# PROFILING NETWORK TRAFFIC FOR INTERNAL SECURITY USING TOP-K RANKING SIMILARITY MEASURES

Alvaro I. Parres-Peredo, Hugo I. Piza-Davila, and Francisco Cervantes-Alvarez

PhDEngScITESO-16-17-R.docx

December 1, 2016
Tlaquepaque, Mexico 45604

Doctoral Program in Engineering Sciences
ITESO (*Instituto Tecnológico y de Estudios Superiores de Occidente*)

# PROFILING NETWORK TRAFFIC FOR INTERNAL SECURITY USING TOP-K RANKING SIMILARITY MEASURES

Alvaro I. Parres-Peredo, Hugo I. Piza-Davila, and Francisco Cervantes-Alvarez

December 1, 2016

Doctoral Program in Engineering Sciences
ITESO (*Instituto Tecnológico y de Estudios Superiores de Occidente*)

Tlaquepaque, Mexico 45604
Tel +52 33 3669 3598
E-mail: dci@iteso.mx

*Keywords* *computer network, network security, user profile, network traffic*

## Abstract

The detection of unknown internal network attacks is a challenge in network security field. Some authors analyze the overall network behavior using profiles to detect unknown network attacks. This work proposes a new methodology for analyzing user network profiles, built from the host traffic, using top-k ranking similarity measures and a 4-view network profile. It also presents an experiment where real-time traffic of different users is captured and compared to a reference traffic that corresponds to one of them.

## I. INTRODUCTION

Intrusion Detection Systems (IDS) are technologies used for computer network security, that have the goal of detecting attackers, by monitoring and analyzing the events on systems, computers, or networks. Currently, there exist two types of IDS: 1) signature-based, and 2) anomalies-based. The first one detects well-known attacks, whereas the second tries to detect new types of attacks [1].

Nowadays attackers use advanced techniques to be undetected by IDS, including the following: IP address spoof, encrypted payload, or even social engineering techniques [2]. A common symptom of an attack using these techniques is that the host under attack is having an unexpected network behavior. This is why the use of profiles to determine whether the user is having or not the expected behavior has become necessary as a new way to detect intrusions. Many

This work was supported in part by ITESO's Program for Academic Level Enhancement (*Programa de Superación del Nivel Académico*, PSNA) through an assistantship granted to A. I. Parres-Peredo.

authors [3]–[5] have been working on generating networks profiles, most of them work with the traffic at the border of the network.

The 4-view profile [6] is a network profile built with traffic captured at the host or a near point from it. This will help in future work to determine if a host is having or not the expected behavior, i.e., if the host is under attack or not.

In order to compare two profiles, we propose to treat each view of the 4-view profile as a top-*k* ranking, and to use a similarity measure for top-*k* rankings in order to compare them.

In this work, we propose a 3-phase methodology to calculate a similarity between real-time captured traffic –current behavior– and traffic captured a priori within a period of normal behavior –expected behavior– (see Section III).

In order to test the accuracy of this methodology, we build the normal-behavior profile of a user, and then calculate the similarity factor between this same user and two others.

The present document is organized as follows. Section II presents some works about user profiling and network traffic profiling. Section III explains with detail the proposed methodology. Section IV contains the experimental results. Finally, Section V concludes the report and presents some future work.

## II. RELATED WORK ON USERS PROFILE AND NETWORK SECURITY

The usage of network user profiles for representing the network behavior has been part of the research about computer network security.

Kihl et al. [7] present a work about traffic analysis and characterization of Internet users to help understanding the Internet usage and the demands on broadband access. They use a commercial tool for capturing and classifying traffic according to the Internet protocols and applications. This work concludes that the usage of Internet has changed from traditional WWW requests to a more complex use. Their results about Internet usage in 2010 indicate that most of this traffic comes from: sharing files protocols (74%), media streaming (7.6%), and web-traffic (5.5%). The traffic for this work was collected from a Swedish municipal FTTH network.

Sing et al. [3] present an intrusion detection technique using network-traffic profiling and an online sequential extreme machine-learning algorithm. The proposed methodology uses a profiling procedure, called alpha profiling, that creates profiles on the basis of protocol and service features; and a second profiling process, beta profiling, where the alpha profiles are grouped to reduce the number of profiles. The authors made three different experiments: 1) using all features and alpha profiling, 2) using only some features and alpha profiling, and 3) using only some features, alpha profiling and beta profiling. The best results were obtained from the last experiment using both profiling methods. The dataset used for this work was NSL-KDD.

