# MailRank: Using Ranking for Spam Detection

Yitao Jiang

CS 595 Software Security

# MailRank

- Introduction
- Motivation
- Related work
- Technical details
- Evaluation
- Advantages and limitations

# Introduction

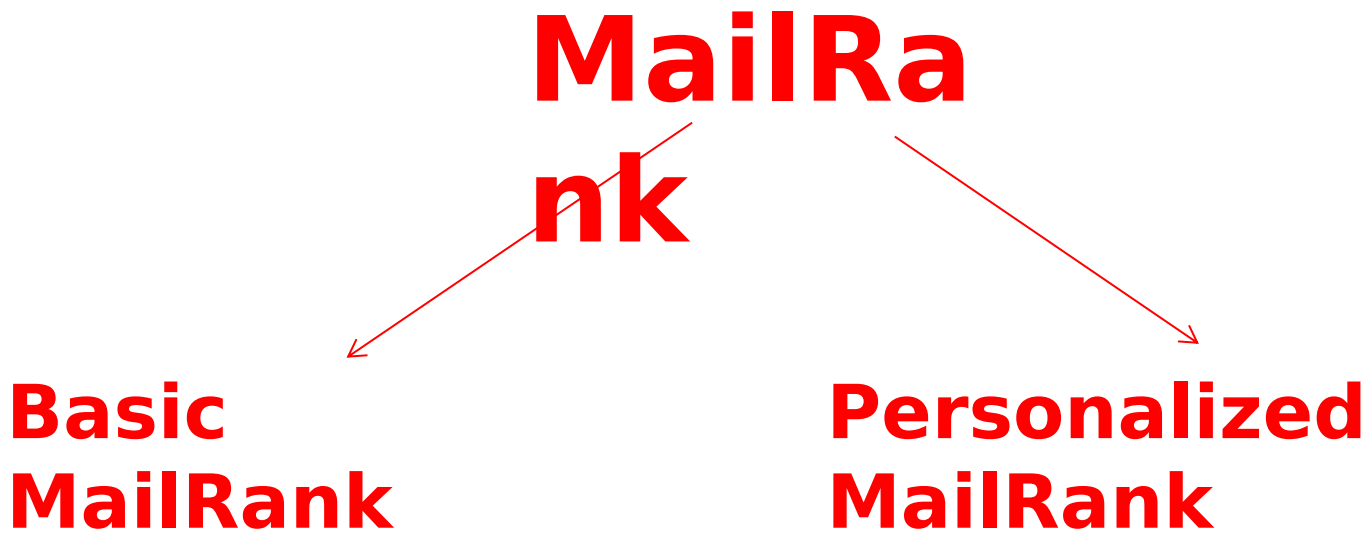
- Existing spam filters exhibit some problems:

- Maintenance
- Error rate
- Too many emails for some high-volume users

# Motivation

- Motivation: address all the problems above
- Social network formed by email communication can be used
  as a strong foundation for spam detection

# Motivation

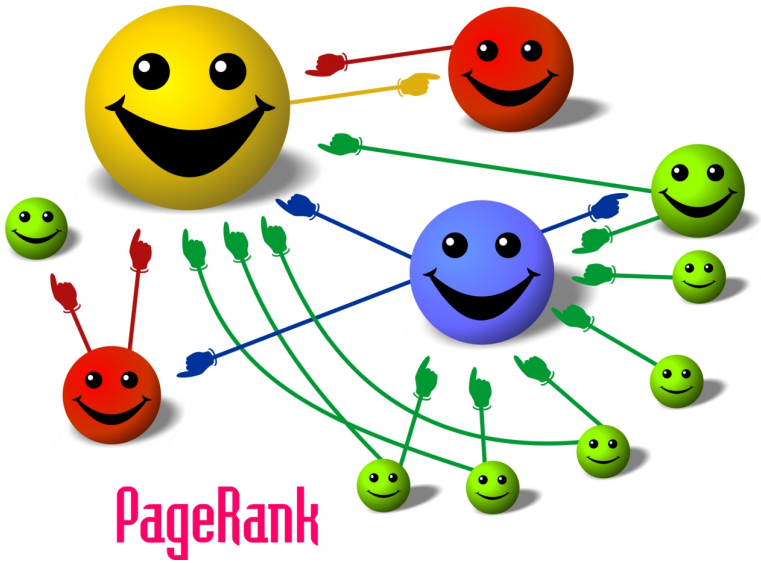


# Related work

- PageRank

a page has a high rank if the sum of ranks of its backlinks is high

$$PR(p) = c \cdot \sum_{q \in I(p)} \frac{PR(q)}{\|O(q)\|} + (1 - c) \cdot E(p)$$




