Server-side Prediction of Source IP Addresses using Density Estimation

ARES 2009 Conference

Markus Goldstein[1], Matthias Reif[1]
Armin Stahl[1], Thomas M. Breuel[1,2]

[1]German Research Center for Artificial Intelligence (DFKI),
Kaiserslautern, Germany
[2]Technical University of Kaiserslautern, Germany

March 16, 2009

Outline

Introduction

Overview

- Predict whether a source IP of a new incoming connection is likely to appear

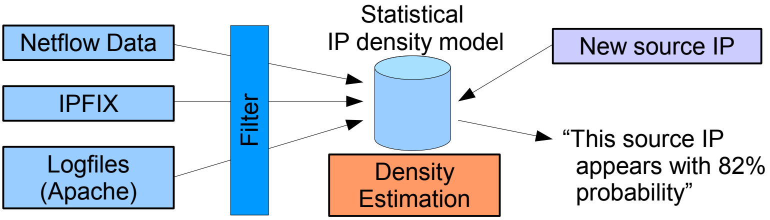
Applications

- Quality of Service (QoS)
- Click fraud detection
- Optimizing request routing in P2P networks
- **DDoS Mitigation**

Introduction

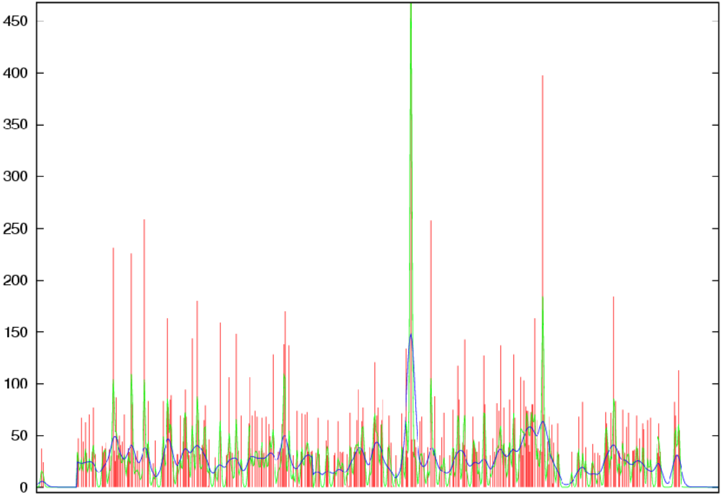
Overview

- Training phase: filter data, compute density estimation
- Test phase: classify new connections



Introduction

Density Estimation



Reviewing Existing Methods

- Models are often used implicitly
- Compute the probability density function (PDF):

$$\sum_{i=0}^{N-1} P(S = s_i) = 1 \tag{1}$$

where $P(S = s_i) = p_i$ is the probability of an IP address s_i to be a source IP address that will occur in the future

Reviewing Existing Methods

History-based IP Filtering [Peng et al., 2003]

- Motivation: "Code Red Worms" [Jung et al., 2002]
- Normal operation: 17.1% - 53.3% new IPs
- During Code Red Worm Attack: 86.0% - 99.4% new IPs

Reviewing Existing Methods

History-based IP Filtering [Peng et al., 2003]

- Mitigating DDoS attacks
- Store all source IPs during training in an address database
- Classification rule: seen previously or not
- No density estimation
- PDF:

$$f(s) = \frac{\min(n_s, 1)}{\sum_{i=0}^{N-1} \min(n_{s_i}, 1)} \quad (2)$$

Reviewing Existing Methods

Adaptive History-based IP Filtering [Goldstein et al., 2008]

- AHIF uses histograms with bin size of network masks (/16 ... /24)
- Similar ideas with fixed bin sizes (e.g. /16 in PacketScore)
- Density estimation by bin width and counting
- Adaptivity by selecting proper network mask and PDF threshold
- PDF:

$$f(s) = \frac{n_s}{\sum_{i=0}^{N-1} n_{s_i}} \tag{3}$$

Reviewing Existing Methods

Clustering of Source Address Prefixes [Pack et al., 2006]