A work that builds profiles of network prefixes instead of users is presented by Qin et al [4]. They propose aggregating traffic based on network prefixes in order to reduce the amount of data to be processed, and then calculate clusters using a $k$-means algorithm. Qin found that similar users produce similar traffic; with this information, decisions about security and management can be taken. The traffic used for this work was captured at CERNET backbone.

A similar work is presented by Xu et al. [5], who proposed a methodology that analyzes Internet traffic. This methodology first constructs bipartite graphs; after this, it generates one-mode projections; then, it builds a similarity matrix and generates cluster with a spectral clustering method; finally, it analyzes the clusters. The traffic used in this work was captured at the backbone of a large Internet service provider, aggregating the information using 24-bit length prefix networks, and the network 5-tuple.

As we can see, all the works discussed above have used the traffic captured at a far point from the end-user host, even outside the local network of the user, leaving unattended the internal network security. In addition, the usage of profiles has proved viable to either identify or specify network behaviors.

## III. METHODOLOGY PROPOSED

The proposed methodology has the goal of determining how similar the traffic captured in real time is to the traffic captured a priori in a controlled environment, i.e., the normal-behavior. This methodology uses the 4-view network profile [6] for grouping the captured traffic.

### *A. Build User Normal-Behavior.*

This phase builds a user network profile called normal-behavior, which will be used as reference for calculating the similarity factor. Fig. 1 shows the overall process at this phase.

Network traffic, $P_{SE,x}$, is captured from user $x$ during a period of time $T$ in a secure environment at which we can guarantee that the host will be used only by the expected user and there is no malware, virus, Trojan or any other malicious software installed. The period of time $T$

must be long enough to make sure that all the habitual tasks of the user are registered.

From $P_{SE,x}$ is selected a subset $p$ that corresponds to the period of time $[t\ \cdots t+f]$. With traffic $p$, the 4-view profile is built, and from each view *v*, a top-*k* ranking is calculated and stored in $\mathrm{K}_{v,x}$. This process is repeated while *t* is less than *T*. At the end of each iteration, *t* increments in $\Delta t$, a value smaller than *f* in order to produce overlaps in timeframes.

An extraction of a top-$k$ ranking from the IP view is illustrated in Fig. 2, where $k=10$ and five time frames are listed.

### B. *Capture Regular-Traffic*

In this phase, regular-traffic, $P_{rt,x}$ is captured in real time from the user host during a period of time $[t\ \cdots t+f]$, and for each $\Delta t$ increment. Using this traffic, the 4-view profile is built and, for each view *v*, a top-*k* ranking is calculated. Using each of these top-*k* rankings, the best similarity factor against every top-*k* ranking in $\mathrm{K}_{v,x}$ is found, as it is described in the next section. Fig. 3 shows the flow diagram of this process.

### C. *Calculate Best Similarity Factor.*

The purpose of this phase is to find out if the real-time traffic captured during a period of time $[t \cdots t+f]$ resembles to any traffic within the records of the normal behavior captured during a period of time of length *f*. Thus, a similarity factor is calculated between the top-*k* ranking of the real-time traffic ($\kappa_{v,x\prime}$) and each of the top-*k* rankings stored in $\mathrm{K}_{v,x}$. The best similarity factor is found for each view. According to this value, a decision might be taken about the correspondence of the current network traffic with the expected one. Fig. 4 shows a diagram of this process.

To compare each pair of top-*k*, it is necessary to use a ranking similarity measure, but with the peculiarity that these top-*k* rankings are non-conjoint rankings, this means that not all elements of one of them are present in the other. Thus, we have explored two different measures:

a) Spearman's rho [8]

Spearman's rho is the distance between two permutations, formally:

$$\rho(\sigma_1,\sigma_2)=\left(\sum_{i=1}^{n}|\sigma_1(i)-\sigma_2(i)\ |^2\right)^{1/2} \qquad (1)$$

In the case of non-conjoint rankings, we use the Fagin approach that considers that both top-*k* lists are a bijection from a domain *D*, that contains all the elements. It can be assumed that every non-present element in top-*k* list is ranked in some position after *k*th [8].

