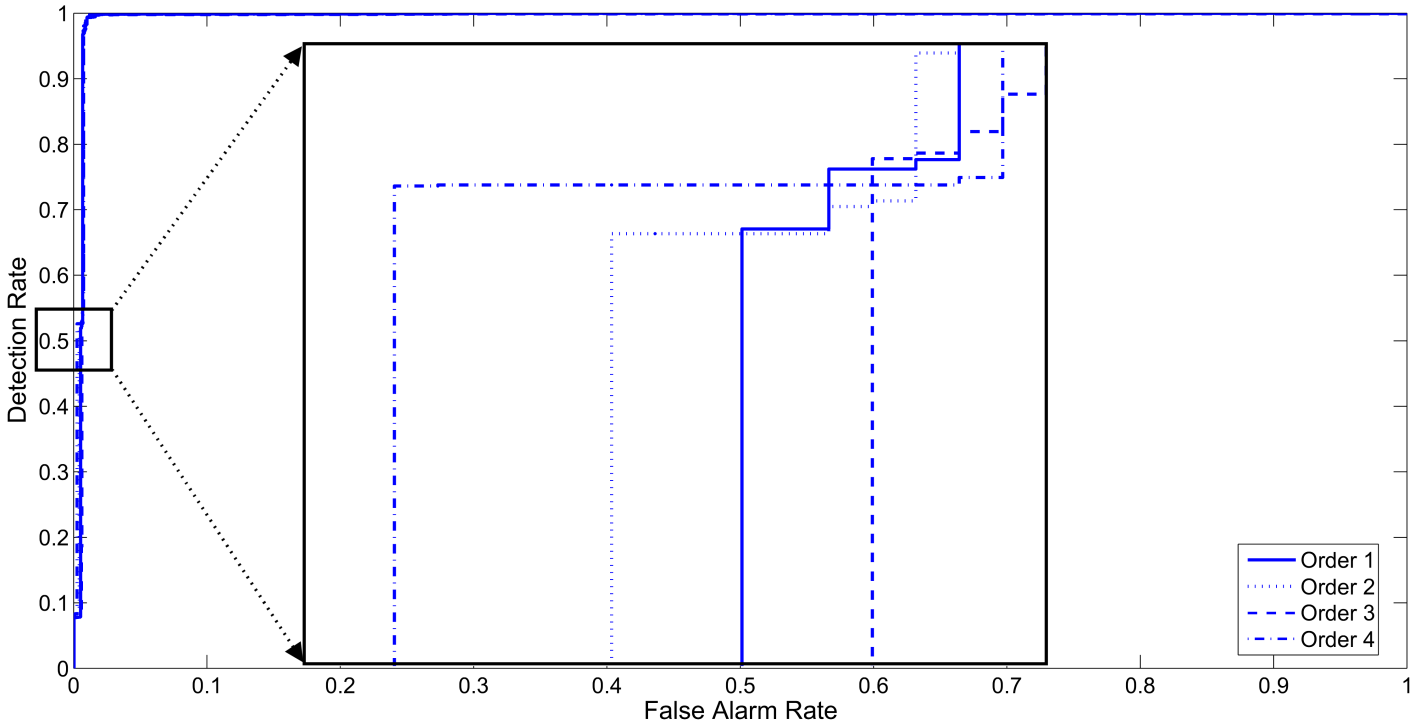
The state of the chain C_t has to be considered as a point in a finite l-dimensional lattice:

$$C_t = (C_t, C_{t-1}, \ldots, C_{t-l+1})$$

Non-Homogeneous Markov Chain



High Order Markov Chain



References

- *N. Ye, Y.Z.C.M Borror*, **Robustness of the Markov-chain model for cyber-attack detection**, IEEE Transactions on Reliability 53, 2004
- *D.-Y. Yeung, Y. Ding* , **Host-based intrusion detection using dynamic and static behavioral models**, Pattern Recognition 36, 2003
- *W.-H. Ju and Y. Vardi* , **A hybrid high-order Markov chain model for computer intrusion detection**, Tech. Rep. 92, NISS, 1999
- *M. Schonlau, W. DuMouchel, W.-H. Ju, A. Karr, M. Theus, and Y. Vardi* , **Computer intrusion: Detecting masquerades**, Tech. Rep. 95, NISS, 1999
- *N. Ye, T. Ehiabor, and Y. Zhanget* , **First-order versus high-order stochastic models for computer intrusion detection**, Quality and Reliability Engineering International, vol. 18, 2002

References

- *A. Raftery* , **A model for high-order markov chains**, Journal of the Royal Statistical Society, series B, vol. 47, 1985
- *A. Raftery and S. Tavare* , **Estimation and modelling repeated patterns in high-order markov chains with the mixture transition distribution (MTD) model**, Journal of the Royal Statistical Society, series C - Applied Statistics, vol. 43, 1994
- *Y. Vardi and D. Lee* , **From image deblurring to optimal investments: Maximum likelihood solutions for positive linear inverse problem**, Journal of the Royal Statistical Society, series B, vol. 55, 1993
- *C. Callegari, S. Vaton, and M. Pagano* , **A new statistical approach to network anomaly detection**, Performance Evaluation of Computer and Telecommunication Systems (SPECTS), 2008

Outline

Theoretical Background

Entropy

The entropy H of a discrete random variable X is a measure of the amount of uncertainty associated with the value of X
Referring to an alphabet composed of n distinct symbols, respectively associated to a probability p_i, then

$$H = -\sum_{i=1}^{n} p_i \cdot log_2 p_i \text{ bit/symbol}$$

The starting point

The entropy represents a lower bound to the compression rate that we can obtain: the more redundant the data are and the better we can compress them.

Compression Algorithms

- **Dictionary based algorithms**: based on the use of a dictionary, which can be static or dynamic, and they code each symbol or group of symbols with an element of the dictionary
  - Lempel-Ziv-Welch (LZW)
- **Model based algorithms**: each symbol or group of symbols is encoded with a variable length code, according to some probability distribution.
  - Huffman Coding (HC)
  - Dynamic Markov Compression (DMC)

Lempel-Ziv-Welch

- Created by Abraham Lempel, Jacob Ziv, and Terry Welch. It was published by Welch in 1984 as an improved implementation of the LZ78 algorithm, published by Lempel and Ziv in 1978
- Universal adaptative[1] lossless data compression algorithm
- Builds a translation table (also called dictionary) from the text being compressed
- The string translation table maps the message strings to fixed-length codes

[1] The coding scheme used for the k^{th} character of a message is based on the characteristics of the preceding $k-1$ characters in the message

Huffman Coding

- Developed by Huffman (1952)
- Based on the use of a variable-length code table for encoding each source symbol
- The variable-length code table is derived from a binary tree built from the estimated probability of occurrence for each possible value of the source symbols
- Prefix-free code[2] that expresses the most common characters using shorter strings of bits than are used for less common source symbols

[2] The bit string representing some particular symbol is never a prefix of the bit string representing any other symbol

Dynamic Markov Compression

- Developed by Gordon Cormack and Nigel Horspool (1987)
- Adaptative lossless data compression algorithm
- Based on the modelization of the binary source to be encoded by means of a Markov chain, which describes the transition probabilities between the symbol “0” and the symbol “1”
- The built model is used to predict the future bit of a message. The predicted bit is then coded using arithmetic coding

System Design

Input

- The system input is given by raw traffic traces in libpcap format
- The 5-tuple is used to identify a connection, while the value of the TCP flags is used to build the "profile"
- A value s_i is associated to each packet:

$$s_i = SYN + 2 \cdot ACK + 4 \cdot PSH + 8 \cdot RST + 16 \cdot URG + 32 \cdot FIN$$

thus each "mono-directional" connection is represented by a sequence of symbols s_i, which are integers in $\{0, 1, \cdots, 63\}$

System Design

Training Phase

- Choose one of the three previously described algorithms (Huffman, DMC, or LZW)
- The compression algorithms have been modified so as that the “learning phase” is stopped after the training phase:
  - Huffman case: the occurency frequency of each symbol is estimated only on the training dataset
  - DMC case: the estimation of the Markov chain is only updated during the training phase
  - LZW case: the construction of the dictionary is stopped after the training phase
- Detection performed with a compression scheme that is “optimal” for the “normal” traffic used for building the considered “profile” and suboptimal for “anomalous” traffic

System Design

Detection Phase

- Append each distinct "observed" connection b, to the training sequence A
- Compute the "compression rate per symbol":

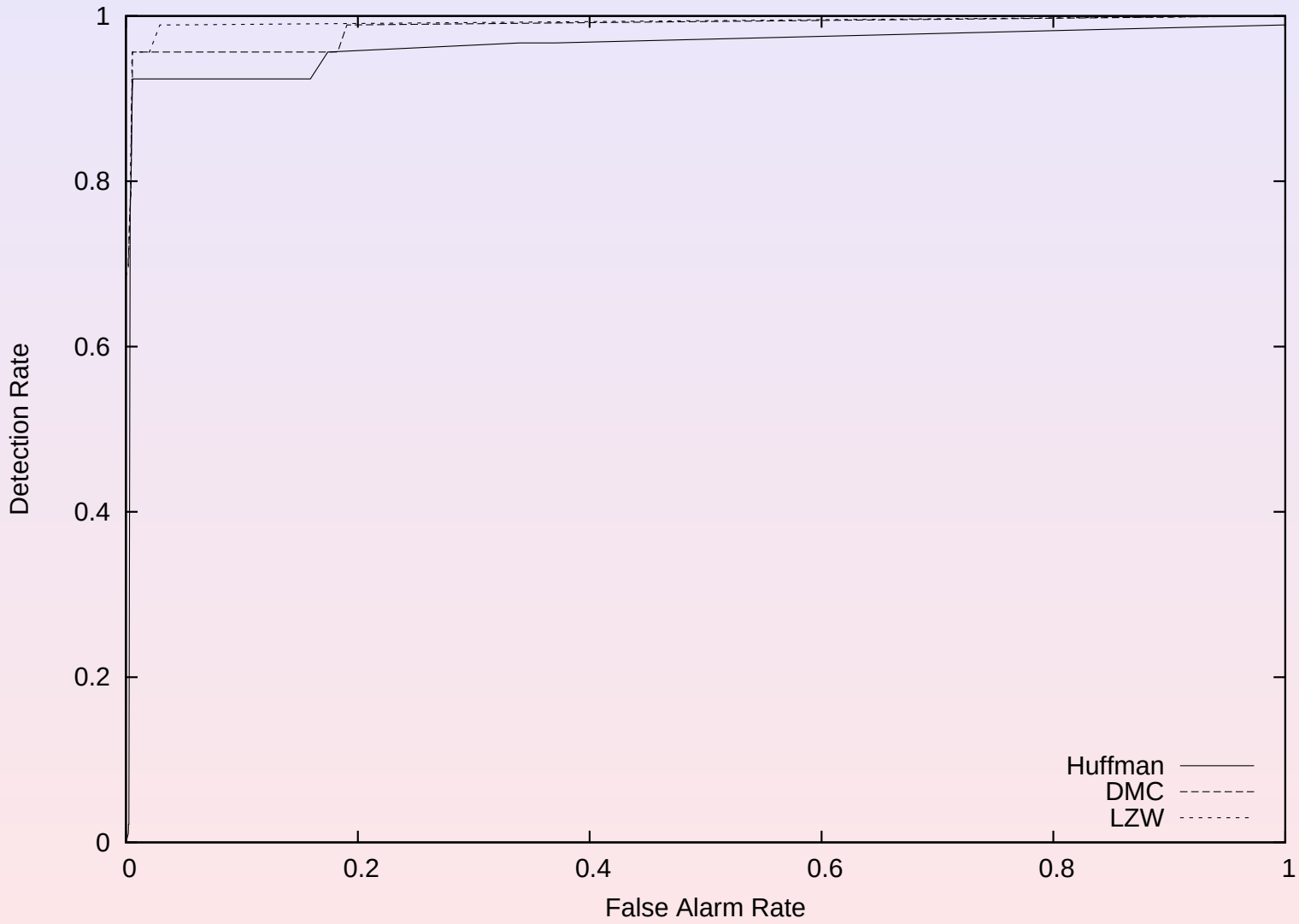
$$X = \frac{dim([A|b]^*) - dim([A]^*)}{Length(b)}$$

where $[X]^*$ represents the compressed version of X

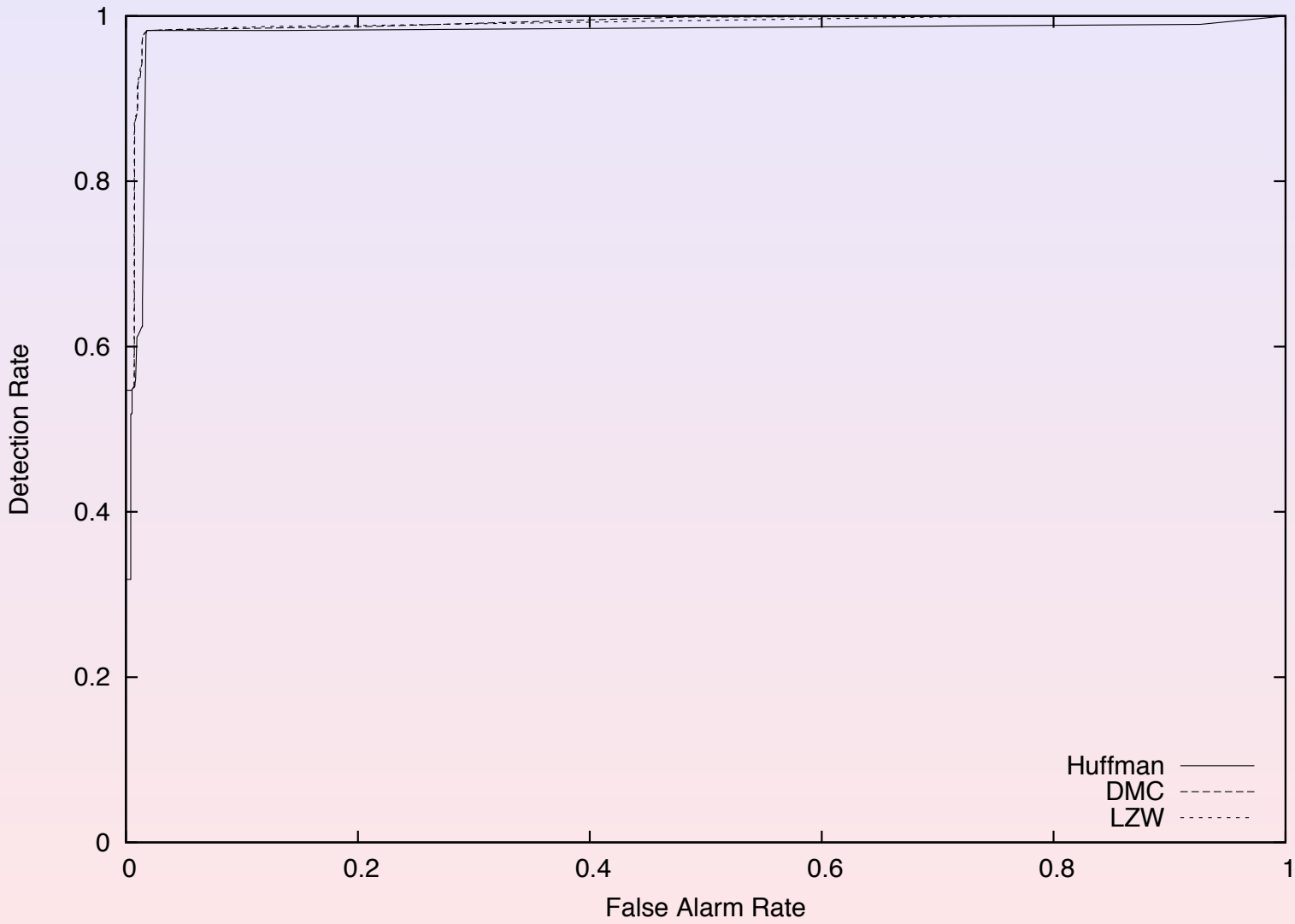
- Choose between a single hypothesis H_0 (normal traffic), and the composite hypothesis H_1 (anomaly)

$$X \underset{H_1}{\overset{H_0}{\lessgtr}} \xi$$

Results - System Comparison



Results - On-line System



References

- *T. Cover and J. Thomas* , **Elements of information theory**, Wiley-Interscience, 2nd ed., 2006
- *C. E. Shannon* , **A mathematical theory of communication**, Bell System Technical Journal, vol. 27 1948
- *D. Huffman* , **A method for the construction of minimum-redundancy codes**, Proceedings of the Institute of Radio Engineers, vol. 40, 1952
- *G. Cormack and N. Horspool* , **Data compression using dynamic Markov modelling**, vol. 30, 1987

References

- *J. Ziv and A. Lempel* , **Compression of individual sequences via variable-rate coding**, IEEE Transactions on Information Theory, vol. 24, 1978
- *T. Welch* , **A technique for high-performance data compression**, IEEE Computer Magazine, vol. 17, no. 6, 1984
- *Christian Callegari, Stefano Giordano, Michele Pagano* , **On the Use of Compression Algorithms for Network Anomaly Detection**, IEEE International Conference on Communications (ICC 2009)

Outline

Data Stream Mining

Data Stream mining

The data stream mining is a set of techniques which permits to analyze a data flow, almost in real-time, without storing all the data

- Can be used to analyze, for example, the network traffic
- The aim can be the calculation of histograms, the individuation of the most common features, etc.
- Two constraints
  1. Almost real-time
  2. Without storing all the data

The Count-Min Sketch Algorithm

Proposed by Cormode and Muthukrishran, 2004

- Each element of the data flow is identified by
  - **id** $i_t \in \{1, 2, \ldots, N\}$, with N big
  - **label** $c_t \in \mathbb{R}$
- The data flow is the sequence $(i_t, c_t)_{t \in \mathbb{N}}$
- An example
  - The data flow is the network traffic
  - i_t is the source IP address of packet t
  - c_t is the length of packet t

The Count-Min Sketch Algorithm

- At each instant $\tau \in \mathbb{N}$ and for each id i, we define a count $a_i(\tau)$