- Uses hierarchical clustering to estimate densities
- Adaptivity by stopping aggregation at a certain point
- But: too compute intense since all distances must be calculated
- Not applicable on our data set (1.3 m IPs $\rightarrow$ 3TB)

Distance Measures

Euclidean distance

- $\Delta_{Eucl}(s_i, s_j) = |s_i - s_j|$
- Does not take network boundaries into account
- e.g. 1.1.1.1 and 1.1.1.3 have a larger distance than 1.1.1.255 and 1.1.2.1

Distance Measures

Xor Distance

- Introduced with hierarchical clustering [Pack et al., 2006]
- Takes network boundaries into account
- $\Delta_{Xor}(s_i, s_j) = 2^{\lfloor \log_2(s_i \oplus s_j) \rfloor}$
- Xor the two IP addresses together and use the highest order bit set as distance
- Example:

346	101011010
365	101101101
32	000100000

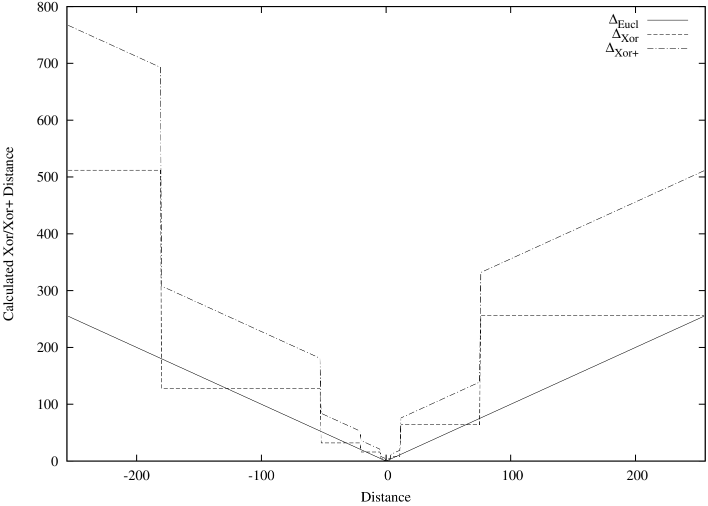
- Distances within a specific network mask are constant

Distance Measures

Xor+ Distance

- Takes network boundaries into account
- Use euclidean distance in addition within the same network mask
- $\Delta_{Xor+}(s_i, s_j) = 2^{\lfloor \log_2(s_i \oplus s_j) \rfloor} + |s_i - s_j|$
- Distance function is not continuous, but still is a (mathematical) metric
- Mean is still computable

Distance Measures



Density Estimation: k-means

- K-means cuts down memory requirements from $O(M^2)$ to $O(M \cdot K)$
- After finding the cluster centers (in dense areas), a variable surrounding area has to be defined.

Area Growing

- Reduce network prefix length [Pack et al., 2006]
- Same size for dense and less dense areas

Weighted Area Growing

- Grow areas with respect to the number of IPs belonging to that cluster
- $w_j = \frac{b_j}{\sum_{i=1}^{k} b_i}$

Subnet Boundary Sensitive Smoothing

Idea

- Use kernel density estimation to smooth the undersampled IP space
- Create normalized histogram of source IPs
- Apply Nadaraya-Watson kernel-weighted average
  $\hat{p}_s = \frac{\sum_{i=0}^{N-1} K_\lambda(s,s_i) p_i}{\sum_{i=0}^{N-1} K_\lambda(s,s_i)}$
- Kernel: $K_\lambda(s, s_i) = D\left(\frac{\Delta(s,s_i)}{\lambda}\right)$

Subnet Boundary Sensitive Smoothing

Kernels

$$\text{Epanechnikov:} \quad D(t) = \begin{cases} \frac{3}{4}(1 - t^2) & |t| \leq 1 \\ 0 & \text{otherwise} \end{cases}$$