Based on this, we define that the distance for a non-present element is *k*. The new formula is:

$$\rho(X,Y) = \left(\sum_{i=1}^{k} |\sigma_X(i) - \sigma_Y(i)|^2\right)^{1/2} \tag{2}$$

where *X* and *Y* are top-*k* lists from the same domain, $\sigma_K(i)$ is the rank of element *i* in a top-*k* list; in case element *i* is not present at this list, the value of $\sigma_K(i)$ is *k*, where *k* is the number of elements of both lists.

Finally, in order to have similar results to the second measure, we re-write the formula to get a normalized value between 0 and 1, where 0 denotes identical top-*k* lists, and 1 denotes totally different top-*k* lists. Formally:

$$\rho(X,Y) = \frac{1}{k^3}\left(\sum_{i=1}^{k} |\sigma_X(i) - \sigma_Y(i)|^2\right) \tag{3}$$

b) Average Overlap

The second measure is based on the work by Webber [9] called average overlap. This measure calculates for each $d \in \{1 \ldots k\}$, the overlap at *d*, and then averages those overlaps to derive the similarity measure.

Formally, the average overlap between two top-k lists can be expressed as:

$$AO(S,T,k) = \frac{1}{k}\sum_{d=1}^{k} A_{S,T,d} \tag{4}$$

where *S* and *T* are top lists of *k* number of elements, and $A_{S,T,d}$ is defined as:

$$A_{S,T,d} = \frac{|S_{:d} \cap T_{:d}|}{d} \tag{5}$$

where $S_{:d}$ and $T_{:d}$ represents each of the rankings with only $d$ top elements. Fig. 5 shows an example where the AO of two top-5 lists is calculated.

Finally, in order to have similar results to the first measure, we calculate the complement, where 0 denotes identical top-*k* lists, and 1 denotes totally different top-*k* lists. Formally:

$$AO(S,T,k) = 1 - \frac{1}{k}\sum_{d=1}^{k} A_{S,T,d} \tag{6}$$

## IV. EXPERIMENTS AND RESULTS

### *A. Experiment Setup*

The experiment was done at a campus area network that has a 16-bit network; it has a Windows domain controller and uses an HTTP proxy. The campus applications include web-apps and remote desktop apps. The email service is provided by Microsoft Exchange Server, which is hosted outside of the campus network.

The target users are full-time professors, who have a computer with two types of access to the network: a wired access with a static IP address and a wireless access with a dynamic IP address.

For this experiment, one full-time professor was selected as User A, for whom we have generated his normal-behavior traffic $\mathrm{K}_A$ and then captured real-time traffic $\kappa_A$. With the goal of validating the methodology proposed, two other professors from the same academic department were selected as Users B and C, and generated real-time traffic $\kappa_B$ and $\kappa_C$.

Different values of the parameters were evaluated: $f$ = {1, 5, 10 minutes}, $\Delta t$ = {10, 30, 60 seconds}, the number of elements selected for the top-*k* rankings, *k* = {10, 25, 50, 100}, and both measure functions.

The best combination <*t*, $\Delta t$, *k*, *Function*> was: *t = 5min*, $\Delta t$ *= 10sec, k=10, Average Overlap*. It maximizes the differences between $AO(\mathrm{K}_A, \kappa_{A'})$ and both $AO(\mathrm{K}_A, \kappa_B)$ and $AO(\mathrm{K}_A, \kappa_C)$.

### *B. Capturing and Processing Traffic*

During 4 labor-days, traffic was captured to characterize normal-behavior from User A's computer. Before start capturing we made a check-up to validate that the computer has not installed any malicious software. During all this period only this user had access to the computer. The size of the traffic captured was 2.56 gigabytes, involving more than four millions of packets. All the packets were processed as describe in Section III.A.

On different days, 8-hour real-time traffic was captured from the computers of users A, B and C ($\kappa_{A'}, \kappa_B, \kappa_C$). This traffic was processed as described in Section III.B.

### *C. Similarity Calculation and Results*

For each top-$k$, $\kappa_{A'}$, $\kappa_B$ and $\kappa_C$ the best similarity factor with respect to $A$ was found and plotted; see Fig. 6 (IP view) and 9 (service view). We can see that User A exhibited a higher similarity to his own normal behavior than user B and C, as expected. Fig. 8 (IP view) shows the similarity factors as a histogram.