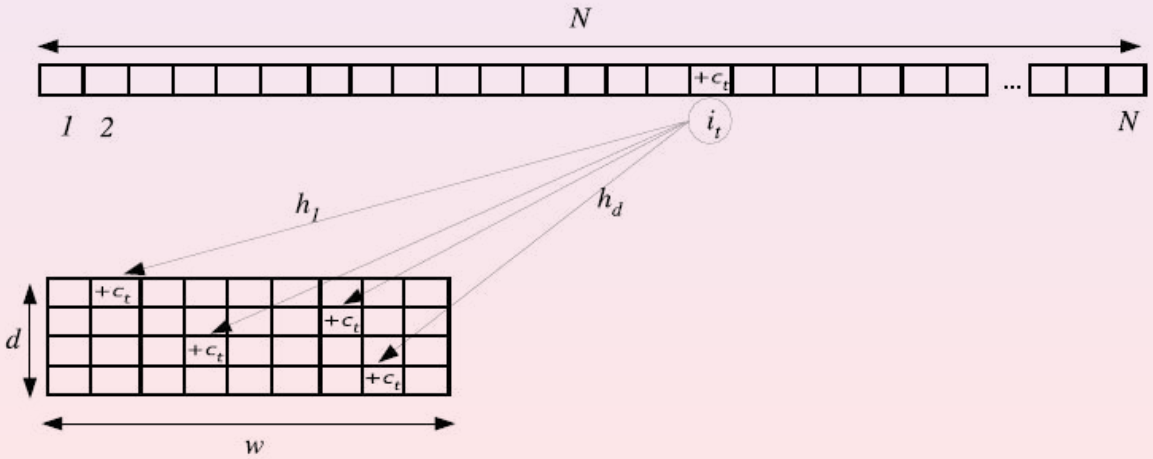
$$a_i(\tau) \stackrel{\text{def}}{=} \sum_{t=0}^{\tau} c_t \delta_{i,i_t}$$

with $\delta_{i,i_t} = 1$ if $i = i_t$ and $\delta_{i,i_t} = 0$ otherwise

- In our example $a_i(\tau)$ is the total number of bytes sent from i up to the instant τ
- First step of the algorithm: estimate $\hat{a}_i(\tau)$

The Count-Min Sketch Algorithm

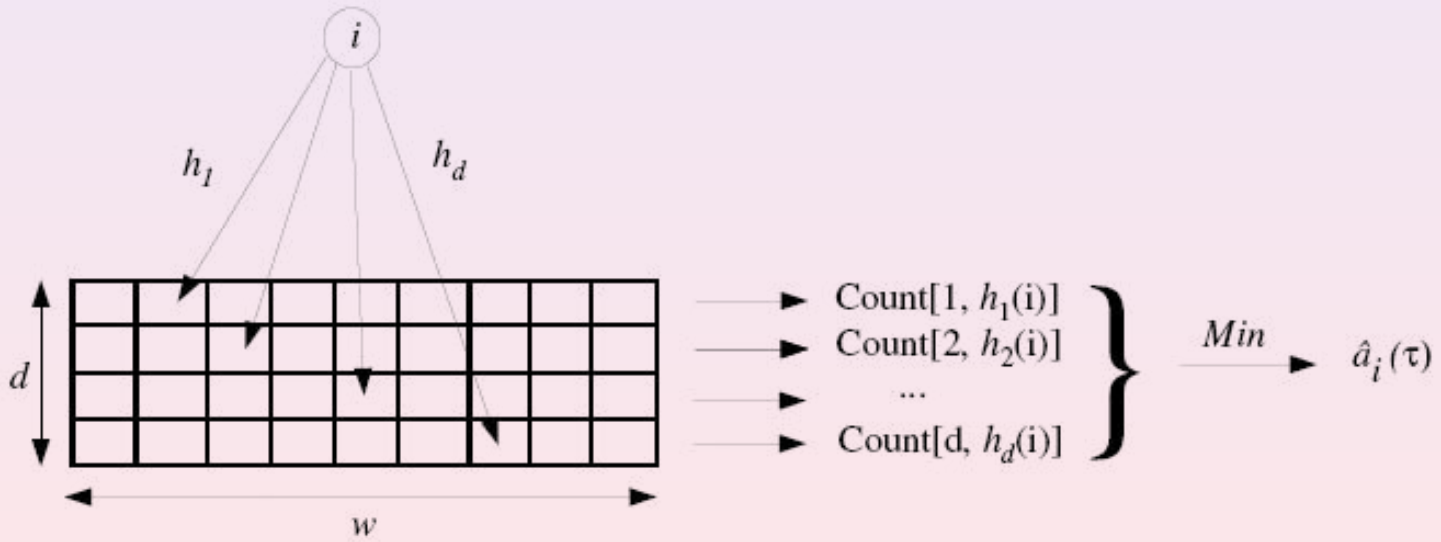
- Fix the precision ϵ and the failure probability δ
- Let's $w = \lceil \frac{e}{\epsilon} \rceil$ and $d = \lceil ln \frac{1}{\delta} \rceil$
- Let's take d independent random hash functions
- Let's allocate a table $Count_{dxw}$ initialized to zero
- For each instant t, let's update the table according to:
  for each
  $j \in \{1, 2, \ldots, d\}$ $Count[j, h_j(i_t)] = Count[j, h_j(i_t)] + c_t$



The Count-Min Sketch Algorithm

At the instant τ and for each id $i \in \{1, 2, \ldots, N\}$ we have an estimation $\hat{a}_i(\tau)$ of the count $a_i(\tau)$

$$\hat{a}_i(\tau) = \min_j Count[j, h_j(i_t)](\tau)$$



The Count-Min Sketch Algorithm

Main properties

- The estimator $\hat{a}_i(\tau) \geq a_i(\tau)$
- $P\{\hat{a}_i(\tau) \leq a_i(\tau) + \epsilon||\overrightarrow{a}(\tau)||_1\} \geq 1 - \delta$
  - $\overrightarrow{a}(\tau) \stackrel{\text{def}}{=} (a_1(\tau), a_2(\tau), \ldots, a_N(\tau))$
  - $||\overrightarrow{a}(\tau)||_1 \stackrel{\text{def}}{=} |a_1(\tau)| + |a_2(\tau)| + \ldots + |a_N(\tau)|$

The Count-Min Sketch Algorithm

Complexity in time

- Number of operations for updating the Count table is $O(ln(\frac{1}{\delta}))$
- Number of operations for calculating $\hat{a}_i$ is $O(ln(\frac{1}{\delta}))$

Complexity in space

- Number of words for storing the hash functions and the Count table is $O(\frac{1}{\epsilon} ln(\frac{1}{\delta}))$

Finding Heavy Hitters

- Aim: find those items whose frequencies exceed a threshold ϕ during the observation window
- These items are called **Heavy Hitters**
- Possible application: finding the IP addresses, whose contribution to the network traffic exceeds the threshold
- An anomaly can be detected as a variation of the heavy hitters distribution

At a given instant τ and for a given threshold ϕ we define heavy hitters all the i, such that $a_i(\tau) \geq \phi \parallel \vec{a}(\tau) \parallel_1$

Application of the Count-Min sketch algorithm

Initialization, at the instant $t = 0$

- Calculate $\| \vec{a}(0) \|_1 = c_0$
- Update *Count*
- Add i_0 and his estimated count $\hat{a}_{i_0}(0)$ to the list L of the potential heavy hitters

Application of the Count-Min sketch algorithm

Iteration, at the instant $t = t$

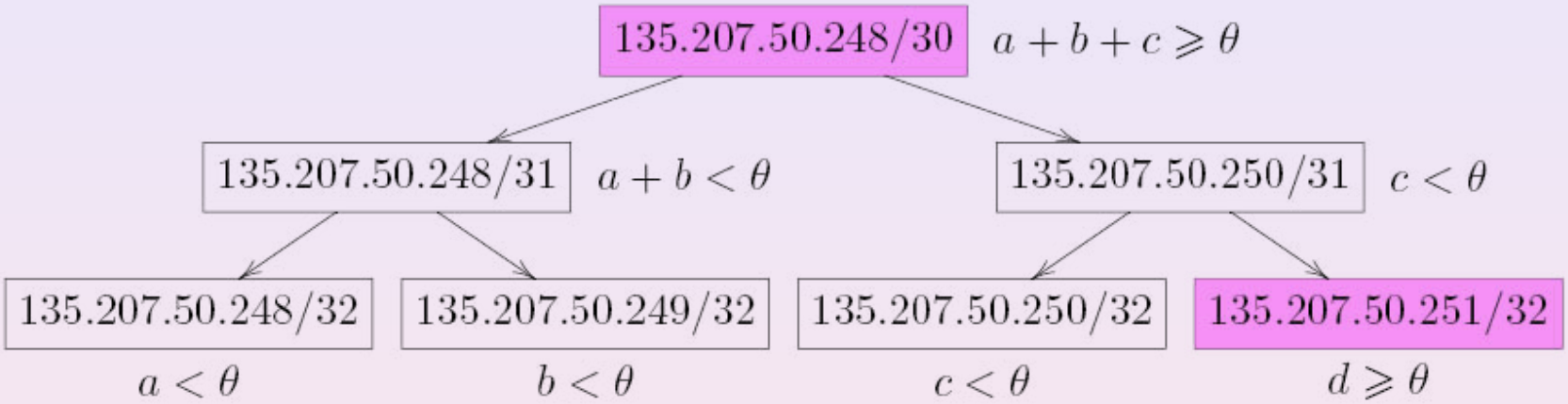
- Calculate $\| \vec{a}(t) \|_1 = \| \vec{a}(t-1) \|_1 + c_t$
- Update *Count*
- Calculate the estimated count $\hat{a}_{i_t}(t)$
- If $\hat{a}_{i_t}(t) \geq \phi \| \vec{a}(t) \|_1$ then
  - If i_t does not belongs to L: add i_t and his estimated count $\hat{a}_{i_t}(t)$ to L
  - Else replace the count corresponding to i_t
- Eliminate from the list every i, whose count is less than $\phi \| \vec{a}(t) \|_1$

After all the iterations, L contains all the heavy hitters
The real count of all the elements of L is greater than $(\phi - \epsilon) \| \vec{a}(t) \|_1$, with probability at least $1 - \delta$

Finding Hierarchical Heavy Hitters

- Extension of the method, which takes into account the hierarchical structure of the IP addresses
- The Hierarchical Heavy Hitters (HHH) are defined recursively from the bottom to the top of the hierarchy
- At the lowest level (level 0), the HHH are the Heavy Hitters (i.e all those source addresses whose counts exceed the threshold)
- At level $l > 0$ an IP prefix is a HHH if its count minus the count of its descendant HHHs is greater that or equal to the threshold

Finding Hierarchical Heavy Hitters



a	b	c	d	θ
10 MB	10 MB	10 MB	40 MB	25 MB

References

- *Graham Cormode, S. Muthukrishnan* , **An Improved Data Stream Summary: The Count-Min Sketch and its Applications**, Theoretical Informatics, 2004
- *Pascal Cheung-Mon-Chan et al* , **Finding Hierarchical Heavy Hitters with the Count Min Sketch**, IPS-MOME 2006

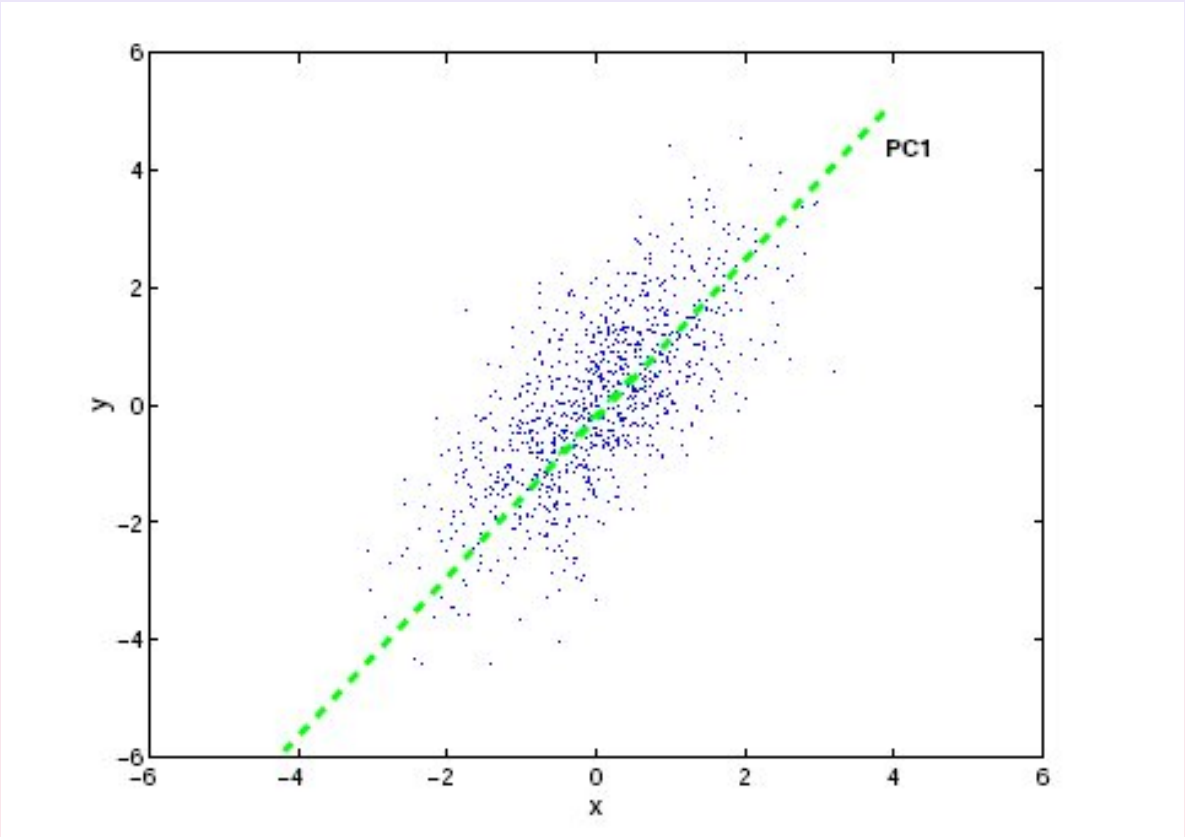
Outline

Principal Component Analysis

PCA is a coordinate transformation method that maps the measured data onto a new set of axes

- These axes are called Principal Components
- Each principal component points in the direction of maximum variation or energy remaining in the data
- The principal axes are ordered by the amount of energy in the data they capture

Geometric illustration



Data

A week of network-wide traffic measurements from Internet2:

- Internet2 samples 1 out of every 100 packets for inclusion in the flow statistics
- In Internet2 packets are aggregated into five-minute time-bins
- Abilene anonymizes the last eleven bits of the IP address stored in the flow records

Routing Info.

In order to aggregate the collected IP flows into OD flows, we also need to parse the routing data. Internet2 deploys Zebra BGP monitors that record all BGP messages they receive.

Data

Measurements Matrix, $X_{t \times p}$:

- Let p denote the number of traffic aggregate
- Let t denote the number of time-bin
- Column i denotes the time-series of the i-th traffic aggregate, with zero mean
- Row j represents an instance of all the traffic aggregate at j-th time-bin
- x, transposed row of X

$$\mathcal{X} = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1p} \\ x_{21} & x_{22} & \cdots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ x_{t1} & x_{t2} & \cdots & x_{tp} \end{pmatrix}$$

Principal Component Analysis

- Using PCA, we find that the set of OD flows has small intrinsic dimension (10 or less)
- Stability over time of this kind of representation (from week to week)

Origin Destination Flows

An OD flow consists of all traffic entering the network at a given point, and exiting the network at some other point, this is one out of the three traffic aggregations analyzed.

- High dimensional multivariate structure
- PCA: lower dimensional approximation

Principal Component Analysis

- Generalization of PCA to higher dimensions, as in the case of X , take the rows of X as points in Euclidean space, so that we have a dataset of t points in $\mathbb{R}^p$
- Mapping the data onto the first r principal axes places the data into an r-dimensional hyperplane.

Linear algebraic formulation

- Calculating the principal components is equivalent to solving the symmetric eigenvalue problem for the matrix $X^T X$
- Each principal component v_i is the i-th eigenvector computed from the spectral decomposition of $X^T X$:

$$X^T X v_i = \lambda_i v_i \quad i = 1, \dots p \tag{3}$$

Where λ_i is the eigenvalue corrisponding to v_i

- k-th principal component:

$$v_k = \arg \max_{||v||=1} ||(X - \sum_{i=1}^{k-1} X v_i v_i^T) v||$$

Principal Component Analysis

Once the data have been mapped into principal component space, it can be useful to examine the transformed data one dimension at a time:

- The contribution of principal axis i as a function of time is given by Xv_i
- This vector can be normalized to unit length by dividing by $\sigma_i = \sqrt{\lambda_i}$
- Thus, we have for each principal axis i:

$$u_i = \frac{Xv_i}{\sigma_i} \qquad i = 1, \ldots, p \tag{4}$$

The u_i (eigenflows) are vectors of size t and orthogonal by construction

Principal Component Analysis

- Thus vector u_i captures the temporal variation common to all flows along principal axis i
- The set of principal components $\{v_i\}_{i=1}^{p}$ can be arranged in order as columns of a *principal matrix* $V_{p\times p}$
- Likewise we can form the matrix $U_{t\times p}$ in which column i is u_i
- Then taken together, V , U, and σ_i can be arranged to write each traffic aggregate X_i as:

$$\frac{X_i}{\sigma_i} = U(V^T)_i \qquad i = 1, \ldots, p \tag{5}$$

X_i is the time-series of the i-th traffic aggregate and $(V^T)_i$ is the i-th row of V

Principal Component Analysis

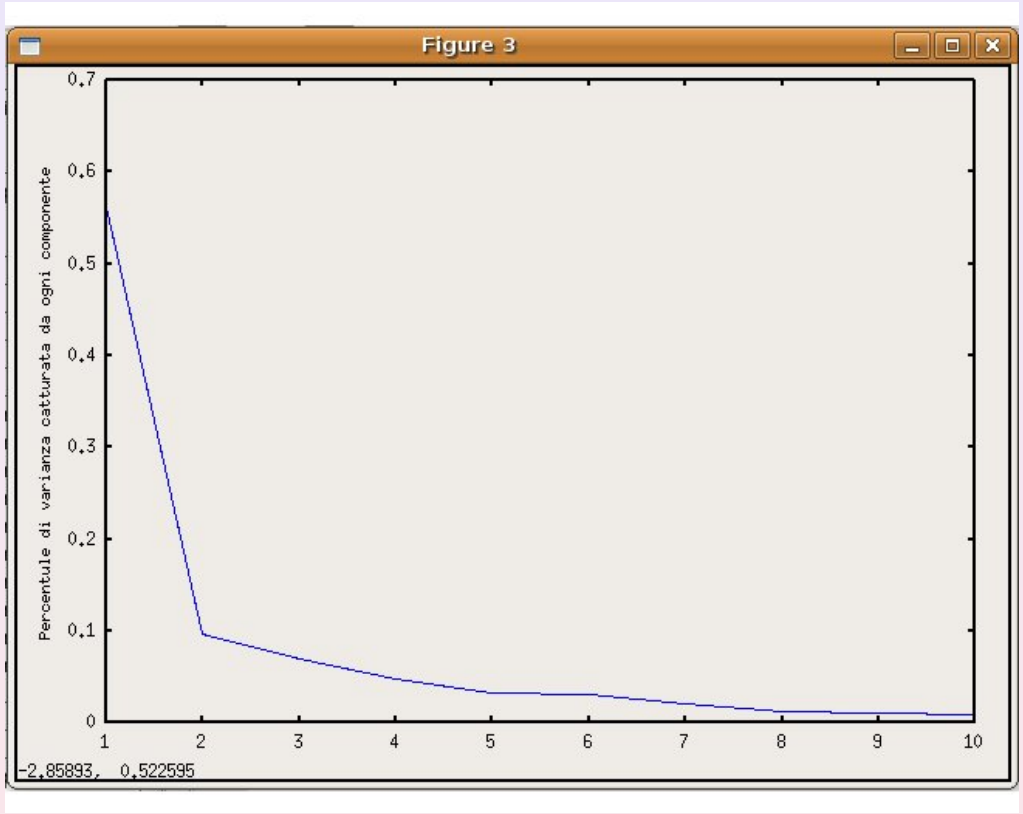
Equation 5 makes clear that each traffic aggregate X_i is in turn a linear combination of the u_i, with associated weights $(V^T)_i$

- The elements of $\{\sigma_i\}_{i=1}^{k}$ are the singular values

$$||Xv_i|| = v_i^T X^T X v_i = \lambda_i v_i^T v_i = \lambda_i \tag{6}$$

- Thus, the singular values are useful for gauging the potential for reduced dimensionality in the data, often simply through their visual examination in a scree plot

Scree Plot



Energy Percentages captured by first PCs

Lower Dimensional Approximation

- Finding that only r singular values are non-negligible, implies that X effectively resides on an r-dimensional subspace of $\mathbb{R}^p$:

$$X' \approx \sum_{i=1}^{r} \sigma_i u_i v_i \tag{7}$$

where $r < p$ is the effective intrinsic dimension of X

Subspace Method

Subspace Method

This method is based on a separation of the high-dimensional space occupied by a set of network traffic measurements into disjoint subspaces corresponding to normal and anomalous network conditions.

