Can't you hear me knocking

Identification of user actions on Android apps via traffic analysis

Candidate:
Riccardo Spolaor

Supervisor: **Prof. Mauro Conti**
Co-Supervisor: **Dr. Nino V. Verde**

April 17, 2014

UNIVERSITÀ DEGLI STUDI DI PADOVA

Outline

Motivation

Attack regarding **user privacy** on smartphones.

Can we be able to recognize **user actions** on his smartphone analyzing the generated **network traffic**?

Contribution

- We prove that this is possible, with an accuracy $> 95\%$.
- Traffic analysis using **machine learning** techniques.

Who do you sync you are? [Stöber et al., 2013]

Recognition of the **set of apps** installed on a smartphone, analyzing **traffic bursts** produced by apps in background.

Network profiler [Tongaonkar et al., 2013]

Building a **profile** for an apps.

Payload analysis on **HTTP** request and response.

Recognize an app from its network traffic.

Observed protocols

Protocol	Presence (packets)
TLSv1	95.4% (54038)
TCP	03.4% (1908)
HTTP	01.1% (644)
SSLv2	00.1% (63)

Table : Presence of different protocols among packets captured.

Payload analysis is not feasible because more than 95% of captured packets are **encrypted with TLS protocol**.

Key concepts

Interactions

Input on a device

E.g., tap, swipe, key press

⇩ used to achieve

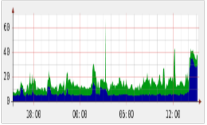
User actions

Operation on apps

E.g., send an email, open a page

⇩ produce

Network flows



Sequence of packets

Couple of IP addresses and ports

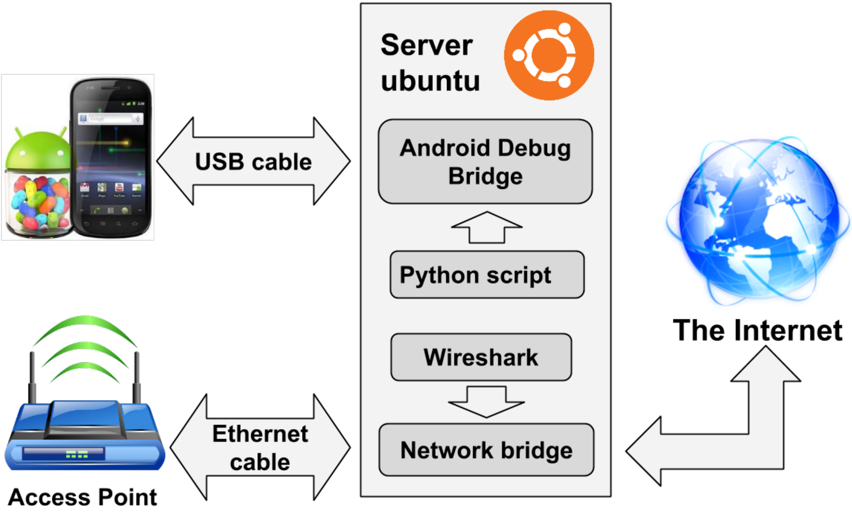
Framework overview

1. Data acquisition.
2. Flow building.
3. Flow clustering.
4. User action classification.

We test our solution on **Twitter**, **Facebook** and **Gmail**.

Data acquisition - Hardware environment

Università degli Studi di Padova



Data acquisition - Python scripts

- **Interactions** simulated issuing **Android Debug Bridge** commands.
- A script defines a **sequence of user actions** to explore the app functionality.
- Log files with a **timestamps** for every user action execution.
- Wireshark captures and log files are **synchronized** using ntp server.