Since top-$k$ rankings are independent from each other, we can order the similarity factors by value to get a clearer graph. Fig. 7 (IP view) and 10 (service view) show all the similarity factors order by value.

HTTP Host view presented poor similarity factors for all the users, so this view could not allow us to differentiate user A from users B and C (see Fig. 11); similar results occur with HTTP URL view.

## V. CONCLUSIONS AND FUTURE WORK

In this work, we have proposed a methodology to determine how similar is the traffic captured in real-time at a user host $\kappa_{v,x'}$, to a previously captured traffic that we called normal-behavior$\mathrm{K}_{v,x}$, at the same host or another. Also, we presented the results of the implementation of such methodology. The similarity was calculated for each view from the 4-view profile.

A preliminary early conclusion from this work is that the IP and Service views allow to determine if the captured traffic corresponds to the expected user. In the presented graphs we can see how the User A has better similarity factors than users B and C. The HTTP Host and HTTP URL views did not present the same behavior than IP and Services view, in many time-frames the similarity factor was better for User B and C than for User A. This might be due to the low volume of HTTP traffic, most of the web traffic is with HTTPS protocol that does not allow the capture process read the request host and URL.

From the results we can see that non top-k ranking from $k_{v,x}$ is identical to any top-$k$ from $K_{v,x}$ , i.e. the similarity factor is 0.0.  A possible reason for this is that multiple IP addresses can be configured by the same host or service, or the user really does not do exactly the same all the time. But we consider that the similarity factors obtained are good enough to differentiate between the expected user from the others.

In future work, the similarity factors obtained will be employed by a real-time process that determines how likely is the current traffic to belong (or not) to the expected user. More experiments are still necessary to determine the consistency of the proposed methodology.

## REFERENCES

[1] A. I Parres-Peredo, H. I. Piza-Davila, and F. Cervantes-Alvarez, “Challenges and opportunities in computer network security,” Internal Report *PhDEngScITESO-15-20-R*, ITESO, Tlaquepaque, Mexico, Dec. 2015.

[2] P. Wood, Ed., *ISTR Internet Security Threat Report*, Symantec, Apr. 2016.

[3] R. Singh, H. Kumar, and R. K. Singla, “An intrusion detection system using network traffic profiling and online sequential extreme learning machine,” *Expert Syst. Appl.*, vol. 42, no. 22, pp. 8609–8624, Dec. 2015.

[4] T. Qin, X. Guan, C. Wang, and Z. Liu, “MUCM: multilevel user cluster mining based on behavior profiles for network monitoring,” *IEEE Syst. J.*, vol. 9, no. 4, pp. 1322–1333, Dec. 2015.

[5] K. Xu, F. Wang, and L. Gu, “Behavior analysis of internet traffic via bipartite graphs and one-mode projections,” *IEEEACM Trans. Netw.*, vol. 22, no. 3, pp. 931–942, Jun. 2014.

[6] A. I. Parres-Peredo, H. I. Piza-Davila, and F. Cervantes-Alvarez, “A novel user network profile based on host network traffic,” Internal Report *PhDEngScITESO-16-08-R*, ITESO, Tlaquepaque, Mexico, Jul. 2016..

[7] M. Kihl, P. Ödling, C. Lagerstedt, and A. Aurelius, “Traffic analysis and characterization of Internet user behavior,” in *2010 International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)*, Moscow, 2010, pp. 224–231.

[8] R. Fagin, R. Kumar, and D. Sivakumar, “Comparing top k lists,” *SIAM J. Discrete Math.*, vol. 17, no. 1, pp. 134–160, Jan. 2003.

[9] W. Webber, A. Moffat, and J. Zobel, “A similarity measure for indefinite rankings,” *ACM Trans Inf Syst*, vol. 28, no. 4, p. 20:1–20:38, Nov. 2010.