- This separation can be performed effectively by Principal Component Analysis
- Once the principal axes have been determined, the data-set can be mapped onto the new axes
  1. Normal subspace: S
  2. Anomalous Subspace: $\tilde{S}$

Modeled and Residual Part of x

Detecting volume anomalies in link traffic relies on the separation of link traffic x at any time-step into *normal* and *anomalous* components:

1. *modeled* part of x
2. *residual* part of x

We seek to decompose the set of link measurements at a given point in time x:

$$x = \hat{x} + \tilde{x}$$

We form $\hat{x}$ by projecting x onto S, and we form $\tilde{x}$ by projecting x onto $\tilde{S}$

Anomalous Subspace

Anomalous Subspace, $\tilde{x}$

To accomplish this, we arrange the set of principal components corresponding to the normal subspace $(v_1, v_2, \ldots, v_r)$ as columns of a matrix P of size $m \times r$ where r denotes the number of normal axes.

$$\hat{x} = PP^T x = Cx \qquad \text{and} \qquad \tilde{x} = (I - PP^T)x = \tilde{C}x$$

Squared Prediction Error

A useful statistic for detecting abnormal changes in $\tilde{x}$ is the Squared Prediction Error (SPE):

$$SPE = ||\tilde{x}||^2 = ||\tilde{C}x||^2$$

and we may consider network traffic to be normal if:

$$SPE \leq threshold$$

Identification

- Now we know the anomalous time bin
- We don't know which is the anomalous traffic aggregate responsible for this anomaly

Identification, for every $||\tilde{x}||^2$ over the threshold:

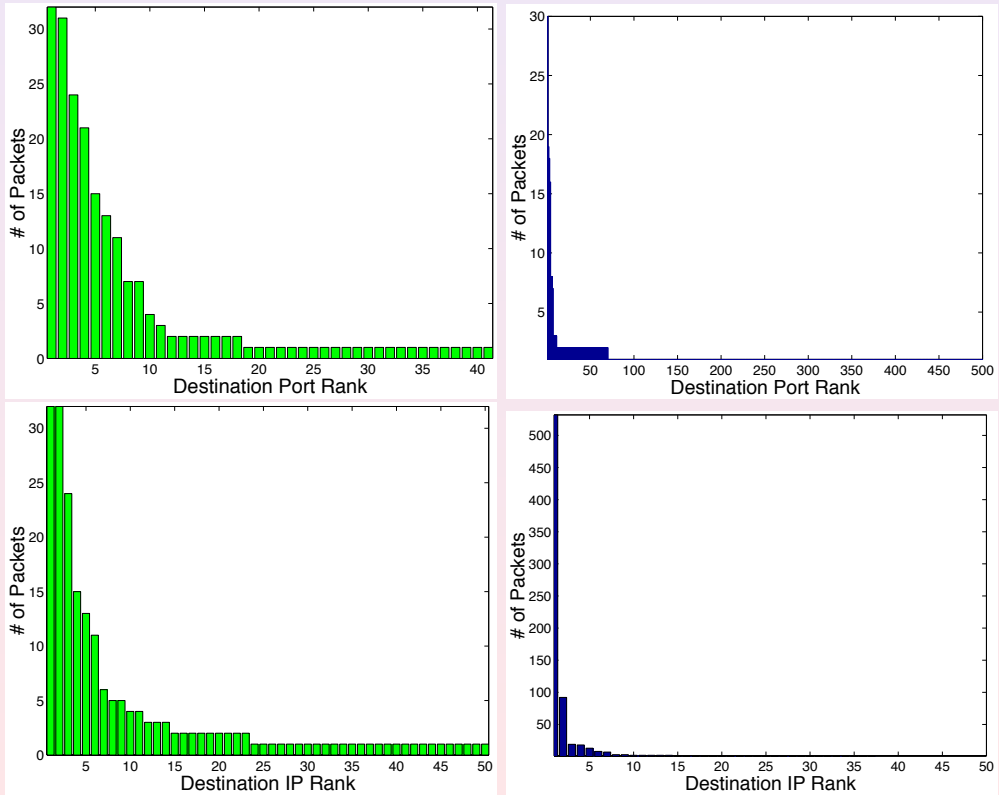
We determine the smallest set of OD flows, which if removed from the corresponding statistic, would bring it under threshold.

Multiway Subspace Method

- The distributions of packet features (IP addresses and ports) observed in flow traces reveal both the presence and the structure of a wide range of anomalies.
- They enable highly sensitive detection of a wide range of anomalies
- Traffic Features:
  1. Src IP
  2. Dest IP
  3. Src Port
  4. Dest Port

Features Distributions

- Many important kinds of traffic anomalies cause changes in the distribution of addresses or ports observed in traffic
- How feature distributions change as the result of a traffic anomaly (e.g., port scan)



(a) Normal (b) During Anomaly

Entropy

The distribution of traffic features is a high-dimensional object and so can be difficult to work with directly.

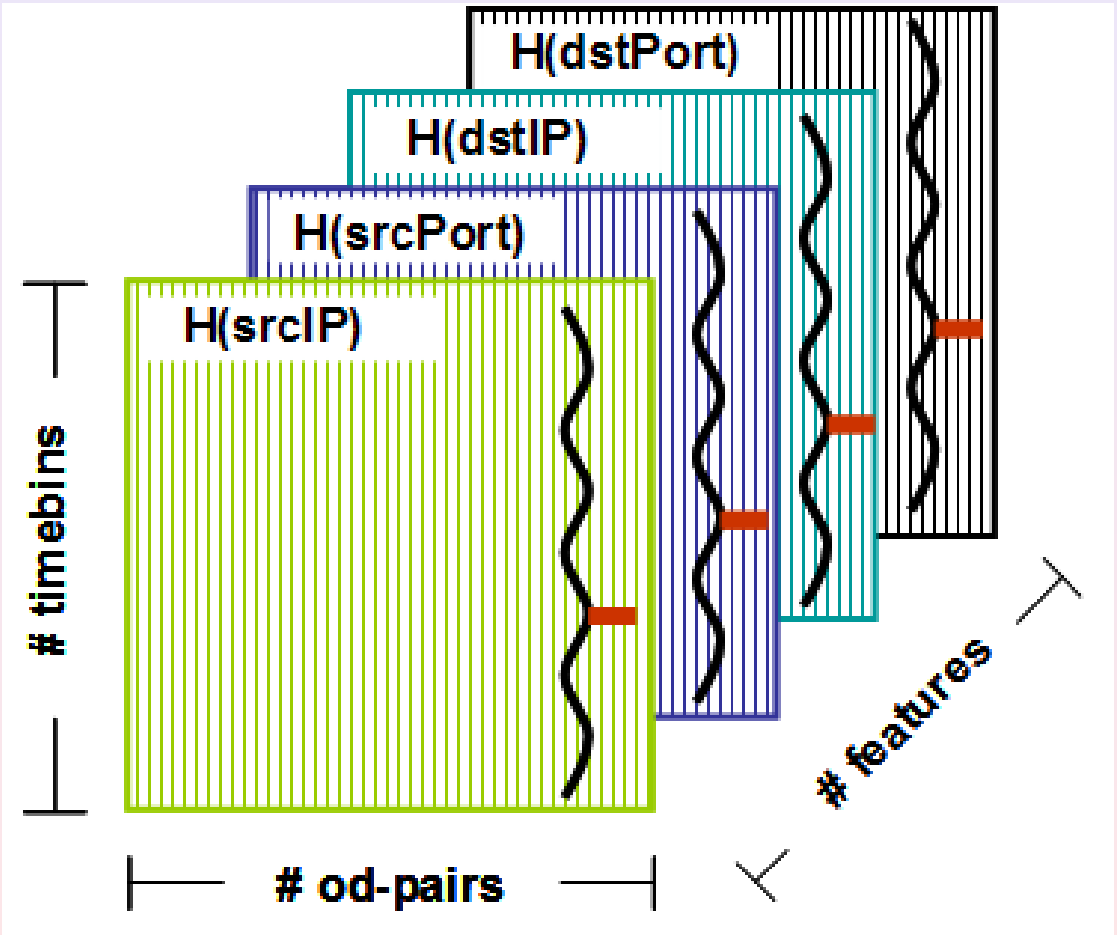
1. Analyze the degree of dispersal or concentration of the distribution
2. A metric that captures the degree of dispersal or concentration of a distribution is sample entropy
3. Empirical histogram $Y = \{n_i, i = 1, \ldots, N\}$
4. Sample entropy:

$$H(Y) = -\sum_{i=1}^{N} \frac{n_i}{S} \log_2 \frac{n_i}{S}$$

where $S = \sum_{i=1}^{N} n_i$ is the total number of observations in the histogram

Multiway Anomalies

Anomalies typically induce changes in multiple traffic features.

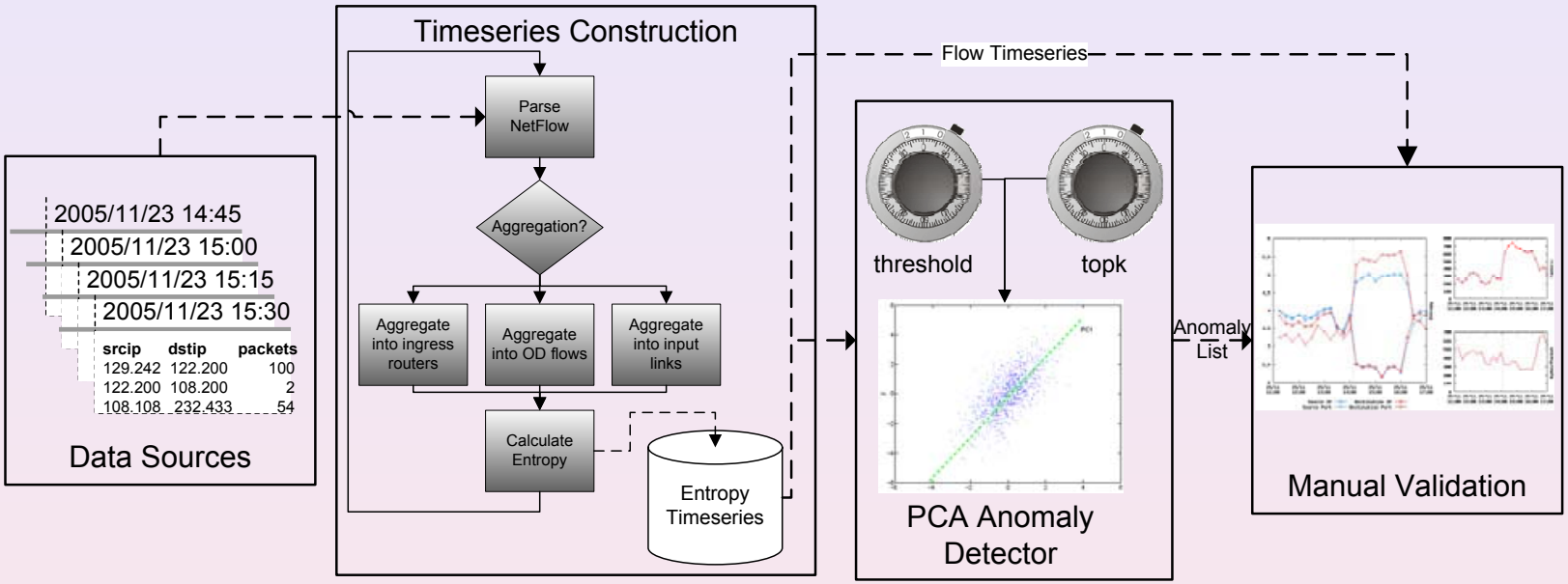


Multiway Subspace Method

Anomalies Spanning multiple traffic features

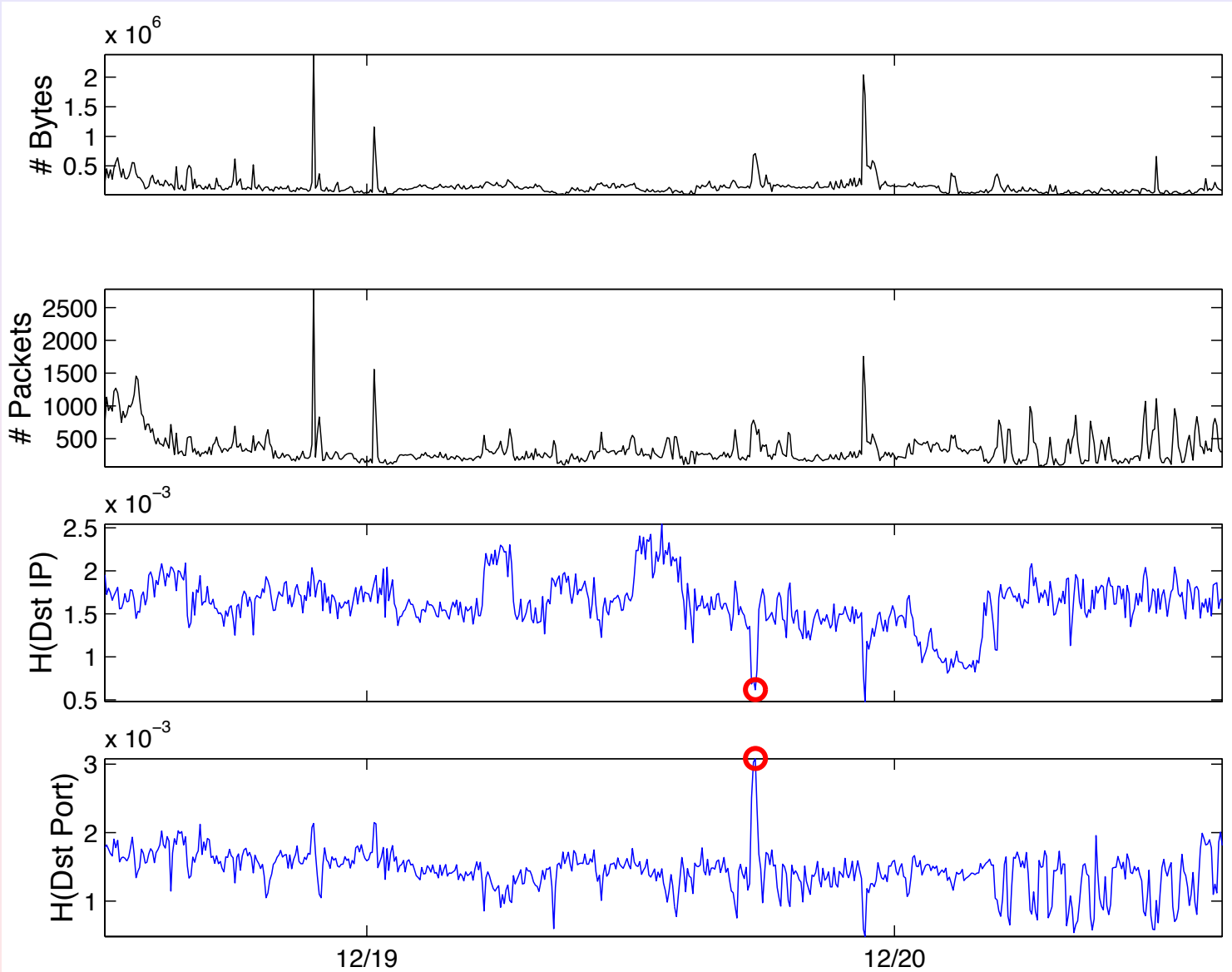
- Unfold the multiway matrix in Figure into a single, large matrix
- With this technique subspace method can detect anomalies spanning multiple traffic features