- Personalized PageRank

Each user select her preferred pages. Then compute personalized rank vectors

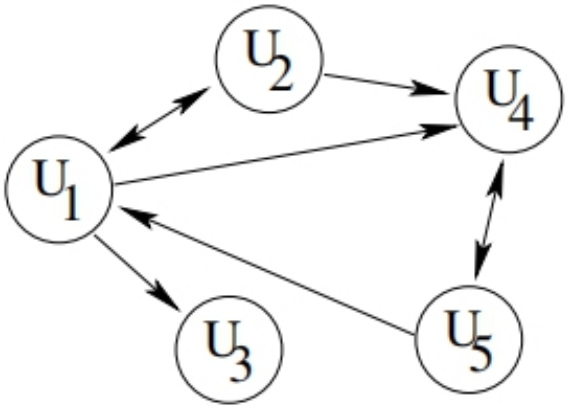
# Related work

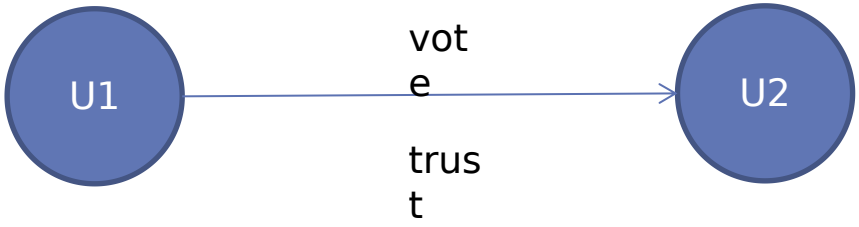
**PageRank** → **Basic MailRank**

**Personalized PageRank** → **Personalized MailRank**

# MailRank

- Build a graph




- If U1 has sent an email to U2, add edge<U1, U2> ,which implies U1 trusts U2

# Basic MailRank

- step1: determine the biased set
  - small set of users with high reputation
  - should not contain any spammer
- step2: apply power iteration algorithm

$$PR(p) = c \cdot \sum_{q \in I(p)} \frac{PR(q)}{\|O(q)\|} + (1 - c) \cdot E(p)$$

# Basic MailRank

power iteration algorithm:

**Algorithm 3.1**. The Basic MailRank Algorithm.

**Client Side**:
Each vote sent to the MailRank server comprises:
*Addr(u)* : The hashed version of the email address of the voter *u*.
*TrustVotes(u)* : Hashed version of all email addresses
*u* votes for (i.e., she has sent an email to)

**Server Side**:
**1**: Combine all received data into a global email network graph. Let
$T$ be the Markov chain transition probability matrix, computed as:
**ForEach** known email address $i$
**If** $i$ is a registered address, i.e., user $i$ has submitted her votes
**ForEach** trust vote from $i$ to $j$
$T_{ji} = 1/\text{NumOfVotes}(i)$
**Else ForEach** known address $j$
$T_{ji} = 1/N$, where $N$ is the number of known addresses.
**3**: Determine the biasing set $B$ (i.e., the most popular email addr.)
**3a**: Manual selection or
**3b**: Automatic selection or
**3c**: Semi-automatic selection
**4**: Let $T' = c \cdot T + (1 - c) \cdot E$, with $c = 0.85$ and
$E[i] = [\frac{1}{||B||}]_{N \times 1}$, if $i \in B$, or $E[i] = [0]_{N \times 1}$, otherwise
**5**: Initialize the vector of scores $\vec{x} = [1/N]_{N \times 1}$, and the error $\delta = \infty$
**6**: **While** $\delta < \epsilon$, $\epsilon$ being the precision threshold
$\vec{x}' = T' \cdot \vec{x}$
$\delta = ||\vec{x}' - \vec{x}||$
**7**: Output $\vec{x}'$, the global MailRank vector.
**8**: Classify each email address in the MailRank network into:
'spammer' / 'non-spammer' based on the threshold $T$

# Basic MailRank

Problem of basic MailRank:

too general with respect to user ranking

users want their acquaintances ranked higher than unknown users

Can we build a personalized ranking vector for every user?