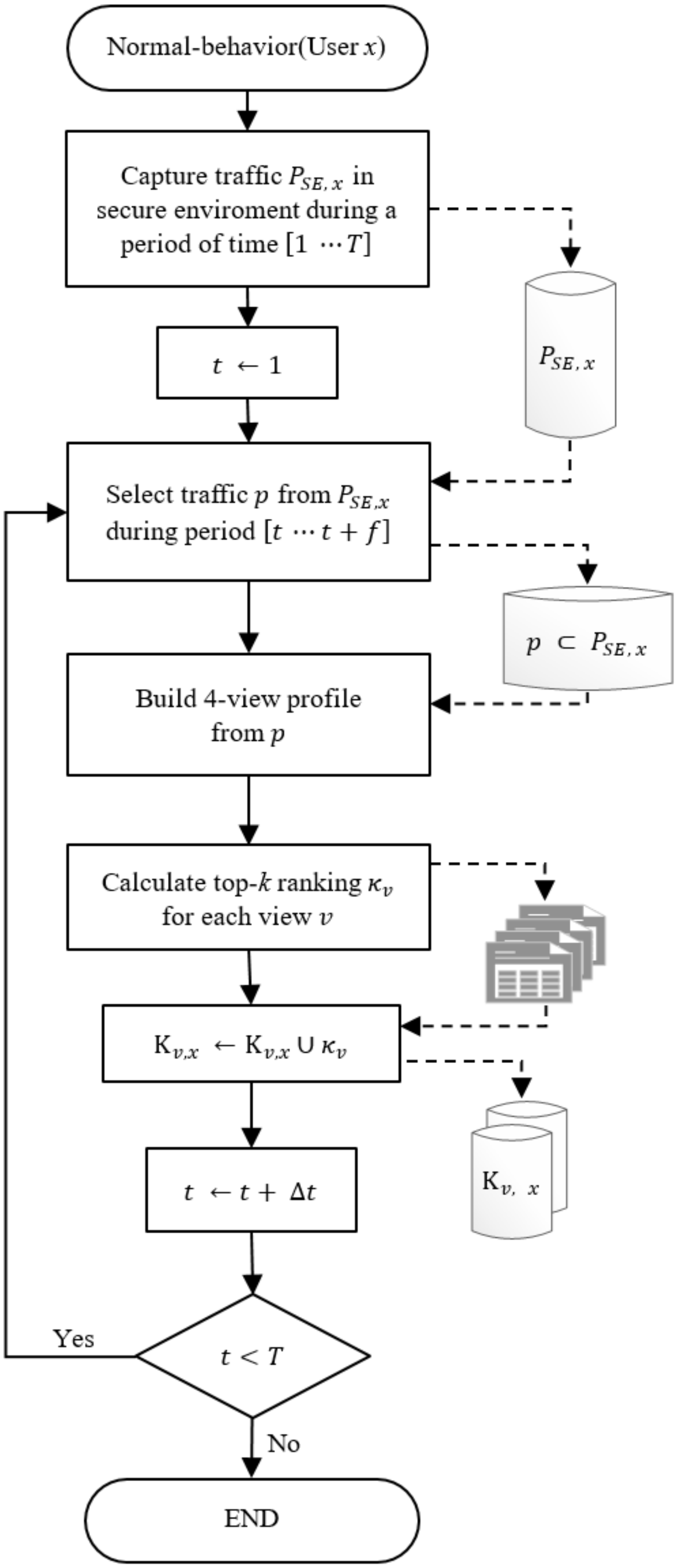

Normal-behavior(User $x$)
Capture traffic $P_{SE,x}$ in secure enviroment during a period of time $[1 \cdots T]$
$t \leftarrow 1$
$P_{SE,x}$
Select traffic $p$ from $P_{SE,x}$ during period $[t \cdots t+f]$
$p \subset P_{SE,x}$
Build 4-view profile from $p$
Calculate top-$k$ ranking $\kappa_v$ for each view $v$
$\mathrm{K}_{v,x} \leftarrow \mathrm{K}_{v,x} \cup \kappa_v$
$\mathrm{K}_{v,x}$
$t \leftarrow t + \Delta t$
$t < T$
Yes
No
END

Fig. 1. Building the normal-behavior profile of user *x* during a period of time *T*.

| Timeframe | Top 1 | Top 2 | ⋯ | Top 10 |
|---|---|---|---|---|
| $[t\ \cdots t+f\ ]$ | 148.201.129.173 | 148.201.140.219 | ⋯ | 148.201.140.50 |
| $[t+\ \Delta t\ \cdots t+\Delta t+f\ ]$ | 148.201.129.173 | 148.201.140.219 | ⋯ | 148.201.140.50 |
| $[t+\ 2\Delta t\ \cdots t+2\Delta t+f\ ]$ | 148.201.129.173 | 148.201.140.148 | ⋯ | 148.201.140.98 |
| $[t+\ 3\Delta t\ \cdots t+3\Delta t+f\ ]$ | 132.245.44.22 | 148.201.140.148 | ⋯ | 148.201.129.43 |
| $[t+\ 4\Delta t\ \cdots t+4\Delta t+f\ ]$ | 148.201.140.148 | 148.201.140.219 | ⋯ | 148.201.129.43 |

Fig. 2. Example of a Top-*k* ranking from IP View.