The Architecture

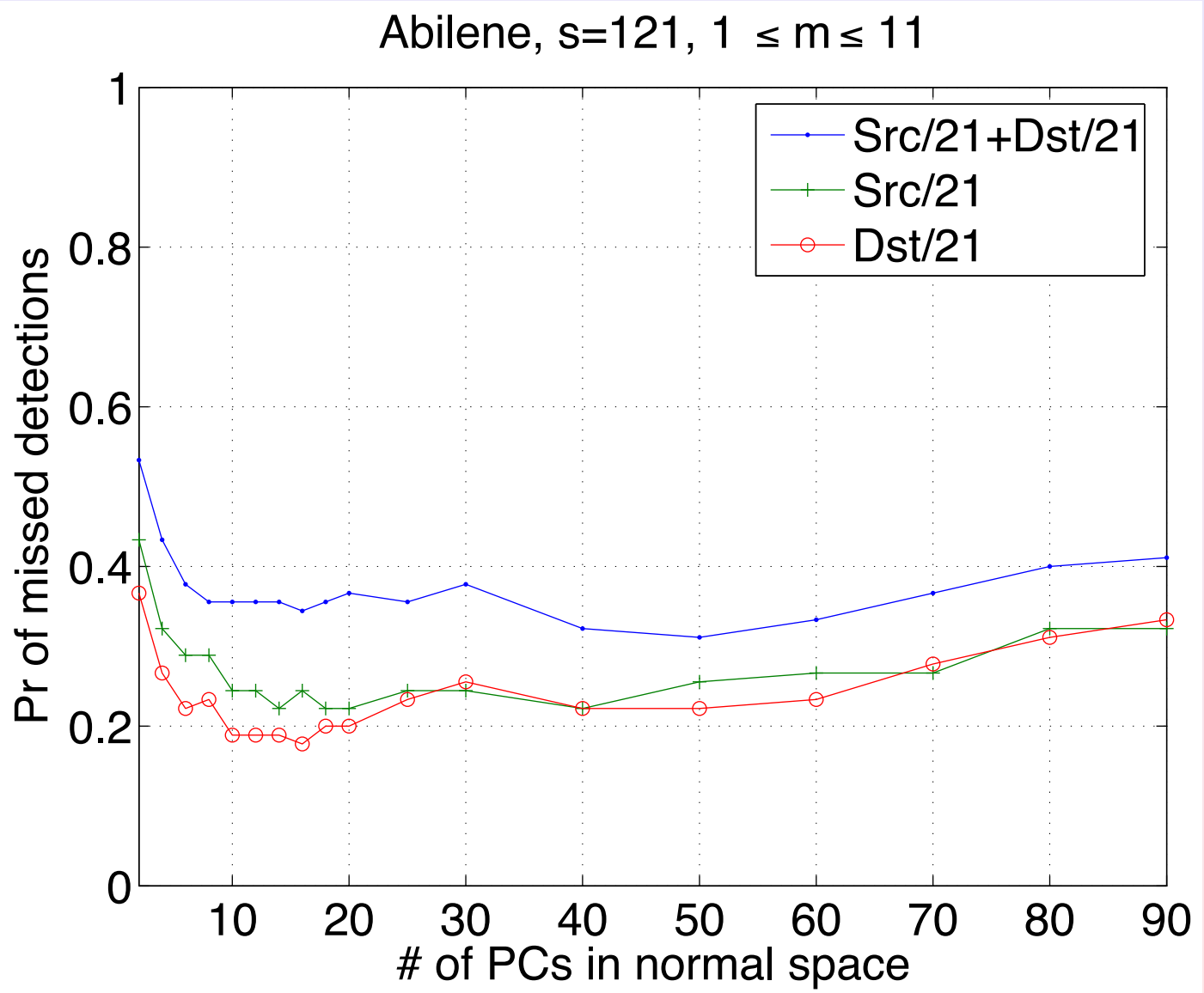


This method has been combined with Sketch

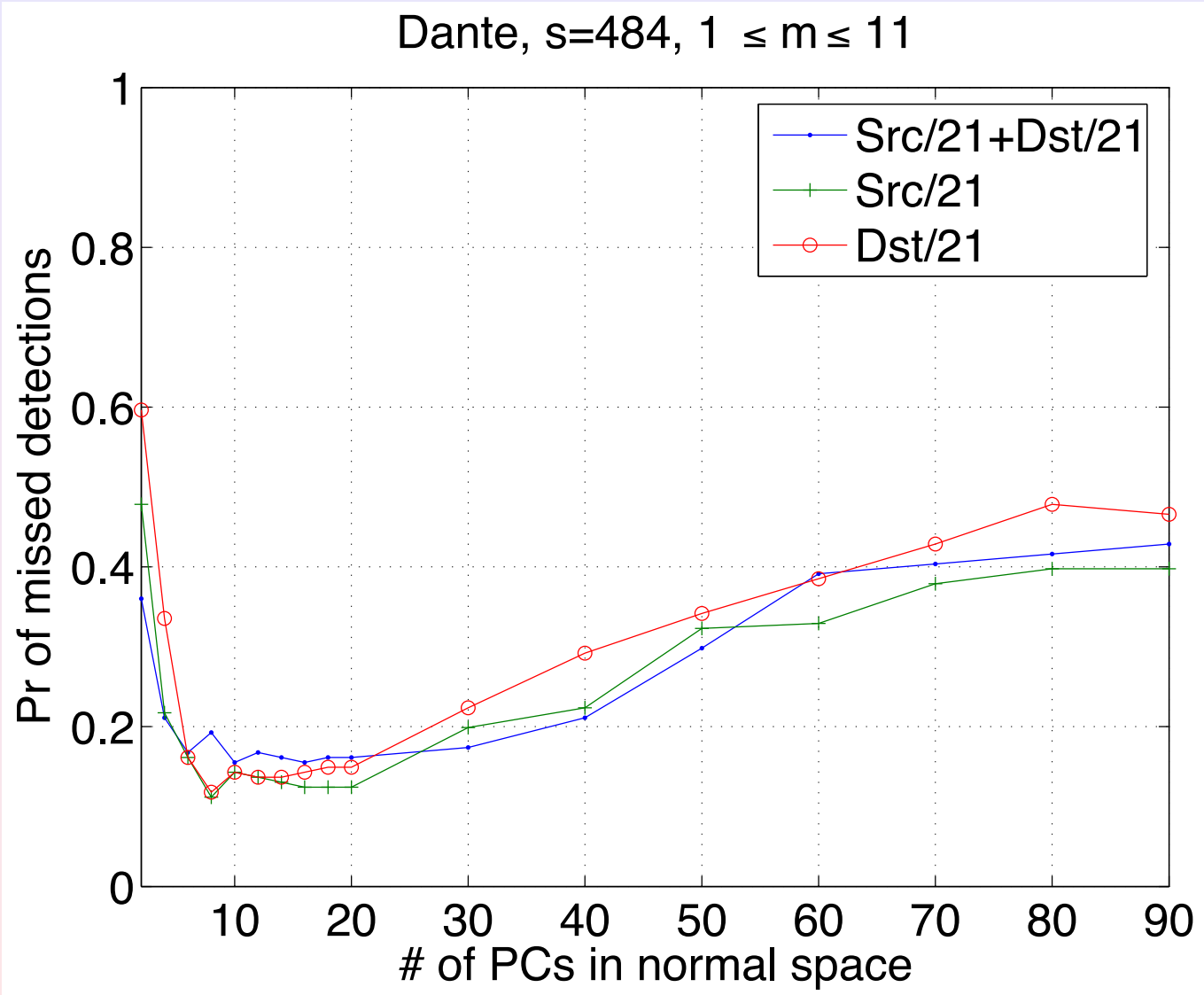
Experimental Results



Experimental Results



Experimental Results

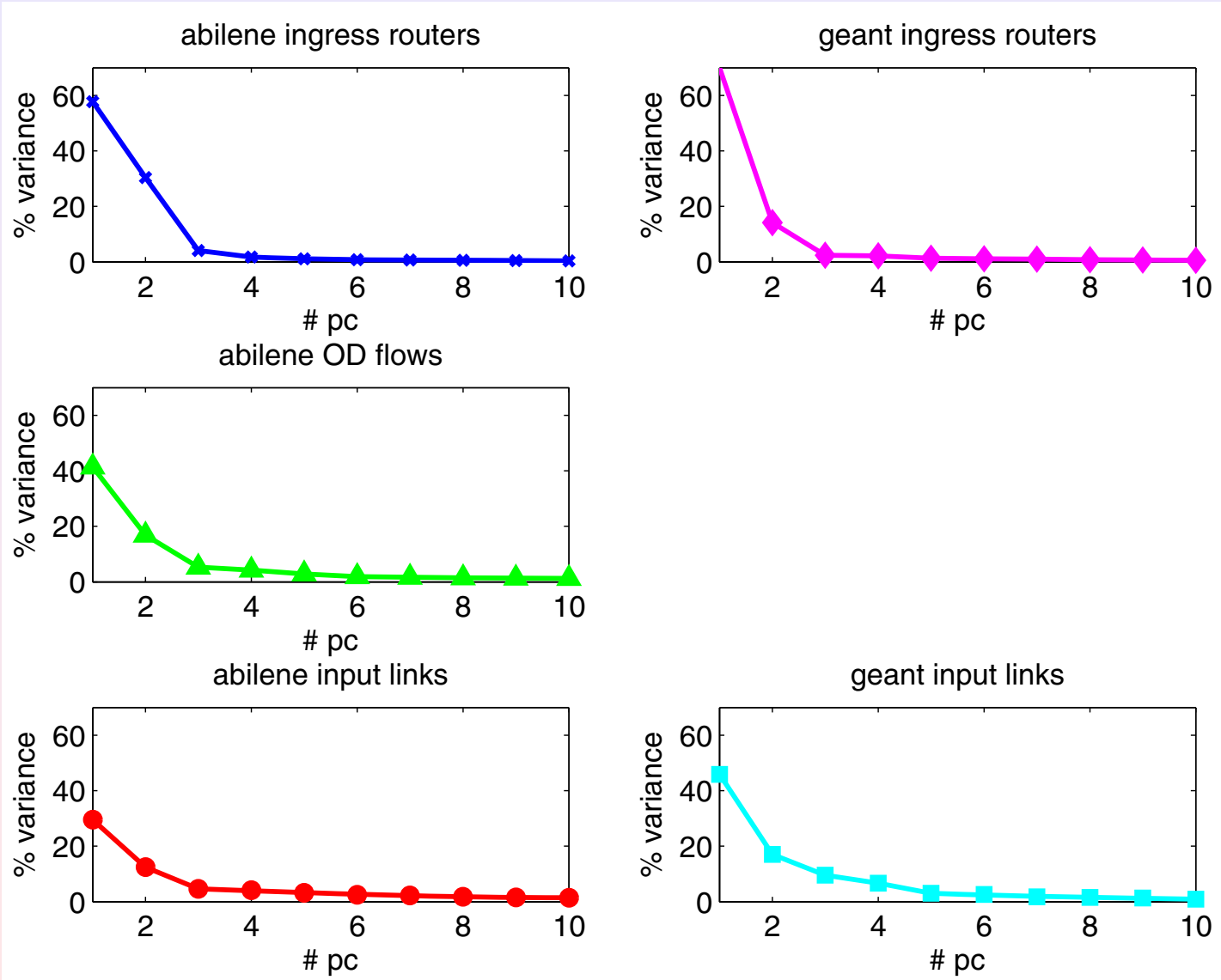


Experimental Results

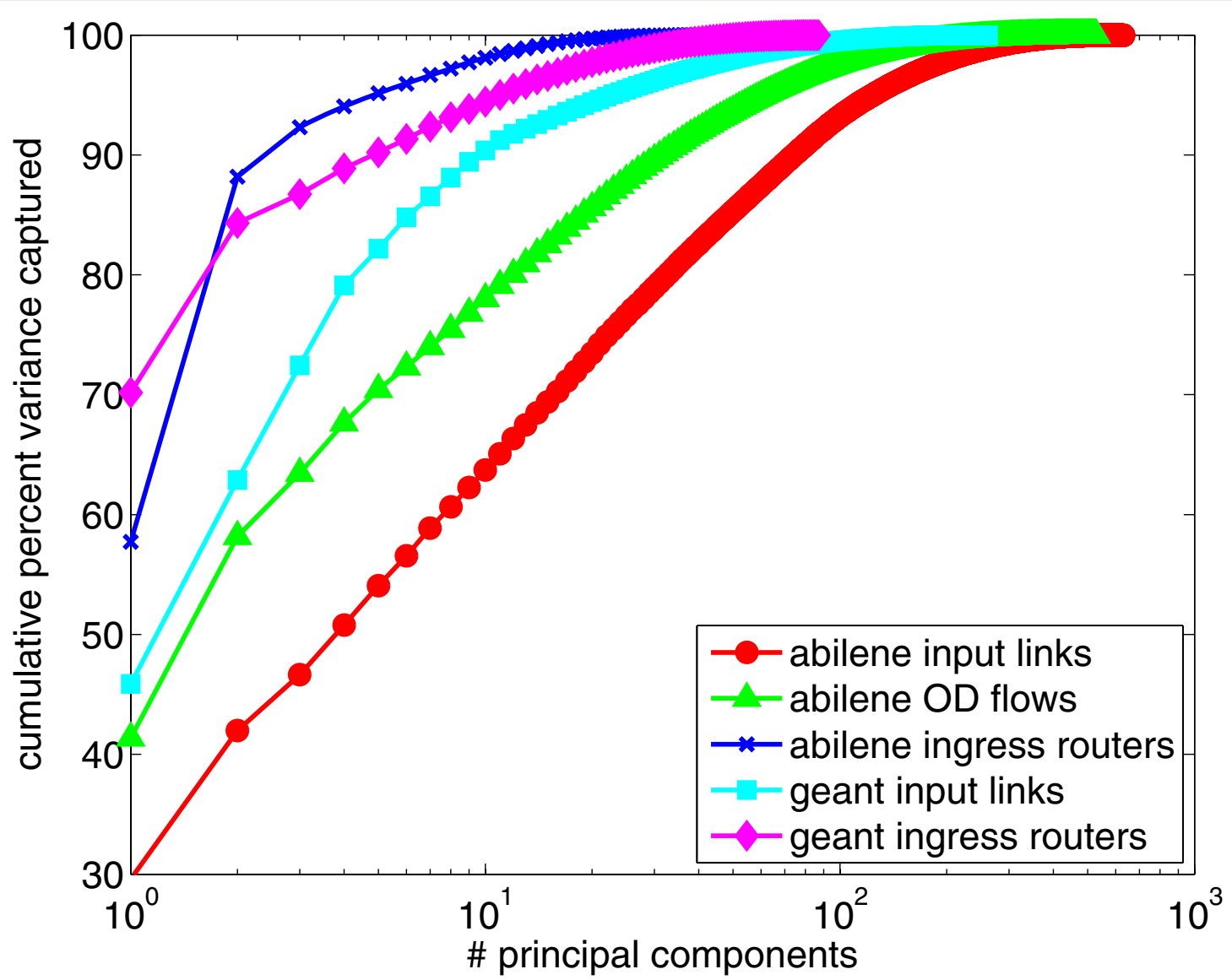
Some remarks on the method:

- The false-positive rate is very sensitive to the dimensionality of the normal subspace
- The effectiveness of PCA is sensitive to the way the traffic measurements are aggregated
- Large anomalies can contaminate the normal sub-space
- Pinpointing the anomalous flows is inherently difficult