# Personalized MailRank

Each user decide a preference set

compute partial vectors for all common users and hub skeleton for each user

combine them to compute PPV( personalized pagerank vector)

# Evaluation

build a power-law model for evaluation

Analysis on three issuse:

effectiveness in case of very sparse MailRank networks

exploitation of spam characteristics

attacks on MailRank

# Evaluation

Effectiveness in case of very sparse MailRank networks

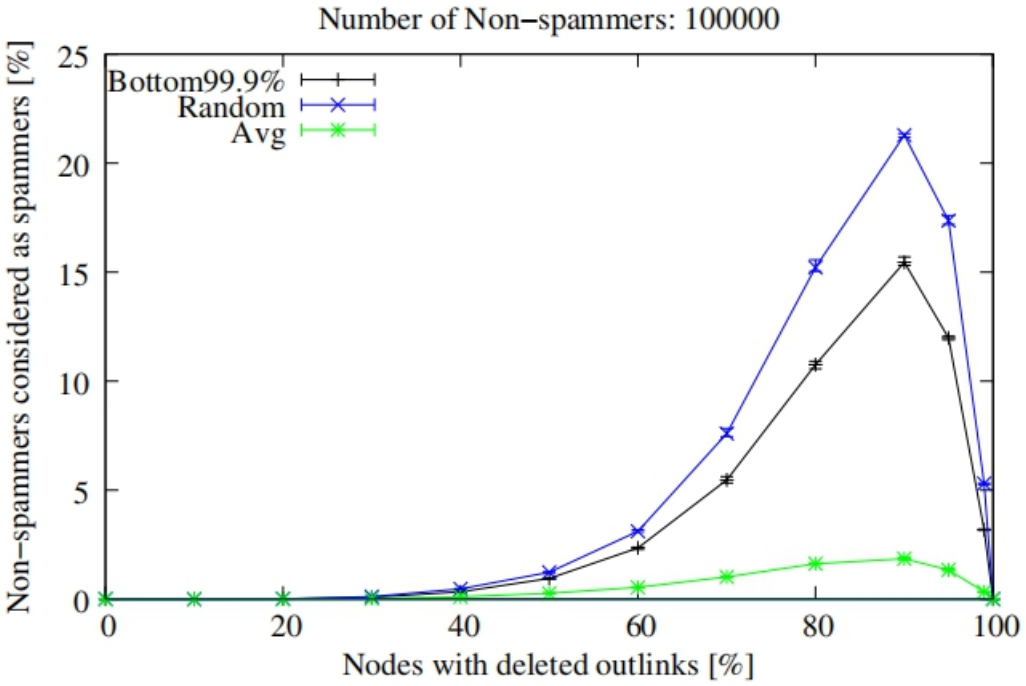



**Figure 3: Very sparse MailRank networks**

# Evaluation

Exploitation of spam characteristics

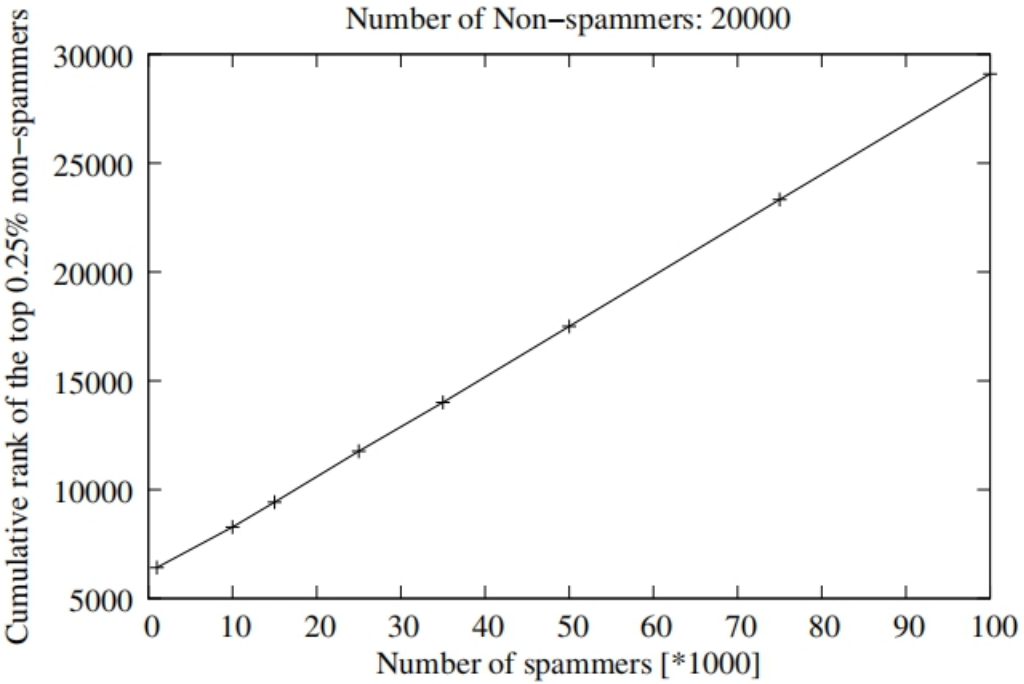