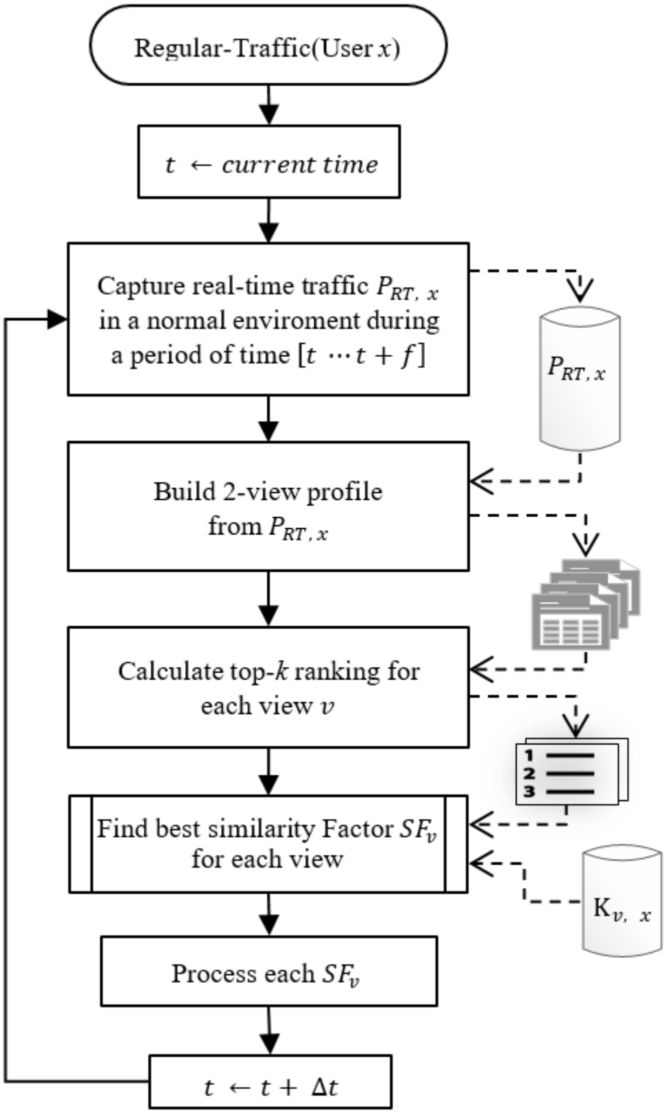

Regular-Traffic(User $x$)
$t \leftarrow current\ time$
Capture real-time traffic $P_{RT,\ x}$ in a normal enviroment during a period of time $[t\ \cdots t+f]$
$P_{RT,\ x}$
Build 2-view profile from $P_{RT,\ x}$
Calculate top-$k$ ranking for each view $v$
1
2
3
Find best similarity Factor $SF_v$ for each view
$\mathrm{K}_{v,\ x}$
Process each $SF_v$
$t\ \leftarrow t+\ \Delta t$

Fig. 3. Process of capture real-time traffic from user *x*.