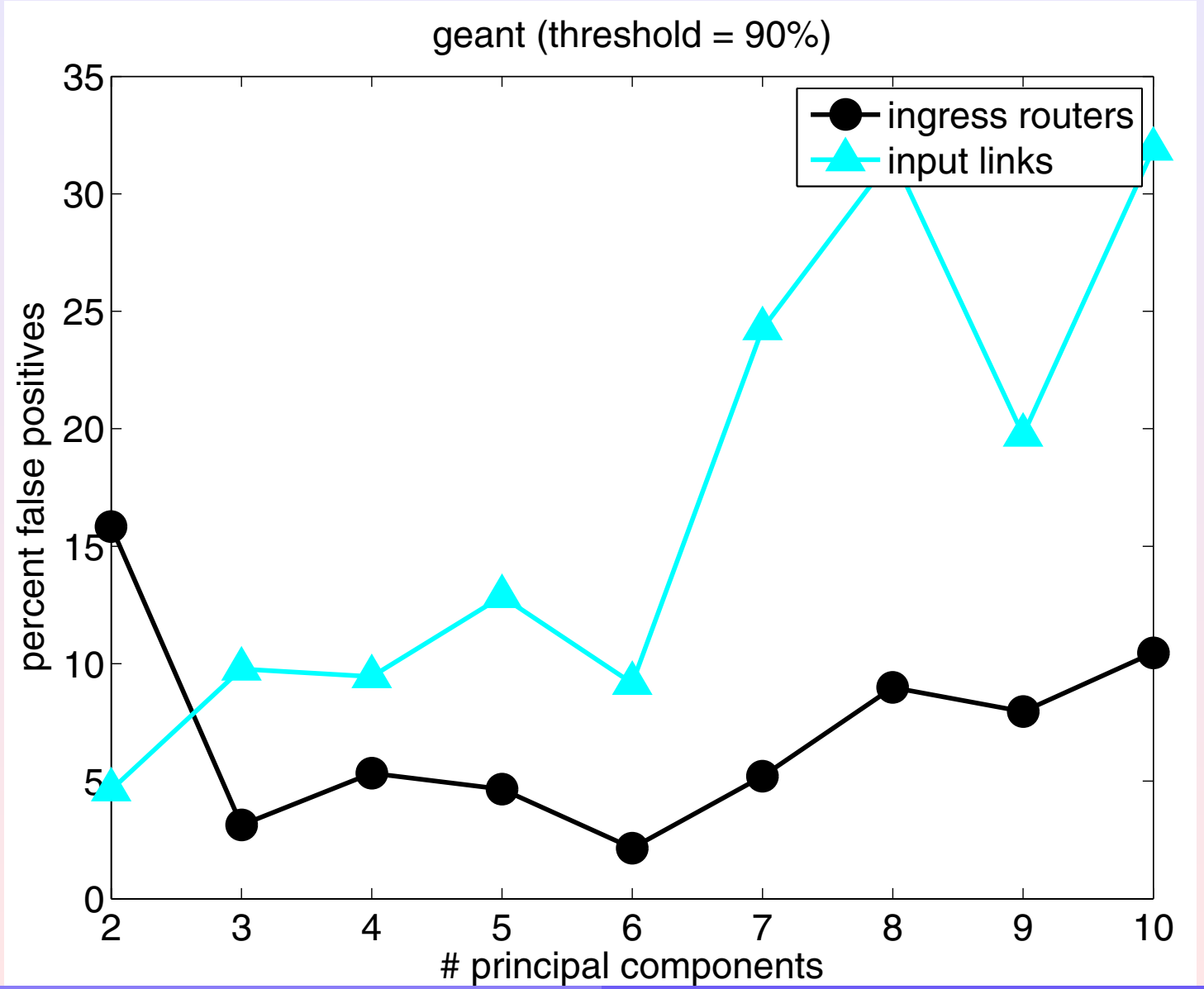
Experimental Results



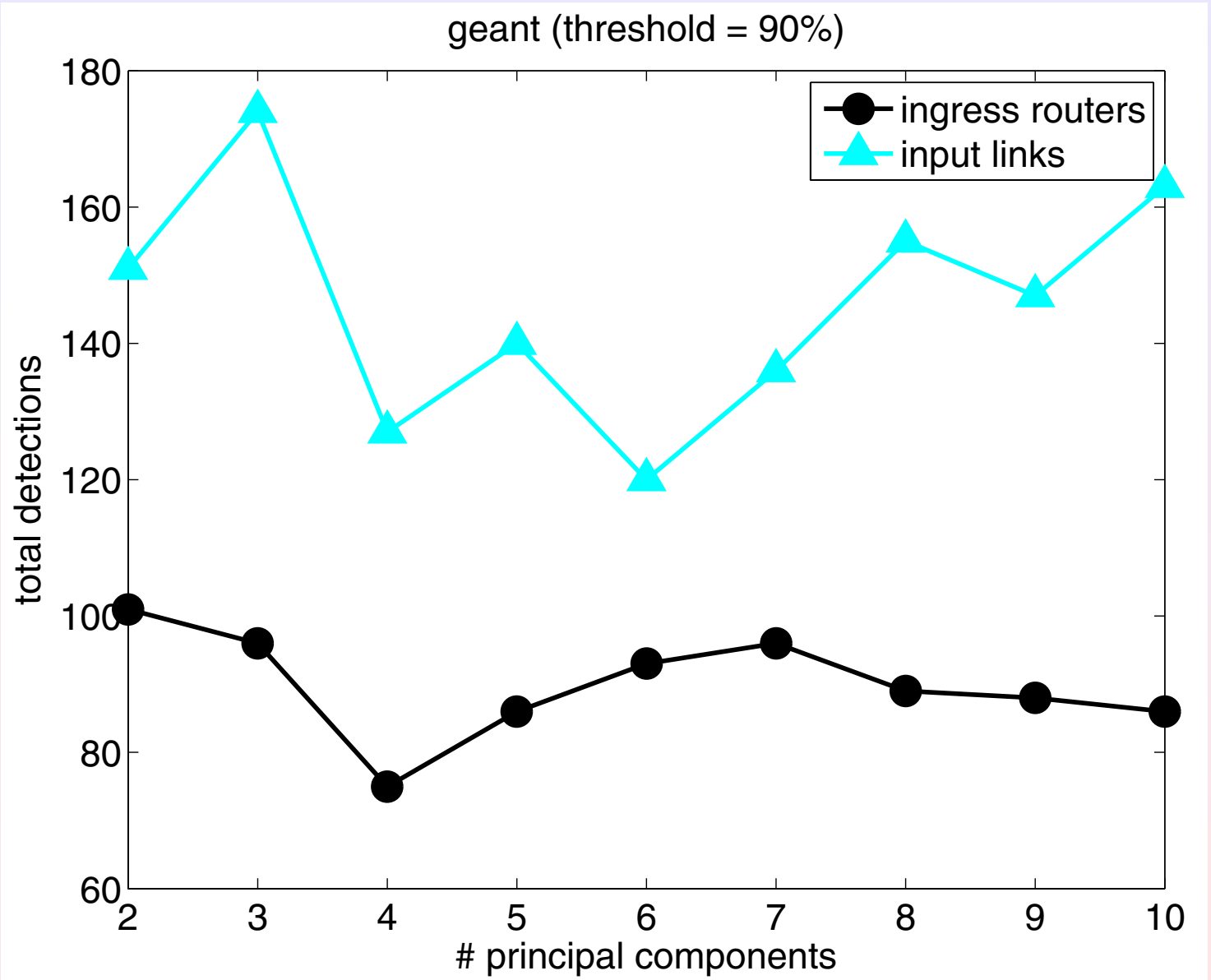
Experimental Results



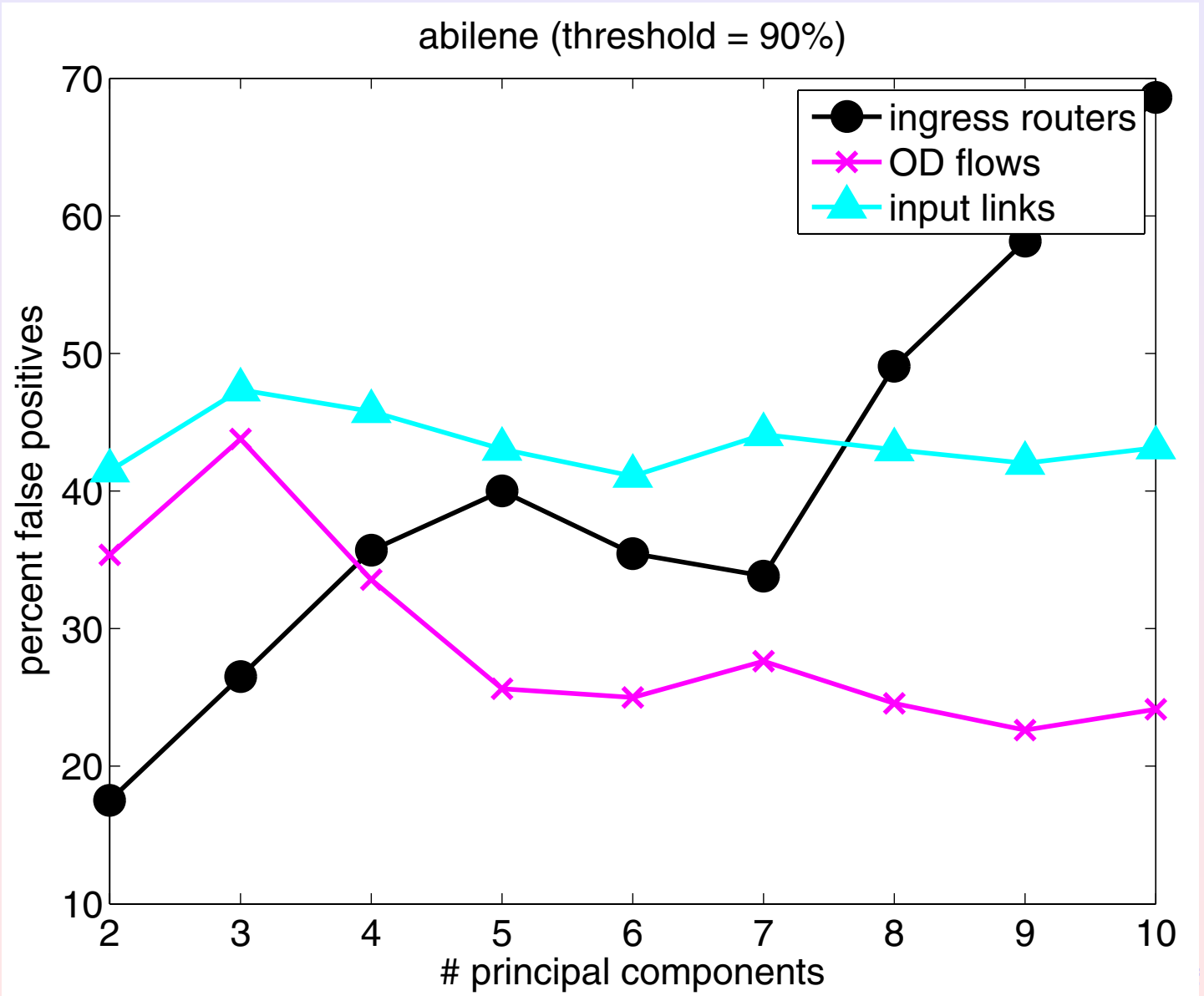
Experimental Results



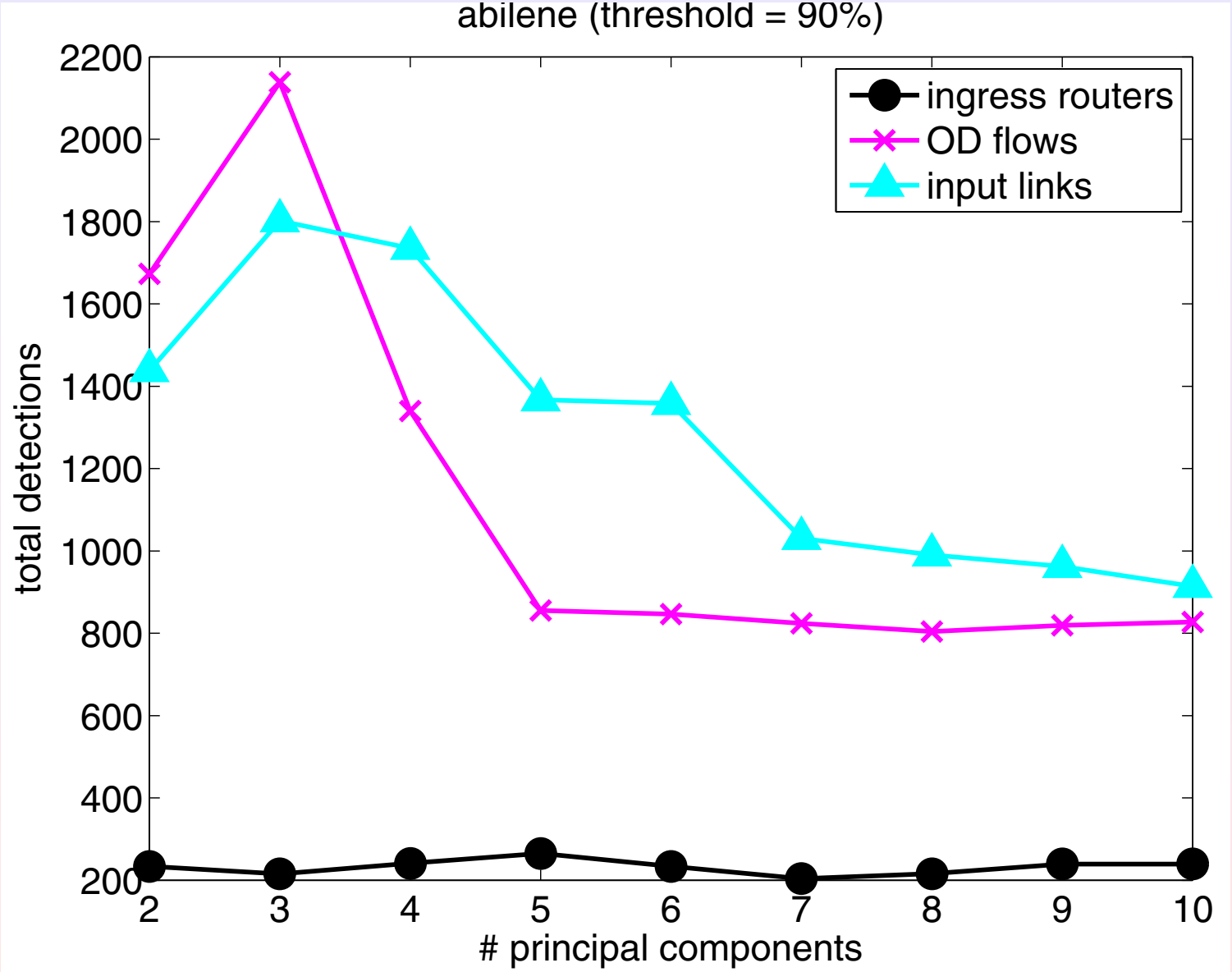
Experimental Results



Experimental Results



Experimental Results



References

- *Papagiannaki K. Crovella M. Diot C. Kolaczyk E. D. Lakhina, A. and N. Taft* , **Structural analysis of network traffic flows**, Tech rep 2004
- *Crovella M. Lakhina, A. and C. Diot* , **Characterization of network-wide anomalies in traffic flows**, ACM SIGCOMM conference on Internet measurement, 2004
- *Crovella M. Lakhina, A. and C. Diot* , **Diagnosing network-wide traffic anomalies**, Conference on Applications, technologies, architectures, and protocols for computer communications, 2004
- *Crovella M. Lakhina, A. and C. Diot* , **Mining anomalies using traffic feature distributions**, ACM SIGCOMM Computer Communication Review, 2005
- *Jennifer Rexford Christophe Diot Haakon Ringberg, Augustin Soule* , **Sensitivity of PCA for Traffic Anomaly Detection**, ACM SIGMETRICS, 2007

Outline

Wavlet Analysis

- The wavelets are scaled and translated copies (known as “daughter wavelets”) of a finite-length or fast-decaying oscillating waveform (known as the “mother wavelet”)
- Wavelet transforms have advantages over traditional Fourier transforms for representing functions that have discontinuities and sharp peaks
- The main difference, with respect to the Fourier transform, is that wavelets are localized in both time and frequency whereas the standard Fourier transform is only localized in frequency

Wavelet Decomposition

- Mother wavelet $\psi(t)$, satisfying the admissibility condition

$$\int \frac{|\Psi(\omega)|^2}{|\omega|} d\omega < \infty$$

- Wavelet basis

$$\{\psi_{m,n}(t)\}_{m,n\in\mathbb{Z}} = \left\{a_0^{-m/2}\psi\left(a_0^{-m}t - nb_0\right)\right\}_{m,n\in\mathbb{Z}}$$

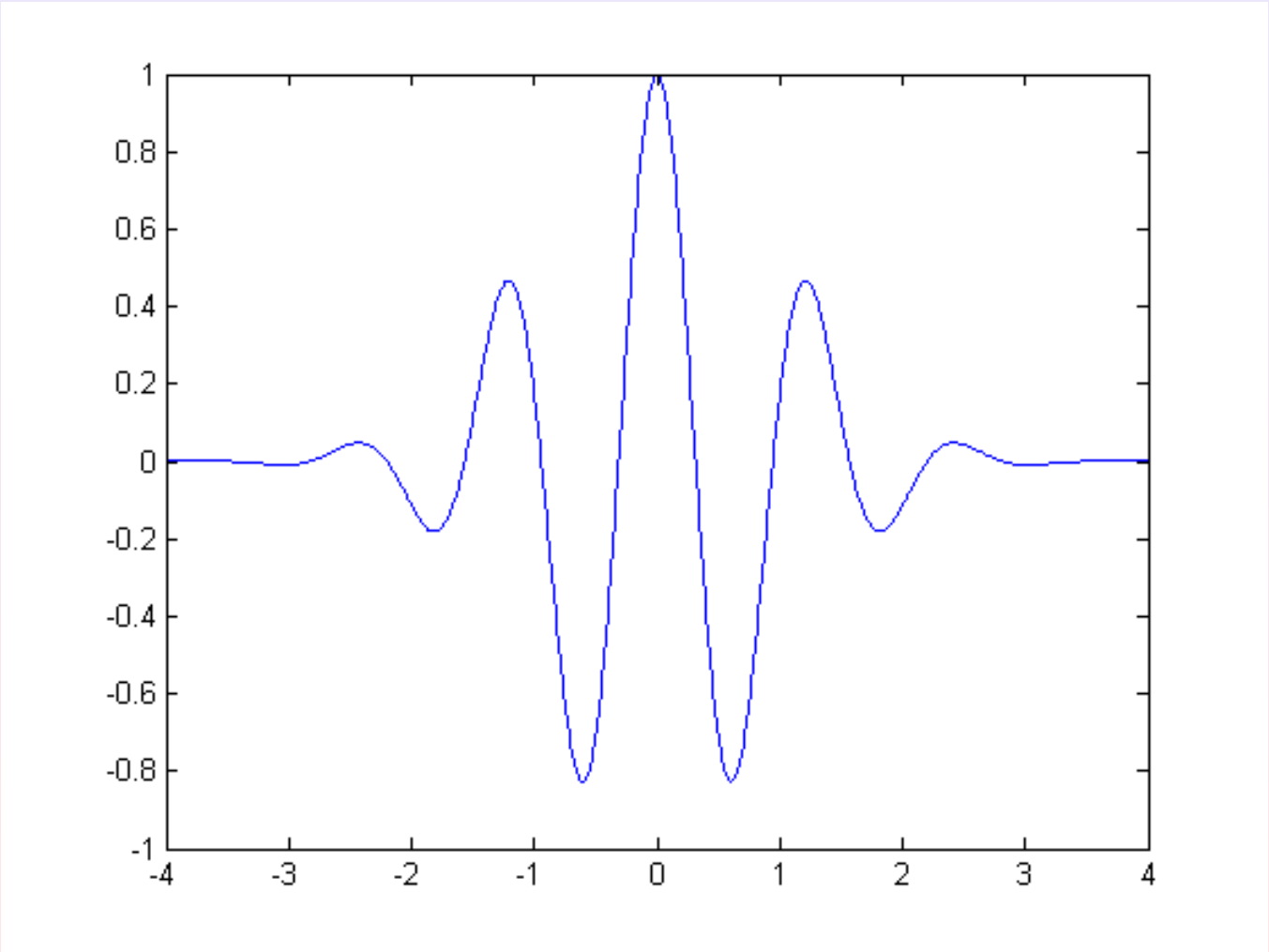
- Representation of any finite–energy signal $x(t) \in L^2(\mathbb{R})$ by means of its inner products $\{x_{m,n}\}_{m,n\in\mathbb{Z}}$ with the wavelets $\{\psi_{m,n}(t)\}_{m,n\in\mathbb{Z}}$:

$$x_{m,n} = \int x(t)\cdot\psi_{m,n}(t)dt = \int x(t)\cdot a_0^{-m/2}\psi\left(a_0^{-m}t - nb_0\right) dt \quad (8)$$

- Orthonormal dyadic wavelet basis
  - $a_0 = 2$ and $b_0 = 1$
  - Stringent constraints on the mother wavelet