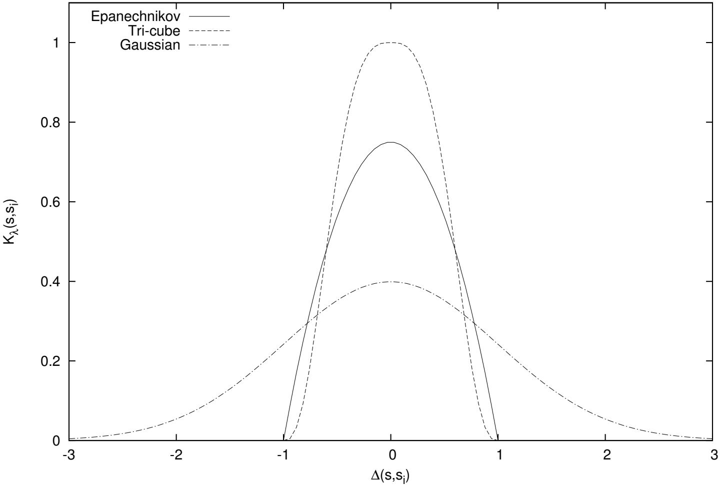
$$\text{Tri-Cube:} \quad D(t) = \begin{cases} (1 - |t|^3)^3 & |t| \leq 1 \\ 0 & \text{otherwise} \end{cases}$$

$$\text{Gaussian:} \quad D(t) = \frac{1}{\lambda\sqrt{2\pi}} e^{-\frac{1}{2}t^2}$$

- Selection of kernel depends on true distribution (unknown)

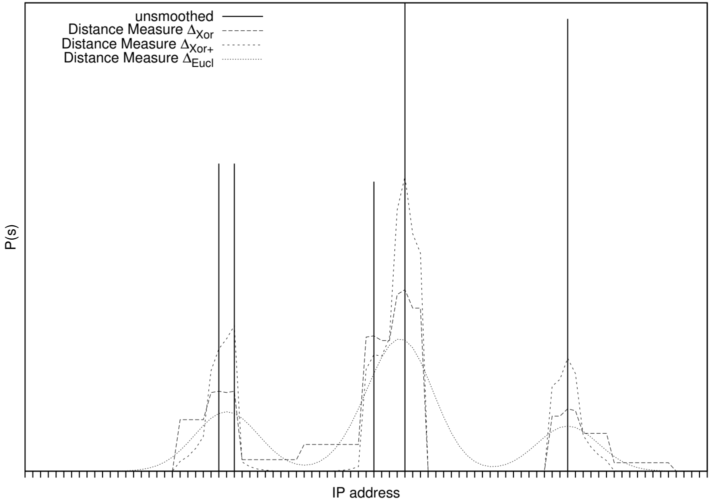
Subnet Boundary Sensitive Smoothing

Kernels



Subnet Boundary Sensitive Smoothing

Example of different Distance Measures



Evaluation

Datasets

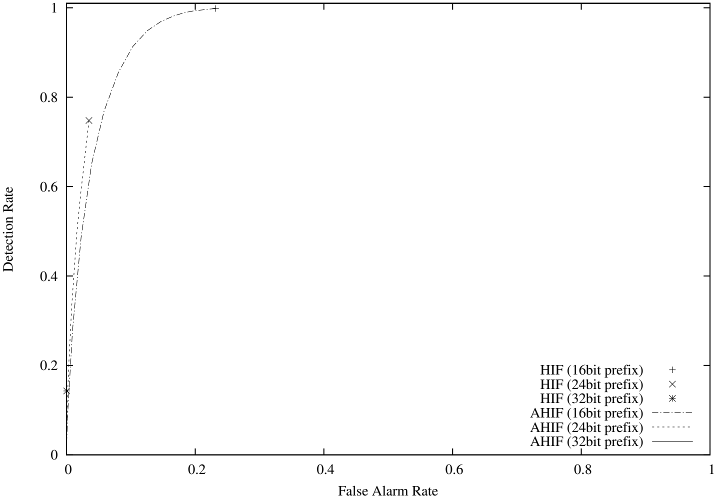
- Public datasets contain anonymized IP addresses
- Neighborship relations have to be destroyed to guarantee anonymity
- We have to use our own datasets for evaluation