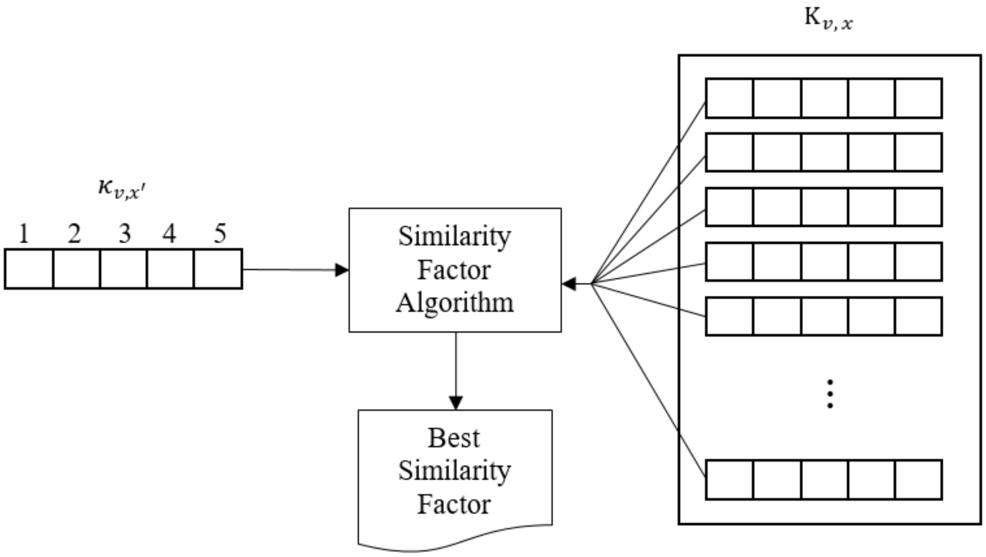


Fig. 4. Calculating best similarity factor.

| *d* | $S_{:d}$ | $T_{:d}$ | $A_{S,T,d}$ | AO(*S*,*T*,*d*) |
|---|---|---|---|---|
| 1 | <a> | <x> | 0.0000 | 0.0000 |
| 2 | <ab> | <xc> | 0.0000 | 0.0000 |
| 3 | <abc> | <xcb> | 0.6667 | 0.2222 |
| 4 | <abcd> | <xcby> | 0.5000 | 0.2917 |
| 5 | <abcde> | <xcbye> | 0.6000 | 0.3534 |

Fig. 5. Similarity calculation of two top-5 lists using average overlap measure.

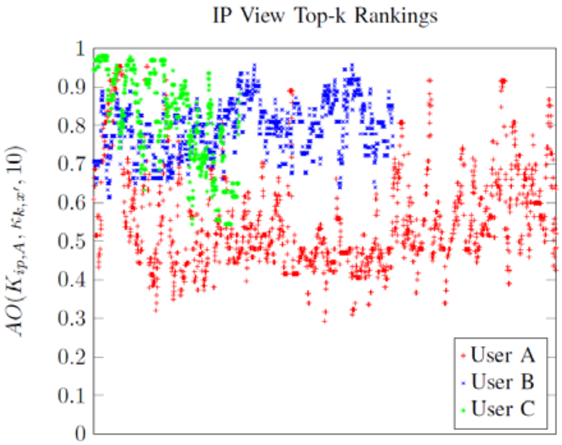


Fig. 6. Similarity Factor of IP View between $\kappa_{A'}$, $\kappa_B$ and $\kappa_C$ against $K_A$

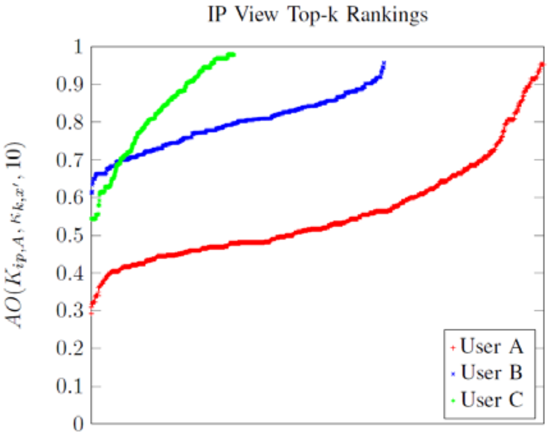


Fig. 7. Similarity factor of IP View between $\kappa_{A\prime}$, $\kappa_B$ and $\kappa_C$ against $K_A$ ordered by value.

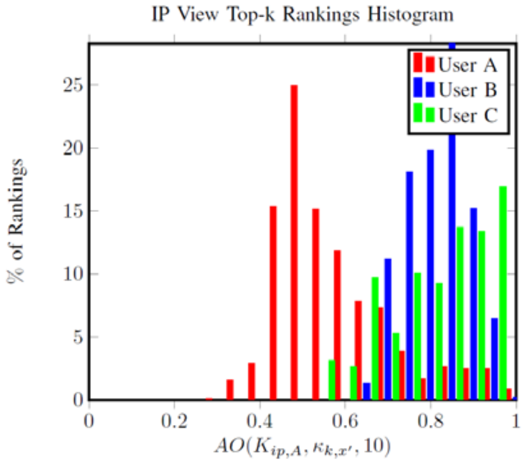


Fig. 8. Histogram of Similarity Factors from IP View between $\kappa_{A\prime}$, $\kappa_B$ and $\kappa_C$ against $K_A$.