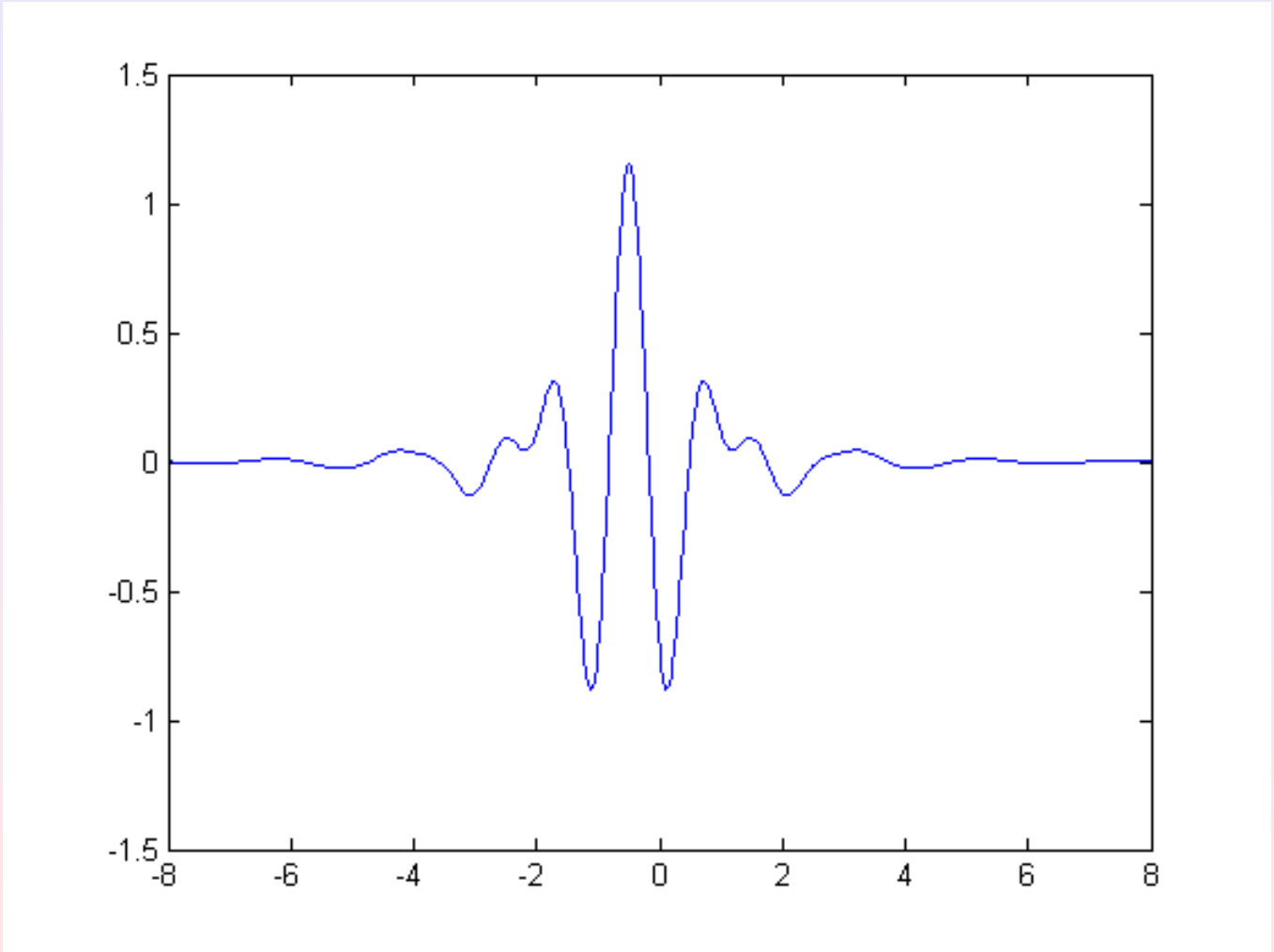
Mother Wavelet

Morlet



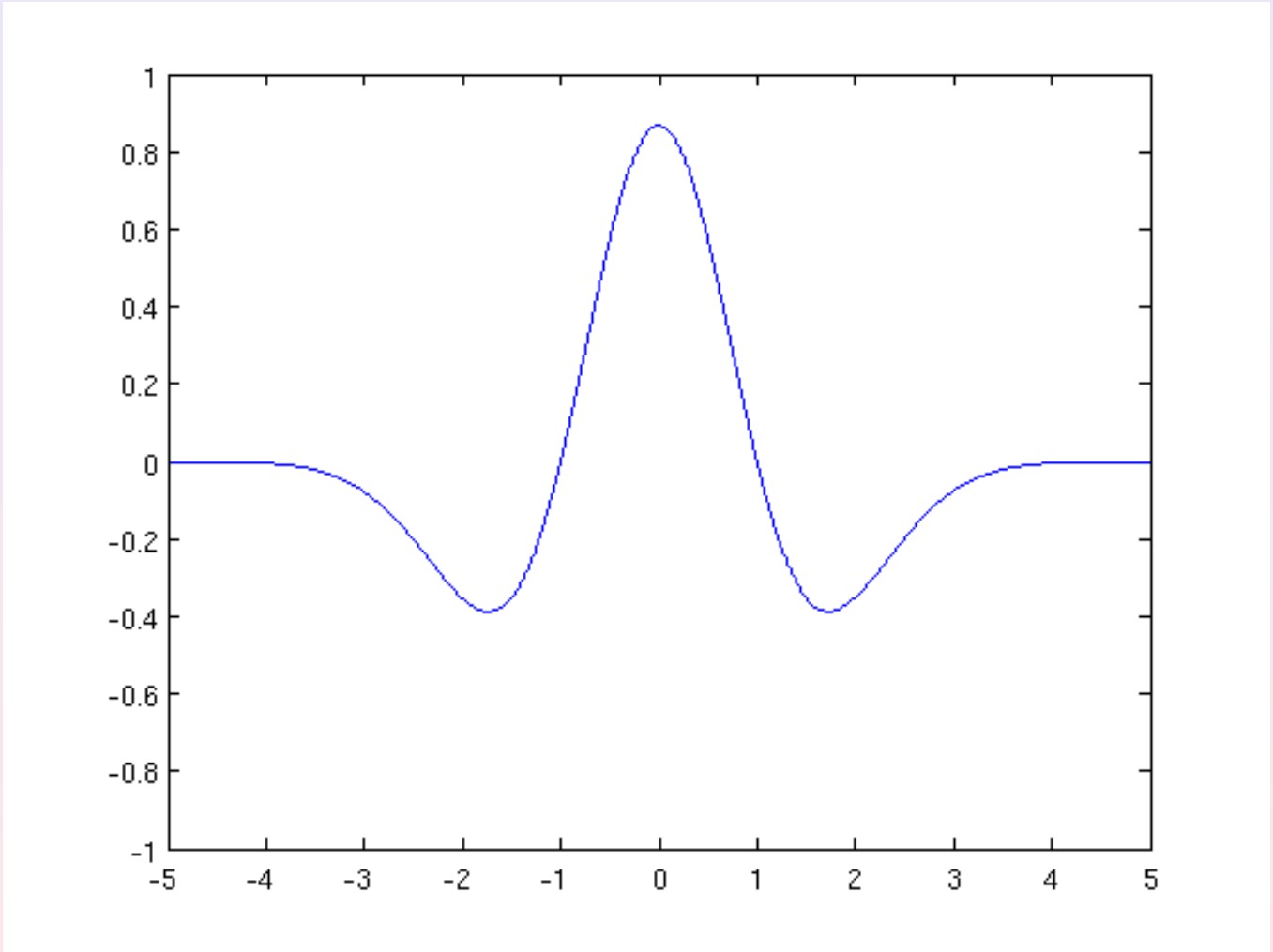
Mother Wavelet

Meyer



Mother Wavelet

Mexican Hat



Filter bank implementation of the Wavelet Transform

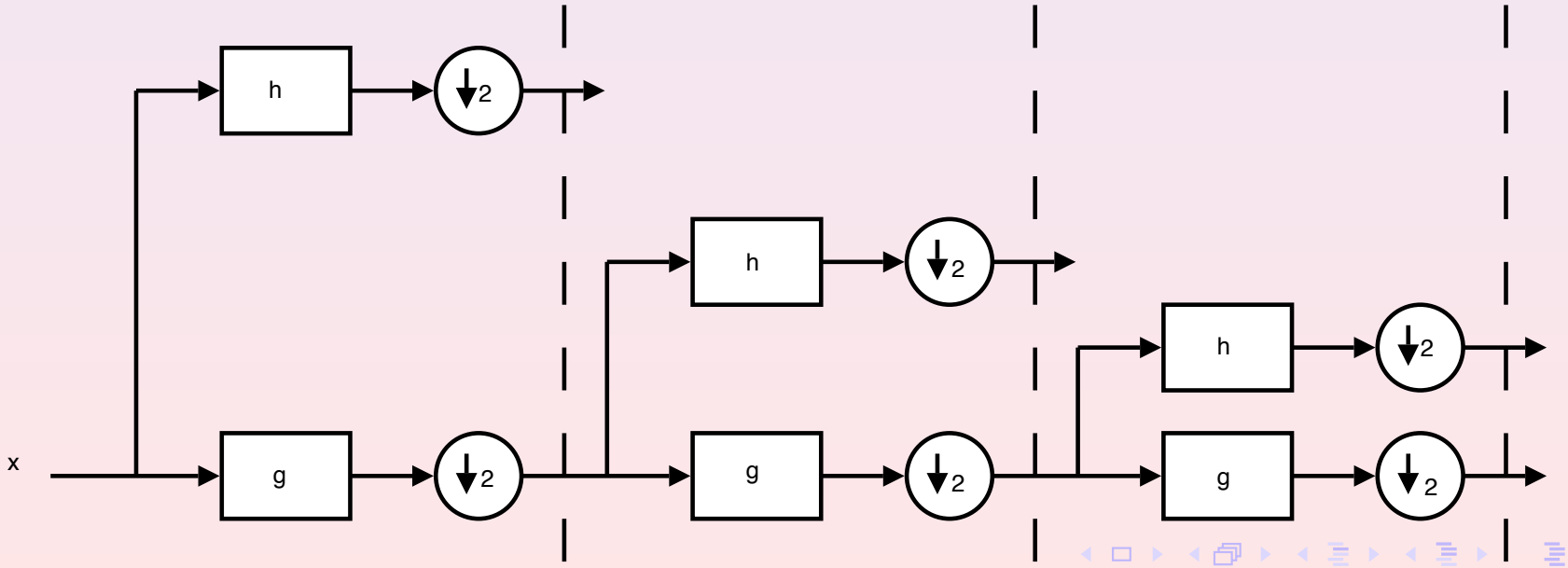
- Two scale difference equation

$$\psi(t) = \sqrt{2}\sum_n g_n \phi(2t-n) \qquad \phi(t) = \sqrt{2}\sum_n h_n \phi(2t-n)$$

where

$$g_n = (-1)^{n-1} h_{-n-1}$$

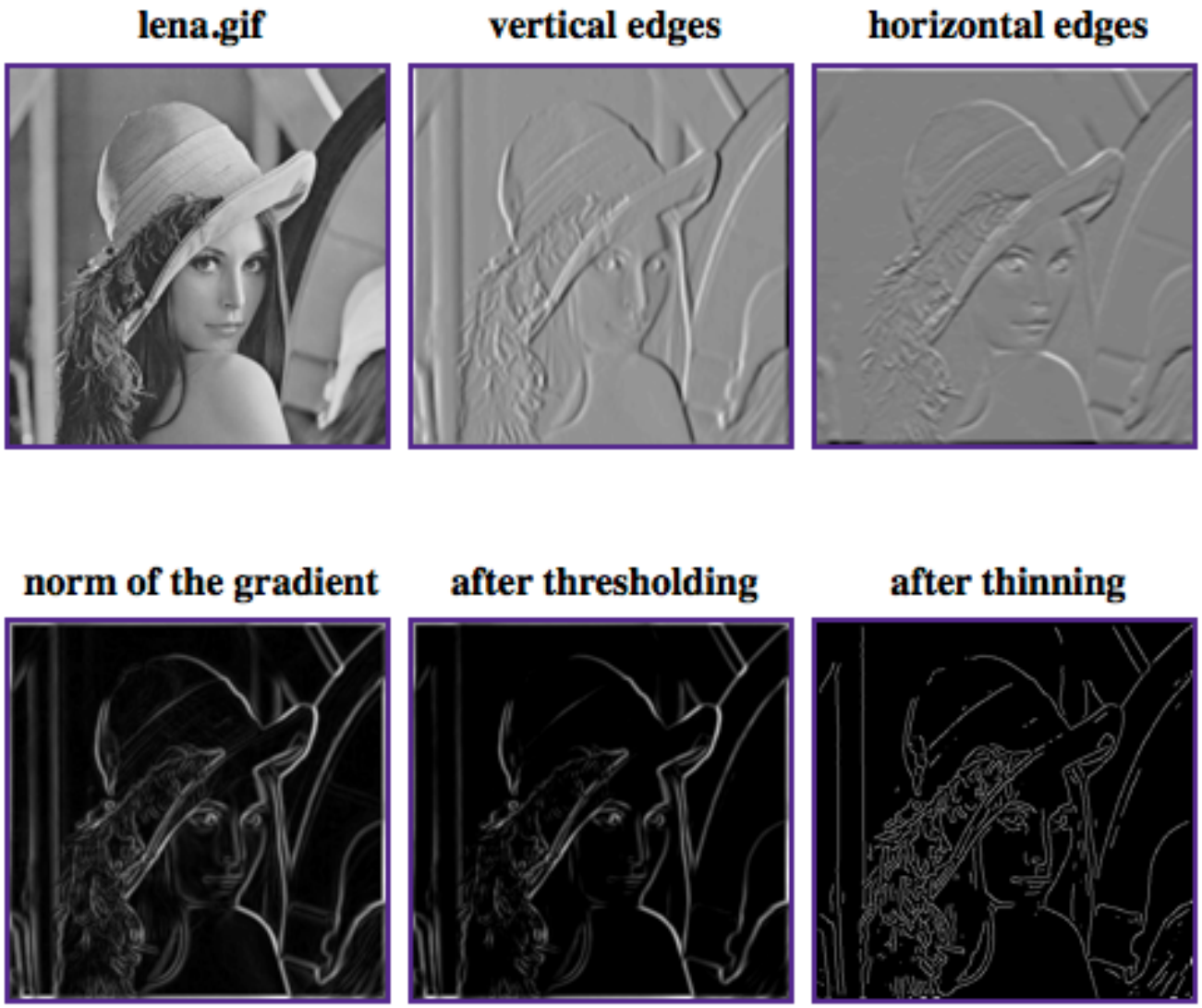
- Let $\boldsymbol{x} = (x_1, x_2, \ldots)$ denote the approximation of a finite–energy signal $x(t)$



Wavelet and Edge Detection

- An edge in an image is a contour across which the brightness of the image changes abruptly
- In image processing,an edge is often interpreted as one class of singularities
- In a function, singularities can be characterized easily as discontinuities where the gradient approaches infinity
- However, image data is discrete, so edges in an image often are defined as the local maxima of the gradient
- Wavelet transform has been found to be a remarkable tool to analyze the singularities including the edges and to detect them effectively

Wavelet and Edge Detection



Wavelet and Anomaly Detection

- The concept of edge can be easily extended to that of anomaly in network traffic
- Classical approaches look at the time series of specific kinds of packets inside aggregate traffic
- They detect irregular traffic patterns in traffic trace

Wavelet analysis is applied to evaluate the traffic signal filtered only at certain scales, and a thresholding technique is used to detect changes

A case study

A Signal Analysis of Network Traffic Anomalies
Paul Barford, Jeffery Kline, David Plonka and Amos Ron
ACM Internet Measurement Workshop 2002

The Measurement Data:

- SNMP and IP flow data
- collected at the border router (Juniper M10) of the University of Wisconsin-Madison campus network
- the campus network consists primarily of four IPv4 class B networks or roughly 256,000 IP addresses of which fewer than half are utilized
- IP connectivity to the commodity Internet and to research networks via about 15 discrete wide-area transit and peering links all of which terminate into the aforementioned router

SNMP Data

- The SNMP data was gathered by MRTG at a five minute sampling interval which is commonly used by network operator
- The SNMP data consists of the High Capacity interface statistics, defined by RFC2863, which were polled using SNMP version 2c
- Byte and packet counters for each direction of each wide-area link, specifically these 64-bit counters: ifHCInOctets, ifHCOutOctets, ifHCInUcastPkts, and ifHCOutUcastPkt

IP Flow Data

- The flow data was gathered using flow-tools and was post-processed using FlowScan
- The Juniper M10 router was running JUNOS 5.0R1.4, and later JUNOS 5.2R1.4, and was configured to perform “cflowd” flow export with a packet sampling rate of 96
- This caused 1 of 96 forwarded packets to be sampled, and subsequently assembled into flow records similar to those defined by Cisco’s NetFlow version 5 with similar packet-sampling-interval and 1 minute flow active-timeout
- Data were post-processed, so as to store mean value (over 5 minutes time-bins) of rate and packet dimension

Anomalies

By manual inspecting the data, 109 anomalies were identified:

- 41 Network Events
- 46 Attacks
- 4 Flash Crowds
- 18 Measurement Events

Necessity of filtering out the daily and weekly variations

The Method

- Framelet system, i.e. a redundant wavelet system (which essentially means that r, the number of high-pass filters, is larger than 1; a simple count shows that, if $r > 1$, the total number of wavelet coefficients exceeds the length of the original signal)
- In chosen system, there is one low-pass filter L and three high-pass filters H_1 , H_2 , H_3

The analysis platform

Derive from a given signal x (that represents five-minute average measurements over a 2 months period) three output signals, as follows

- The L(ow frequency)-part of the signal: all the low-frequency wavelet coefficients from levels 9 and up
  - should capture patterns and anomalies of very long duration: several days and up
  - signal here is very sparse (its number of data elements is approximately 0.4% of those in the original signal), and captures weekly patterns in the data quite well
  - for many different types of Internet data, the L-part of the signal reveals a very high degree of regularity and consistency in the traffic, hence can reliably capture anomalies of long duration

The analysis platform

- The M(id frequency)-part of the signal: the wavelets coefficients from frequency levels 6, 7, 8
  - has zero-mean
  - is supposed to capture mainly the daily variations in the data
  - data elements number about 3% of those in the original signal
- The H(igh frequency)-part of the signal: obtained by thresholding the wavelet coefficients in the first 5 frequency levels
  - need for thresholding stems from the fact that most of the data in the H-part consists of small short-term variations, variations that we think of as “noise”

Detection of anomalies

- Normalize the H- and M-parts to have variance one
- Compute the local variability of the (normalized) H- and M-parts by computing the variance of the data falling within a moving window of specified size
- The length of this moving window should depend on the duration of the anomalies that we wish to captured
  - If we denote the duration of the anomaly by t_0 and the time length of the window for the local deviation by t_1, we need, in the ideal situation, to have $q = t_0/t_1 = 1$
  - If the quotient q is too small, the anomaly may be blurred and lost
  - If the quotient is too large, we may be overwhelmed by anomalies that are of very little interest

Detection of anomalies

- Combine the local variability of the H- part and M- part of the signal using a weighted sum. The result is the V(ariable)-part of the signal
- Apply thresholding to the V-signal. By measuring the peak height and peak width of the V-signal, one is able to begin to identify anomalies, their duration, and their relative intensity
- Needed Parameters:
  - M- window size
  - H- window size
  - weights assigned to the M- and H-parts
  - threshold