Xvid.org

- 100 days logfile data (90 for training, 10 for testing)
- 53,828,308 accesses from 1,284,213 different IPs
- challenging dataset due to many new “one time visitors”
- ROC evaluation with detection rate and false alarm rate

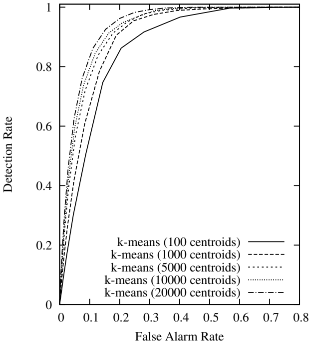
HIF and AHIF Results

Different Prefixes

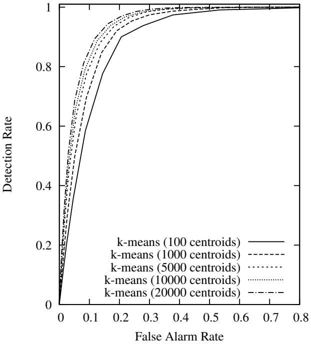


k-means Results

Different Area Growing



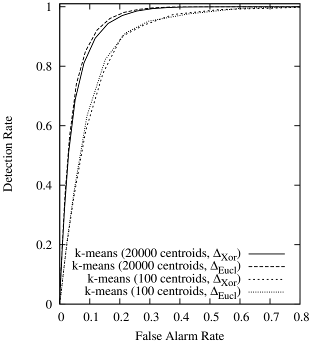
(a) Standard area growing



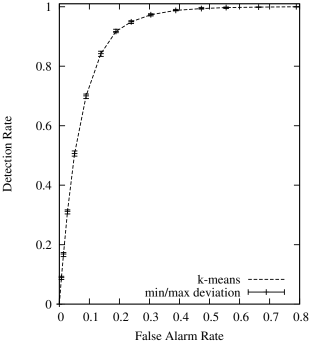
(b) Weighted area growing

k-means Results

Distance Measure and Stability



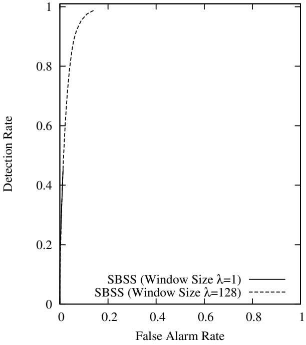
(c) Distance Comparison



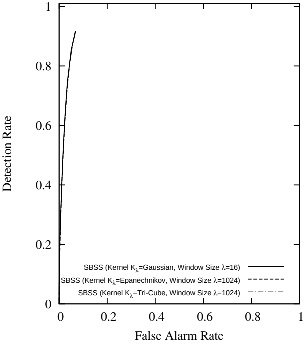
(d) Stability

SBSS Results

Window Sizes and Kernel Types



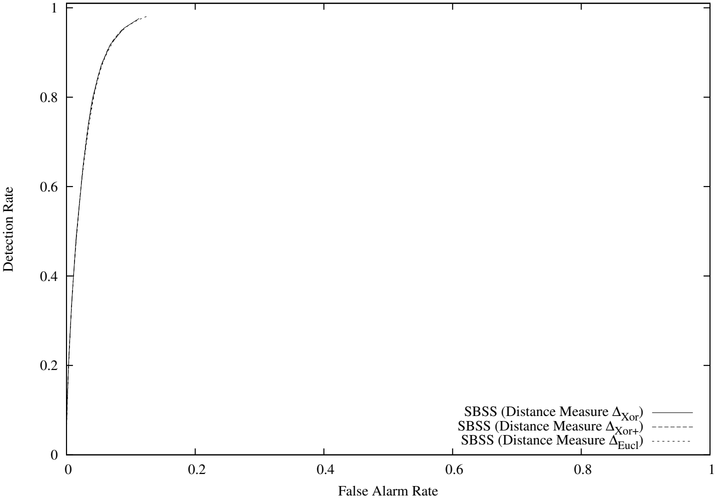
(e) Window Size λ



(f) Kernel Type K_λ

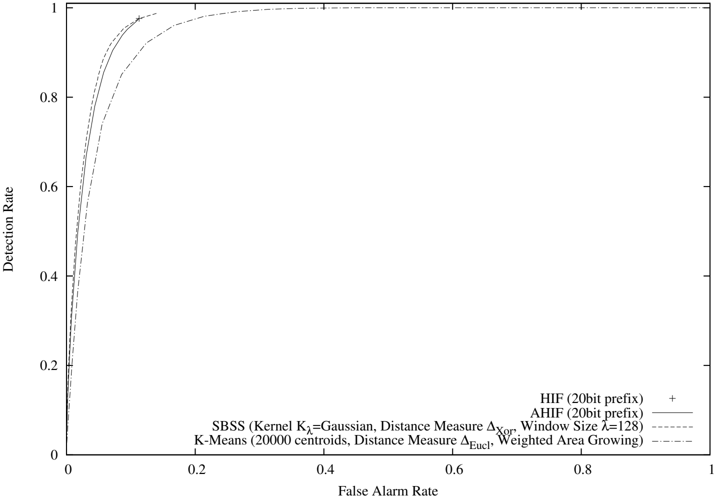
SBSS Results

Different Distance Measures



Method Comparison

All Methods



Evaluation

DDoS Attack Mitigation

- **Efficiency**: correctly denying illegal requests
  efficiency = 1 - false alarm rate
- **Collateral damage**: denying legal users
  collateral damage = 1 - detection rate

Policy

Choose efficiency as low as possible but as high as necessary for the server to serve requests in reasonable time. This minimizes collateral damage.

Evaluation

DDoS Attack Mitigation

	90.0	95.0	99.0
AHIF			
• 32bit prefixes	77.15	81.44	85.87
• 20bit prefixes	4.13	18.71	65.04
• 16bit prefixes	9.43	28.25	71.17
SBSS			
• window size $\lambda = 4$	23.51	24.81	61.68
• window size $\lambda = 32$	4.53	**14.88**	61.81
• window size $\lambda = 128$	**3.58**	**14.88**	**61.52**