**Figure 4: Rank increase of non-spammer addresses**

# Evaluation

## Attacks on MailRank

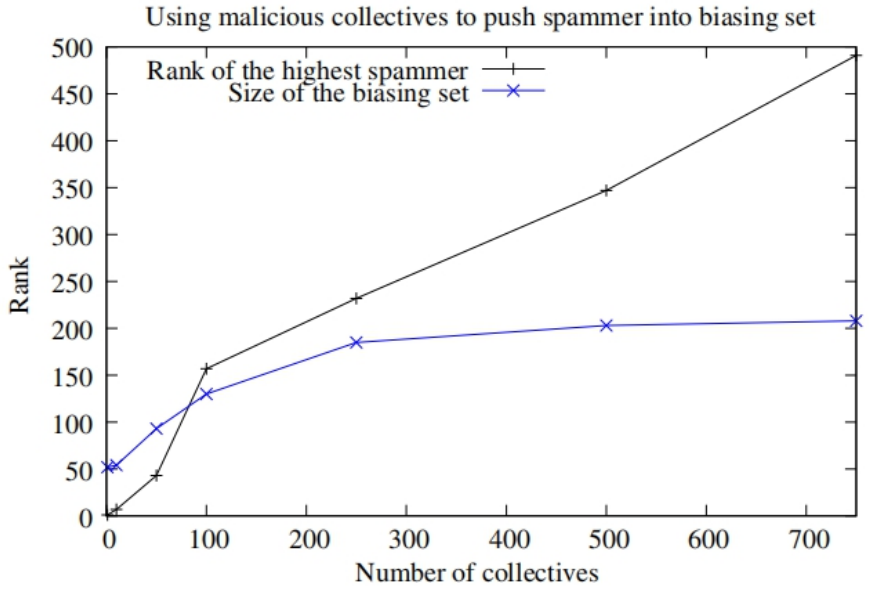



**Figure 5: Automatic creation of the biasing set**

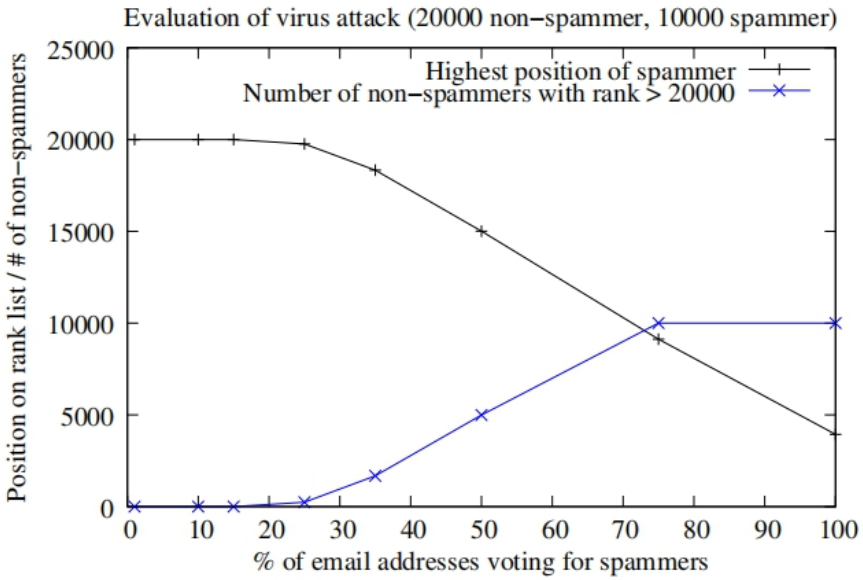



**Figure 6: Simulation results: Virus attack**

# Advantages and limitation

## Advantages:

Shorter individual cold-start phase

High attack resilience

Stable results

Partial participation

...

## Limits:

cannot prevent address spoofing attack

may misdetect some special non-spammer users

highly rely on the central server

# Thank you

Yitao Jiang

CS 595 Software Security