Experimental Results

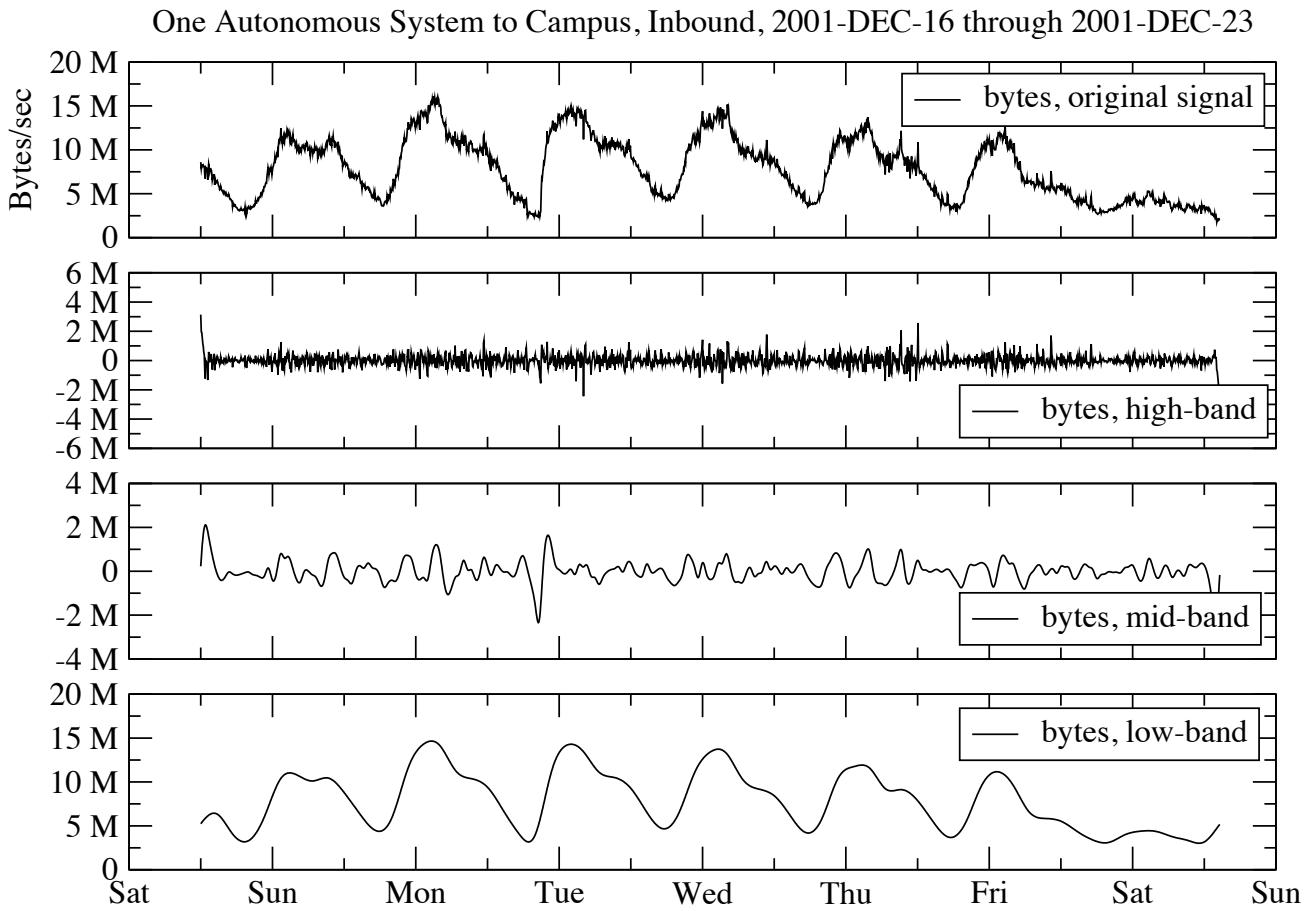


Fig. 1. Aggregate byte traffi c from IP fbw data for a typical week plus high/mid/low decomposition.

Experimental Results

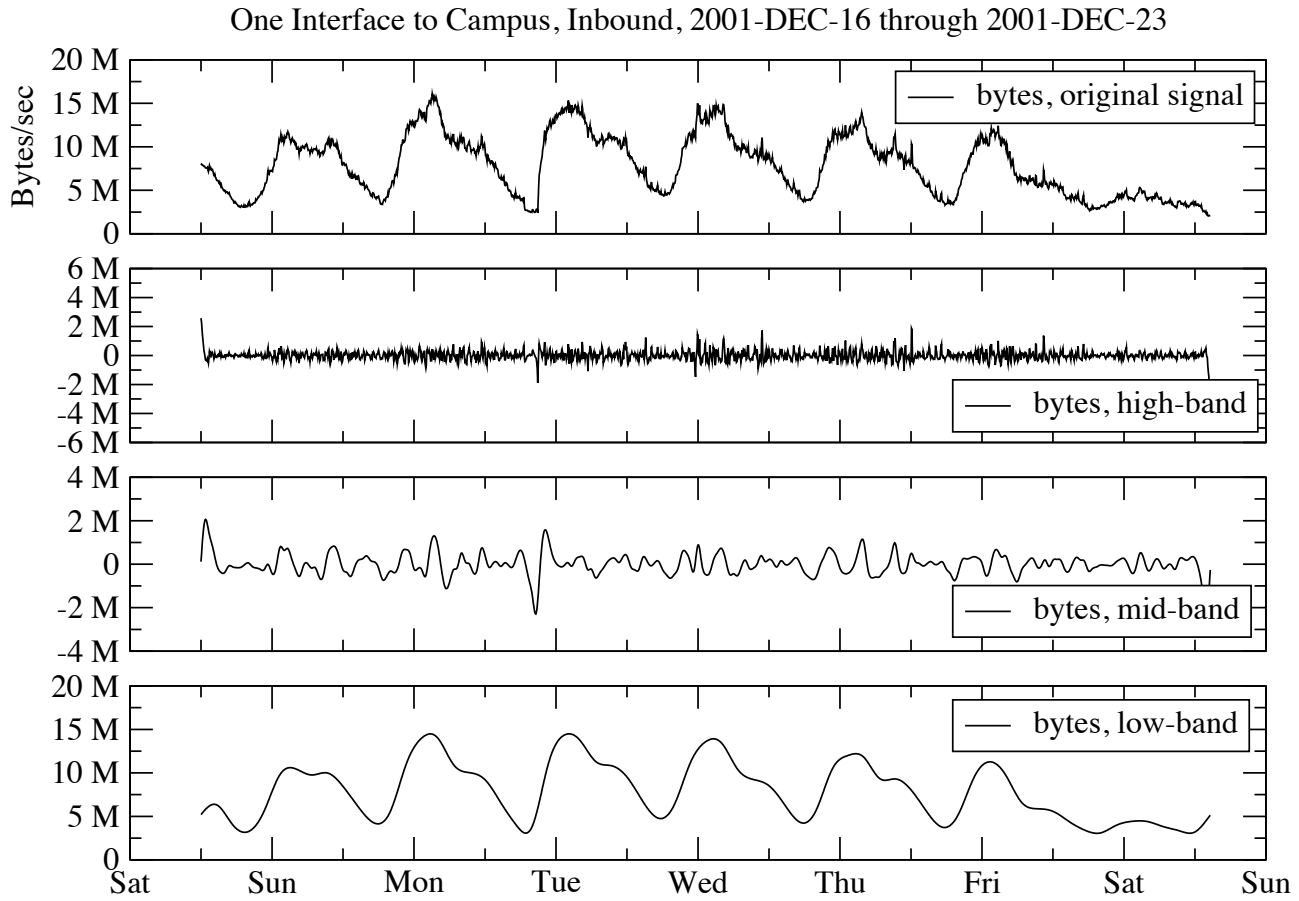


Fig. 2. Aggregate SNMP byte traffi c for the same week as Figure 1 plus high/mid/low decomposition.

Experimental Results

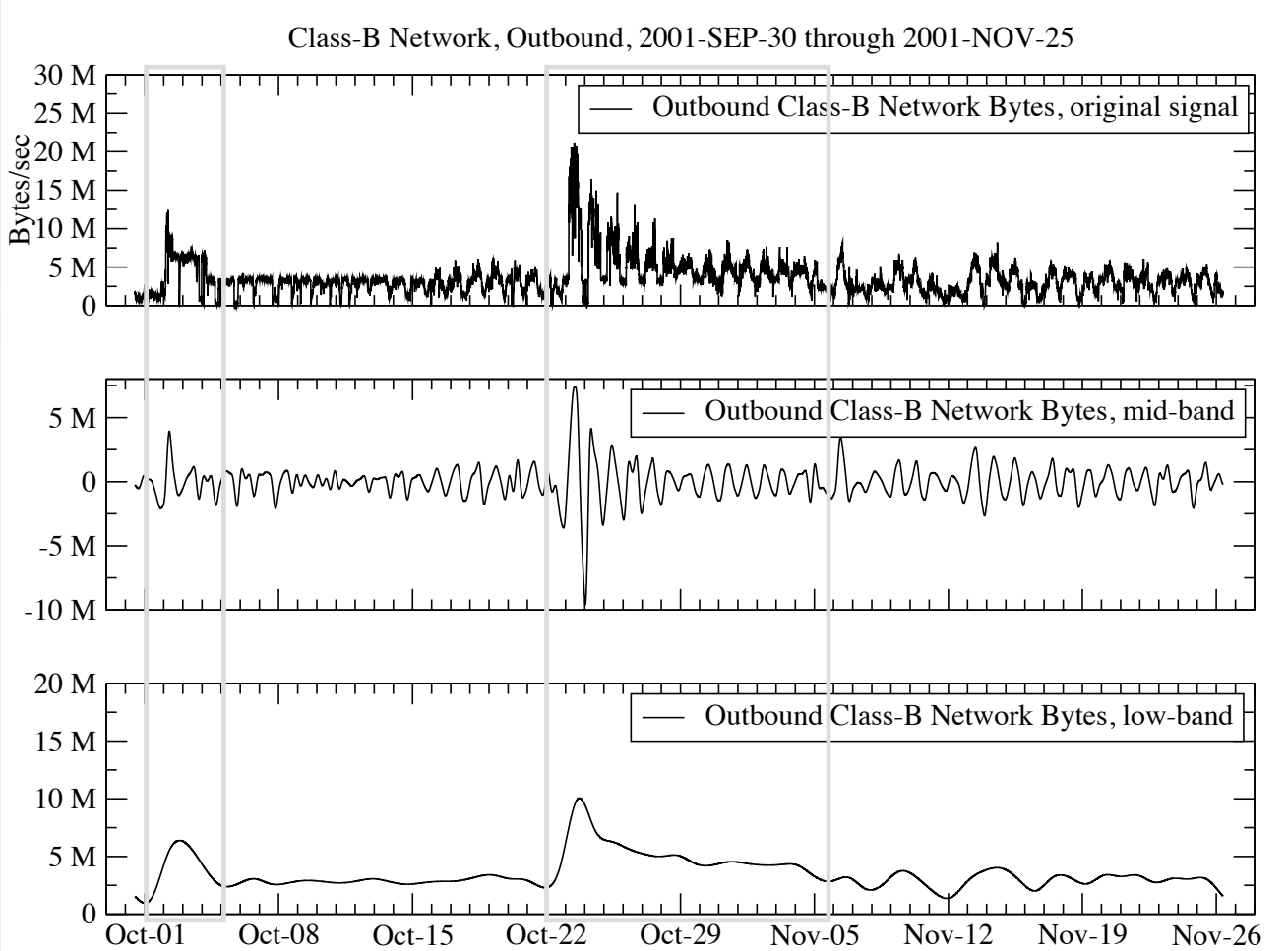


Fig. 3. Baseline signal of byte traffi c for a one week on either side of a flash crowd anomaly caused by a software release plus high/mid/low decomposition.

Experimental Results

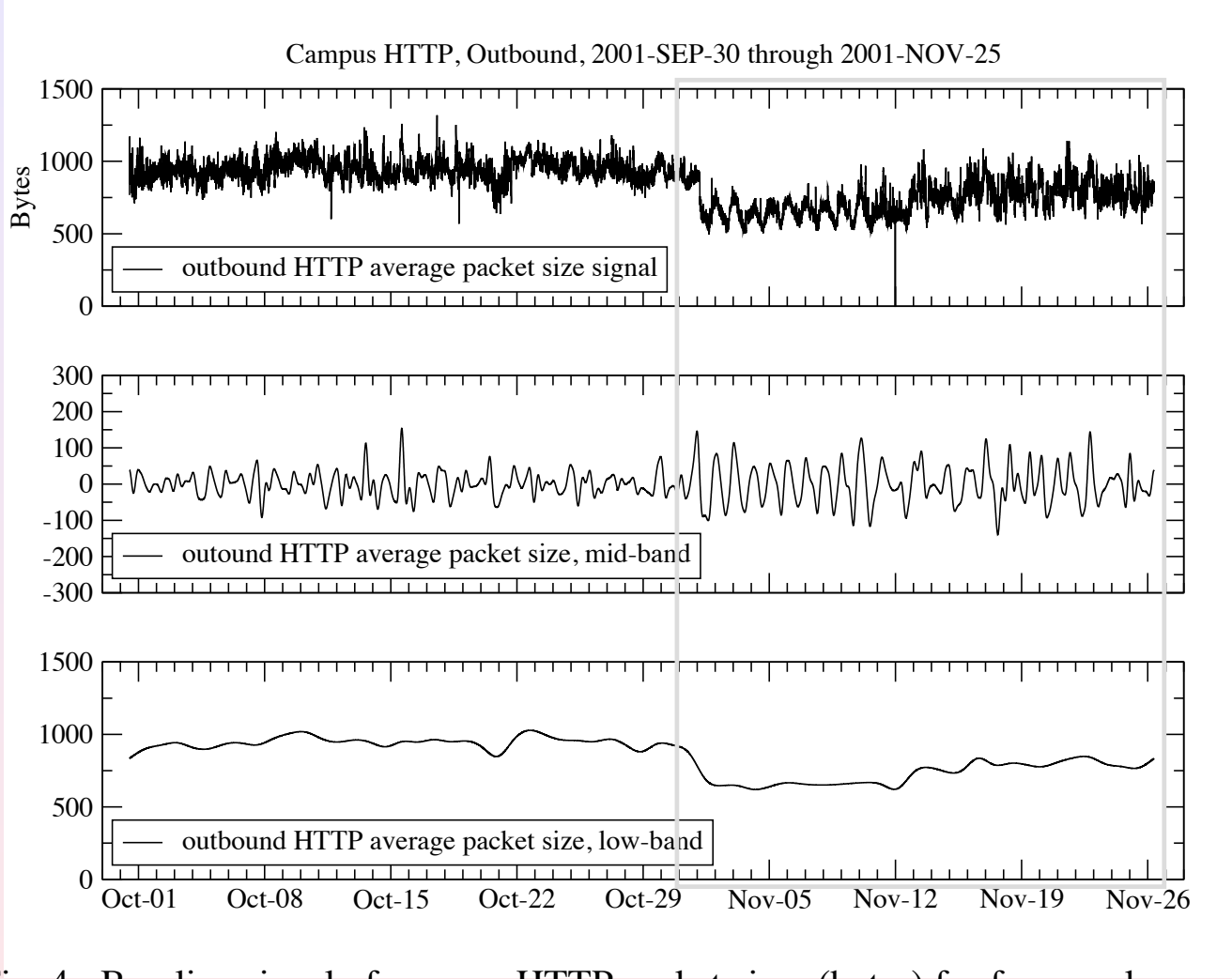


Fig. 4. Baseline signal of average HTTP packet sizes (bytes) for four weeks on either side of a flash crowd anomaly plus mid/low decomposition.

Experimental Results

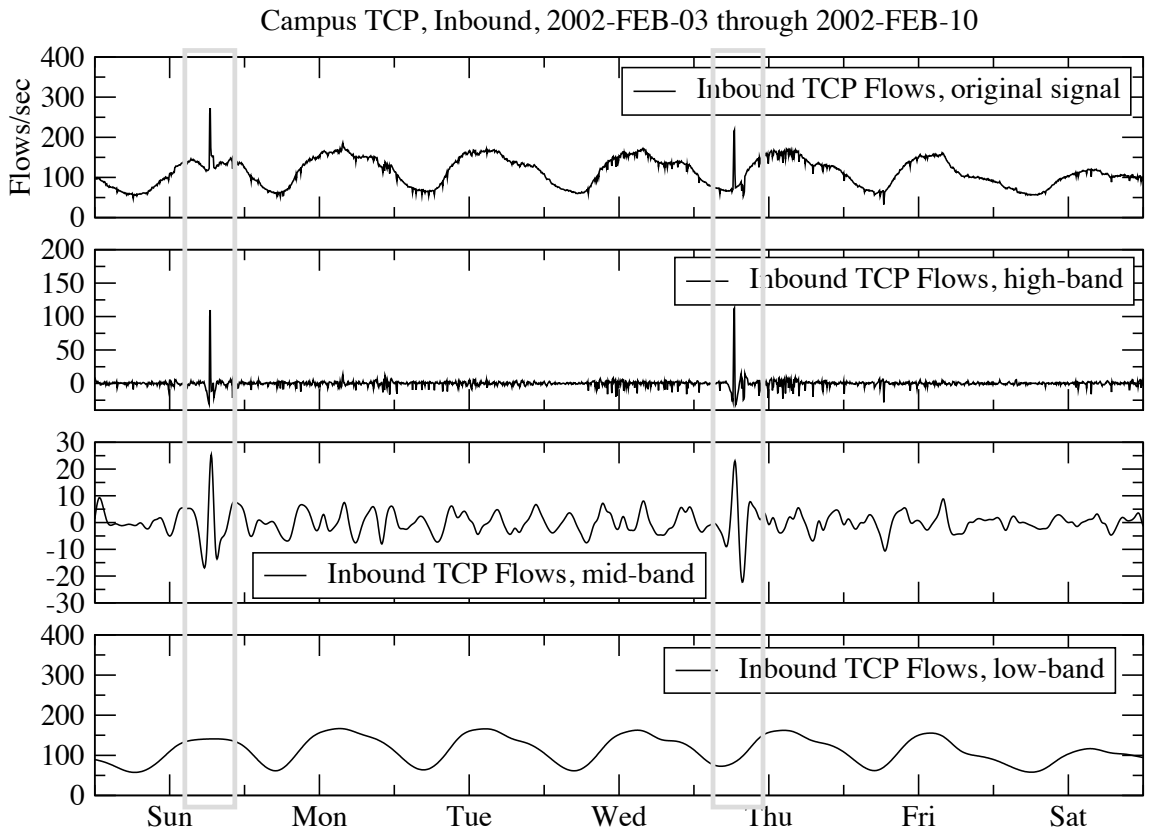


Fig. 5. Baseline signal of packet flows for a one week period highlighting two short-lived DoS attack anomalies plus high/mid/low decomposition.