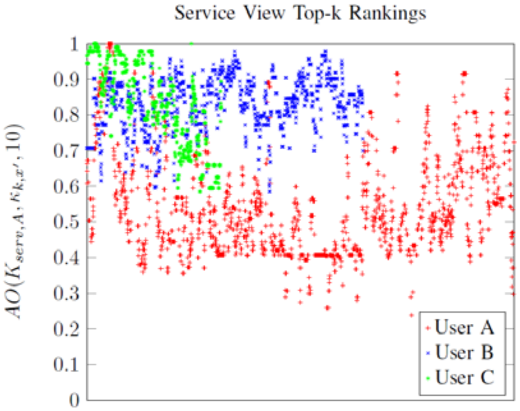


Fig. 9. Similarity Factor of Service View between $\kappa_{A'}$, $\kappa_B$ and $\kappa_C$ against $K_A$.

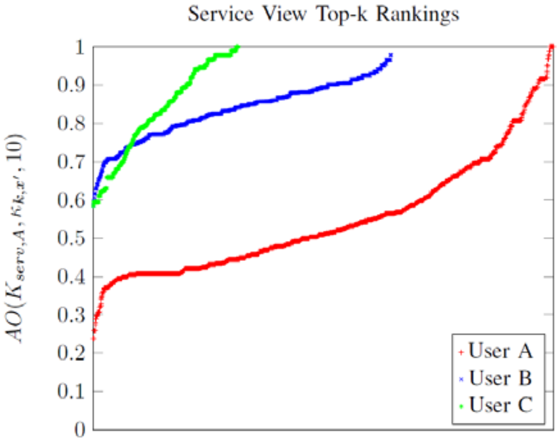
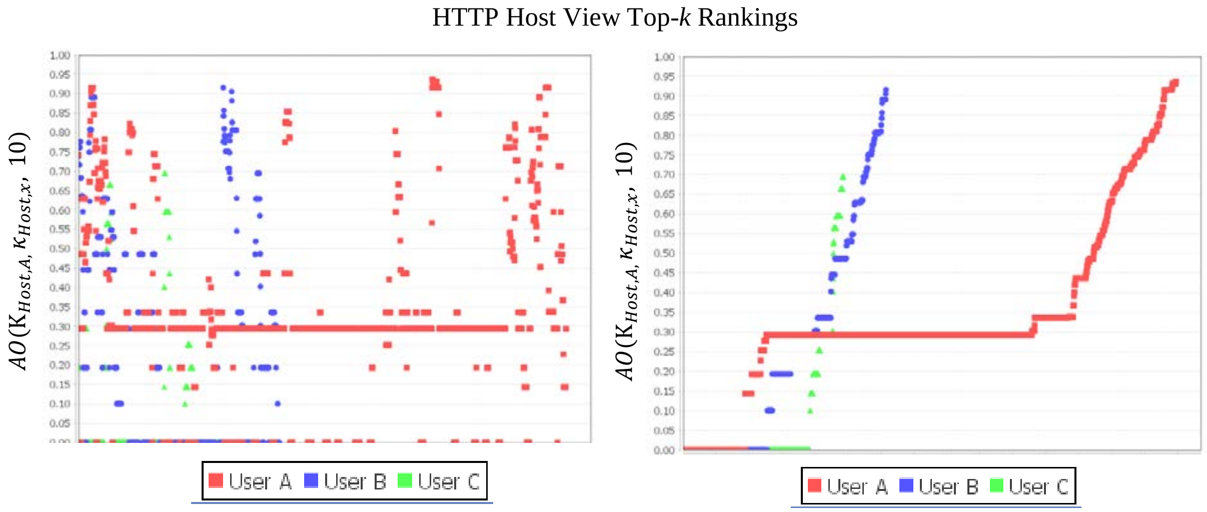


Fig. 10. Similarity factor of Service View between $\kappa_{A'}$, $\kappa_B$ and $\kappa_C$ against $K_A$ ordered by value.



a) b)

Fig 11. Similarity Factor of HTTP Host View between $\kappa_{A'}$, $\kappa_B$ and $\kappa_C$ against $\mathrm{K}_A$ ordered by: a) capture timestamp; b) by value.