k-means			
• 100 centroids	33.75	60.45	91.40
• 5000 centroids	16.45	37.34	80.52
• 20000 centroids	12.29	30.17	77.07

Conclusion

Distance Measure

- The different distance measures play a minor role (regardless of the method)

Method

- There is no uniform better method, selection depends on the application
- k-means works worse then SBSS, but usefull if very high detection rates are required

DDoS Mitigation

- SBSS works best
- AHIF also appealing if low computational effort is required

Thank you

Online Demo

SBSS Online Demo for creating DDoS Firewall rules

http://demo.iupr.org/ip-density

Thank you for your attention!

http://netsec.iupr.com

References I

Goldstein, M., Lampert, C., Reif, M., Stahl, A., and Breuel, T. (2008).
Bayes optimal ddos mitigation by adaptive history-based ip filtering.
pages 174–179, Los Alamitos, CA, USA. IEEE Computer Society.

Jung, J., Krishnamurthy, B., and Rabinovich, M. (2002).
Flash crowds and denial of service attacks: Characterization and implications for cdns and web sites.
In *Proceedings of the International World Wide Web Conference*, pages 252–262. IEEE.

References II

- Pack, G., Yoon, J., Collins, E., and Estan, C. (2006).
  On Filtering of DDoS Attacks Based on Source Address Prefixes.
  In *Proceedings of the 2nd International Conference on Security and Privacy in Communication Networks (SecureComm 2006)*. IEEE.

- Peng, T., Leckie, C., and Ramamohanarao, K. (2003).
  Protection from Distributed Denial of Service attack using history-based IP filtering.
  In *Proceedings of the IEEE International Conference on Communications (ICC 2003)*, Anchorage, AL, USA. IEEE.