Experimental Results

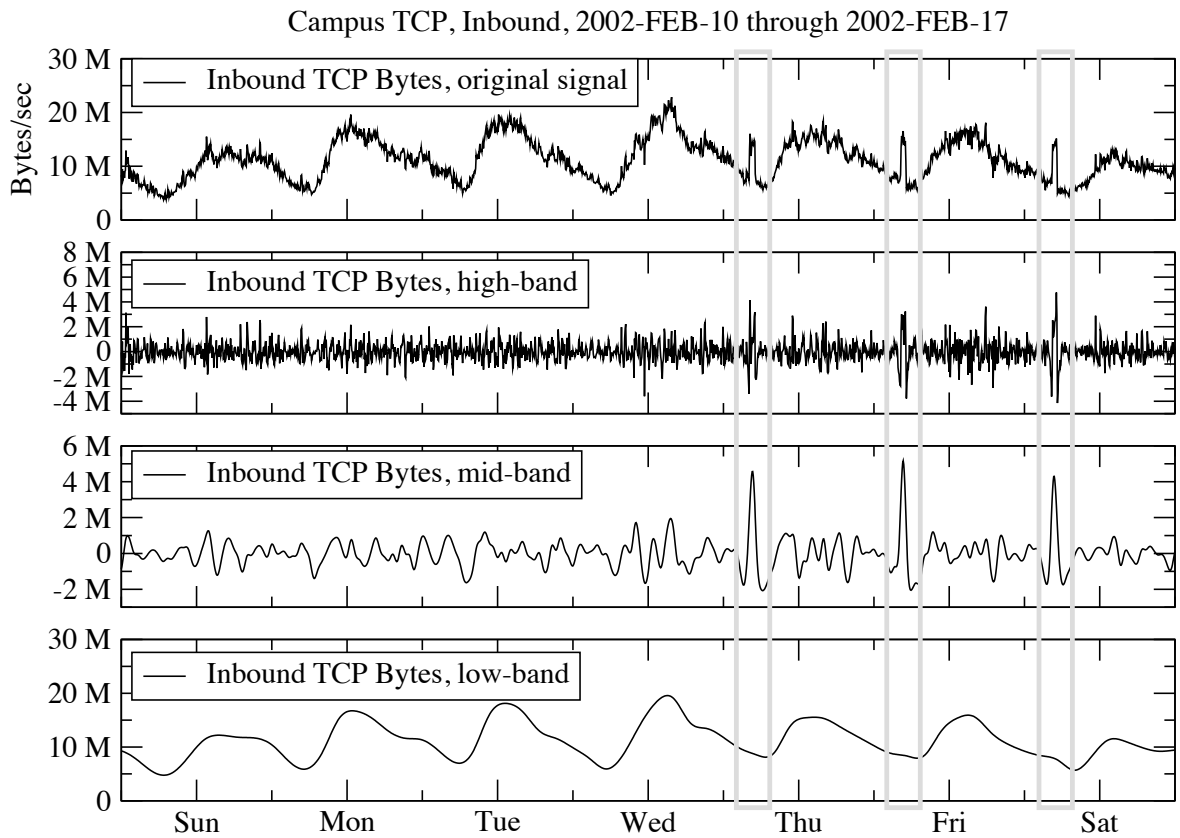


Fig. 6. Baseline signal of byte traffi c from flow data for a one week period showing three short-lived measurement anomalies plus high/mid/low decomposition.

Experimental Results

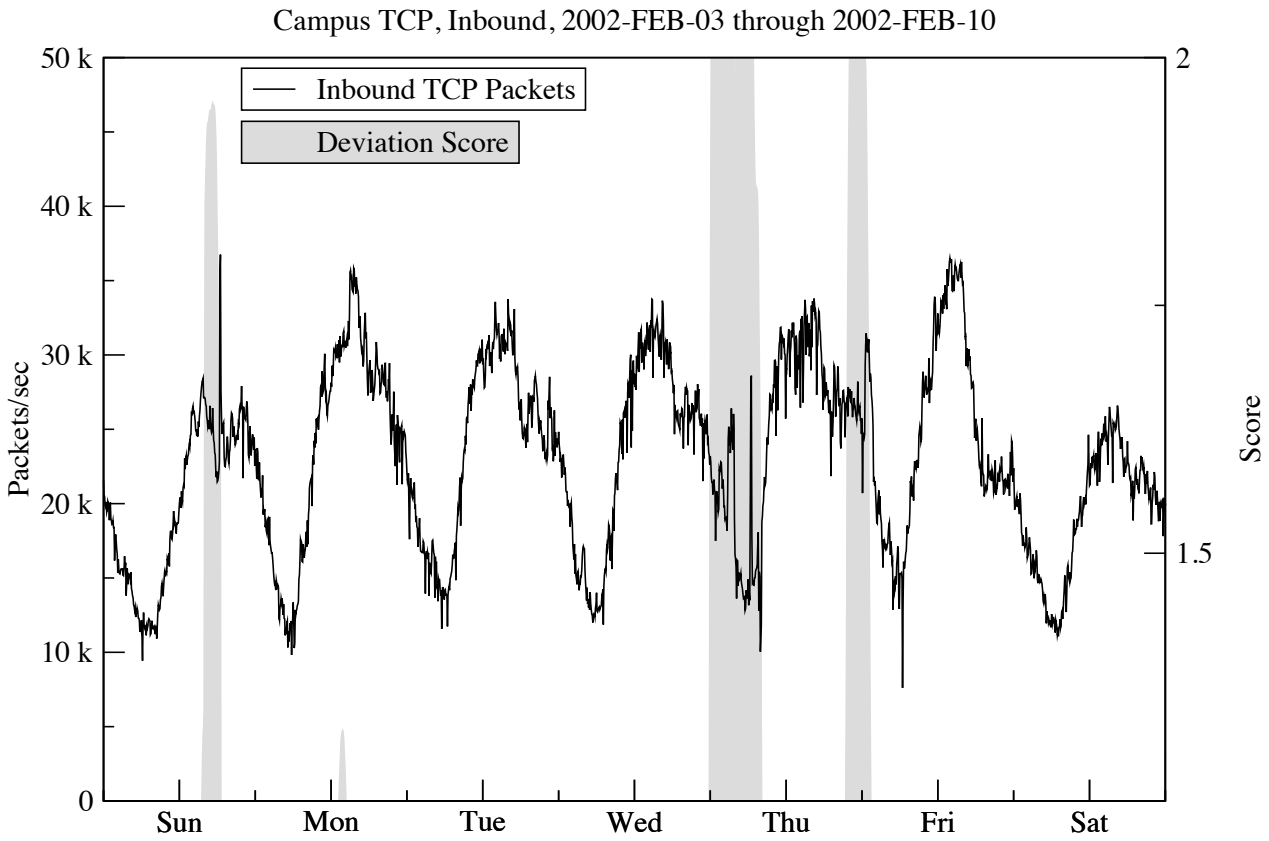


Fig. 7. Deviation analysis exposing two DoS attacks and one measurement anomaly in for a one week period in packet count data.

Experimental Results

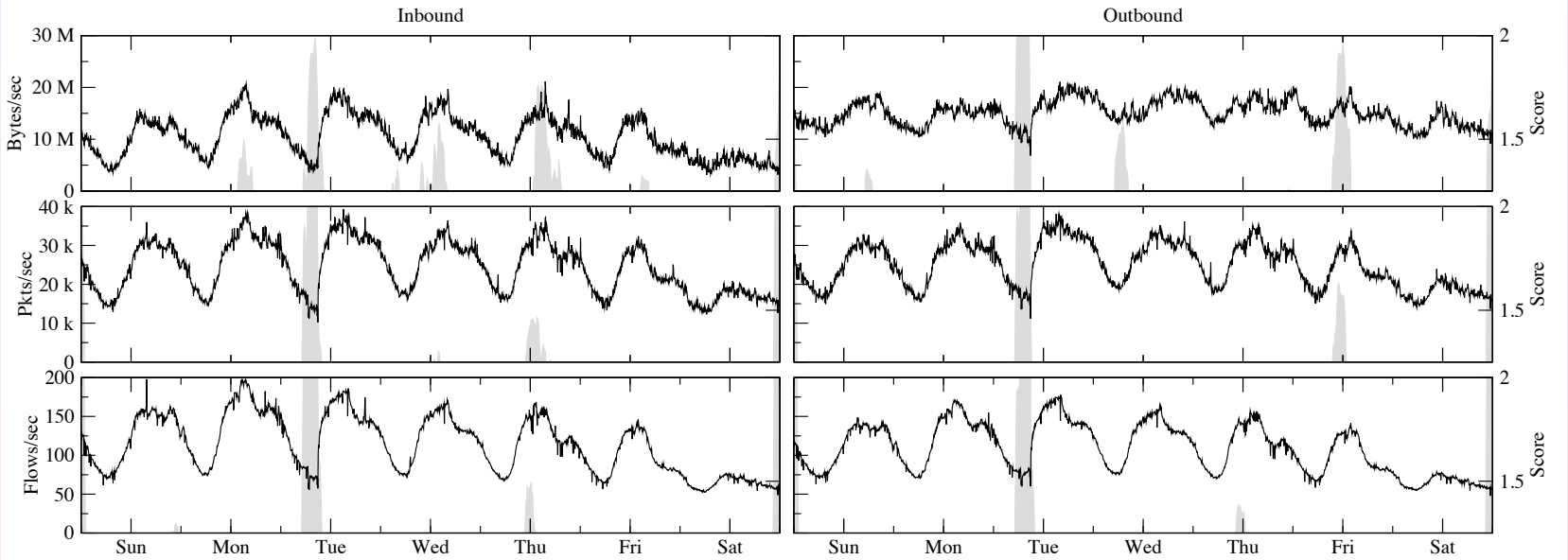
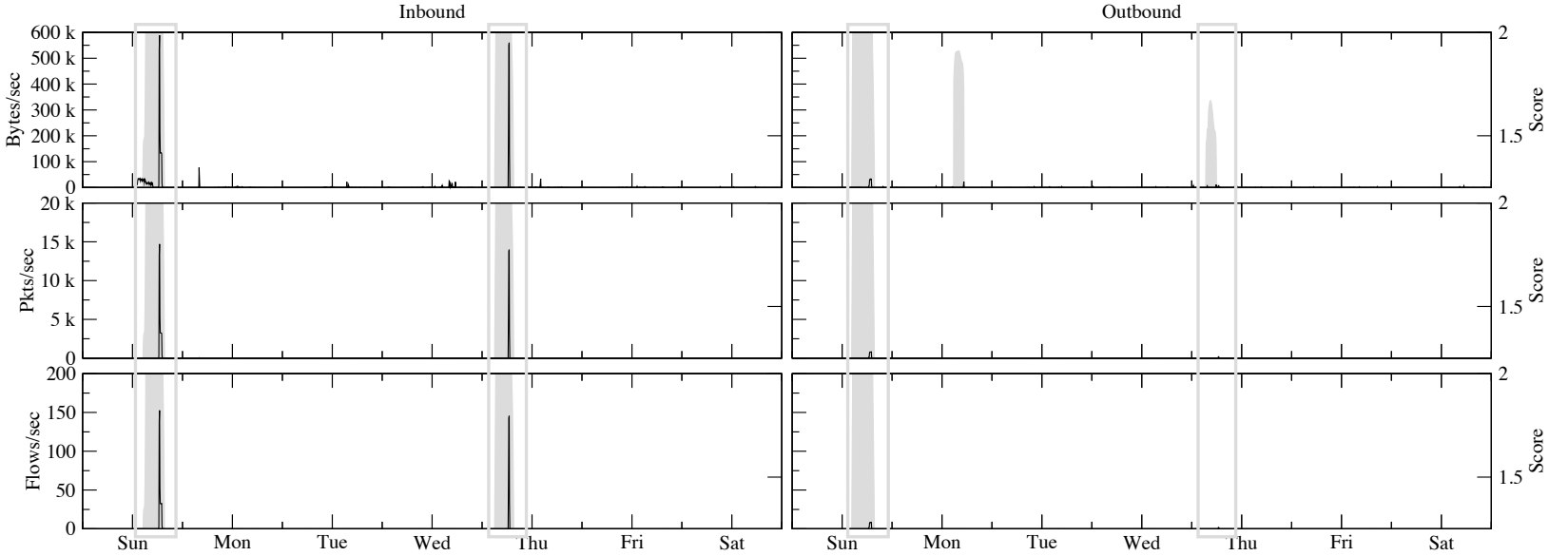


Fig. 8. Deviation analysis exposing a network outage of one (of four) Class-B networks.

Experimental Results



Experimental Results

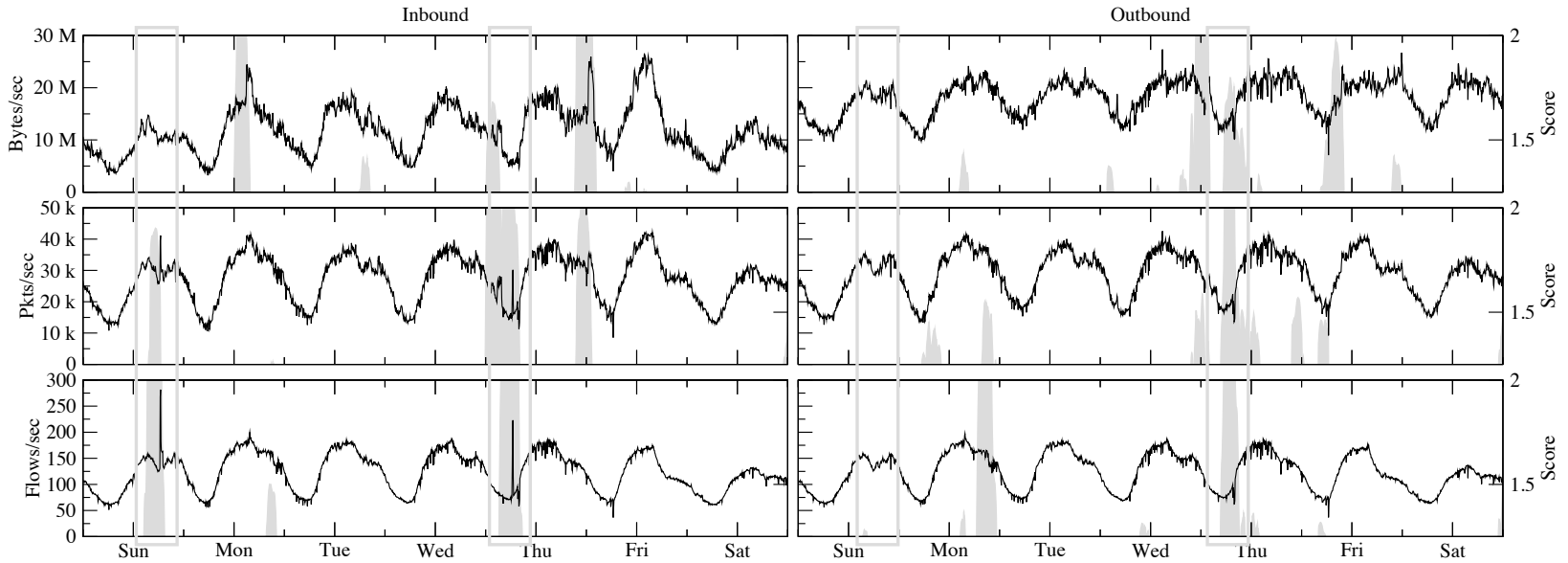


Fig. 9. Deviation analysis of two DoS events as seen in the 254 host subnet containing the victim of the attack (top) and in the aggregate traffi c of the entire campus.

Experimental Results

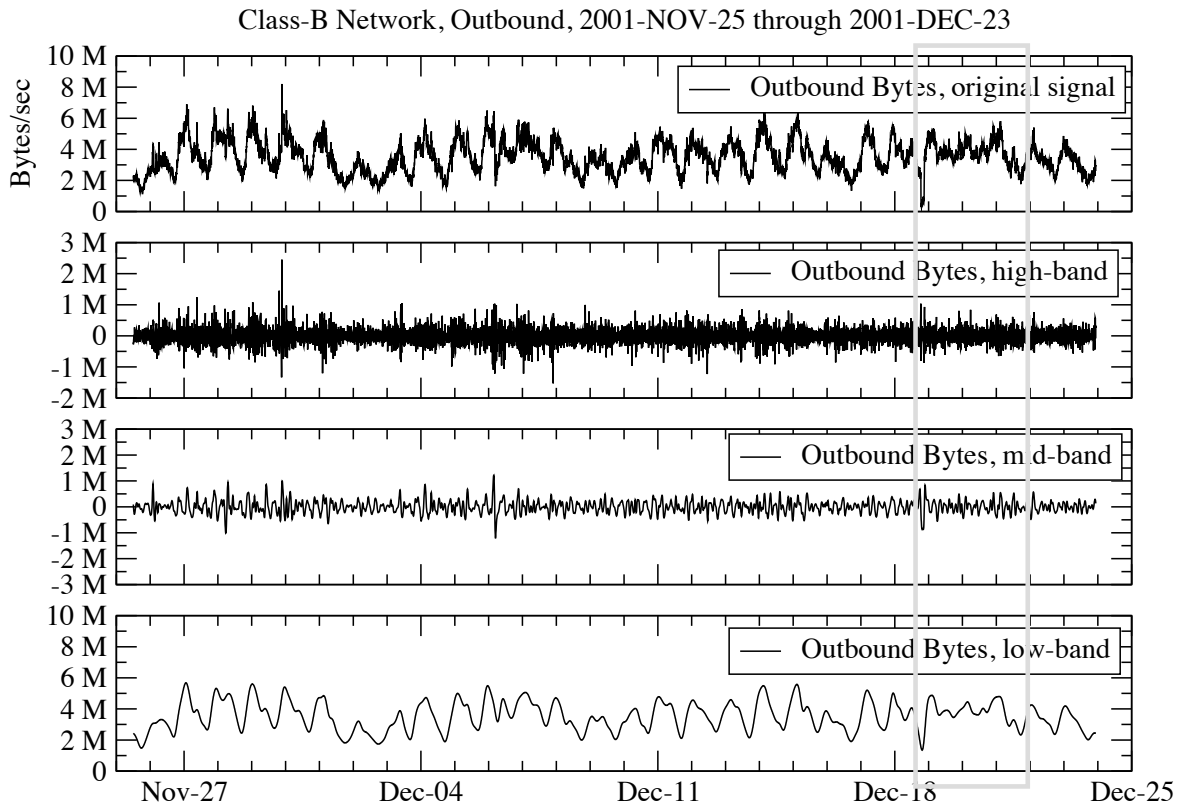


Fig. 10. Example of three-band analysis exposing a multi-day network abuse anomaly.

References

- *P.Barford,J.Kline,D.Plonka,A.Ron* , **A signal analysis of network traffic anomalies**, ACM SIGCOMM InternetMeasurement Workshop, 2002
- *P. Huang, A. Feldmann, W. Willinger* , **A non-intrusive, wavelet-based approach to detecting network performance problems**, ACM SIGCOMM Internet Measurement Workshop, 2001
- *L. Li, G. Lee* , **DDos attack detection and wavelets**, IEEE ICCCN, 2003
- *A. Dainotti, A. Pescape', and G. Ventre* , **Wavelet-based Detection of DoS Attacks**, IEEE Globecom, 2006
- *Christian Callegari, Stefano Giordano, and Michele Pagano* , **Application of Wavelet Packet Transform to Network Anomaly Detection**, nternational Conference on Next Generation Teletraffic and Wired/Wireless Advanced Networking (NEW2AN), 2008

Thank You for your attention