Flow building

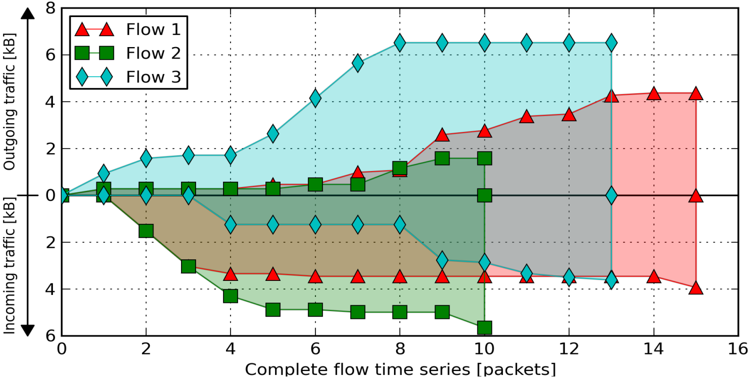
Packets filtering

- **Domain** (WHOIS protocol).
- **Connection control** (three-way handshake and ACK).

Flow conditions

- Consecutive packets with the same IP address destination ad port.
- Terminated if not seen any new packets since 4.5 seconds [Stöber et al., 2013].

Flow building - Flow representation



Flow ID	Flow time series
Flow 1	[282, -1514, -1514, -315, 188, -113, 514, 96, 1514, 179, 603, 98, 801, 98, -477]
Flow 2	[282, -1514, -1514, -1266, -582, 188, -113, 692, 423, -661]
Flow 3	[926, 655, 136, -1245, 913, 1514, 1514, 863, -1514, -107, -465, -172, -111]

Flow clustering

Università degli Studi di Padova

Motivation

A single user action can produce **multiple flows**.

Regroup flows in **clusters** by their similarities.

Missing knowledge about their **number**, nor features.

Implementation

We use **hierarchical agglomerative clustering** based on flows distances.

We evaluate the distance between two **flows time series** using **Dynamic Time Warping** [Müller, 2007].

Starting from hierarchy, we produce **sets of clusters** using thresholds.

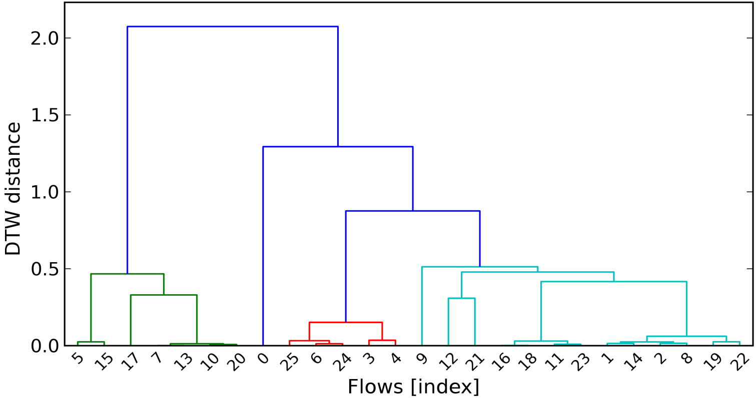


Figure : Dendrogram for an hierarchy of clusters

Flow clustering - Leaders election

Election of a flow f_i as **leader**, for each cluster $C = \{f_1, \ldots, f_n\}$

$$leader_C = \arg\min_{f_i \in C} \left(\sum_{j=1}^{n} dist(f_i, f_j) \right).$$

We use leaders to assign **unseen flows** to clusters.

User action classification

Dataset organization:

- **Classes** → User actions.
- **Features** → Flow presence in a cluster.

ID	User actions	Cluster_1	...	Cluster_k	...	Cluster_n
001	send mail	0	...	2	...	0
002	send reply	1	...	1	...	0
...	...	...	...	...	...	...

The classifier we use is **Random forest**.

Datasets

We collected data from **220 script executions** for each app.

Different **user actions** examples:

- 11660 for Gmail,
- 6600 for Twitter,
- 10120 for Facebook.

Divided in:

- Training set $\rightarrow$ 70%.
- Test set $\rightarrow$ 30%.

Produced with **disjoint accounts** sets.

Significant user actions

We consider actions significant for the **user privacy**.

- **Facebook** → Post a status, send a message.
- **Gmail** → Send an email, reply to an email.
- **Twitter** → Publish a tweet, open contacts page.

We regroup less significant actions in the class "**other**".

In "other", we also include the **background traffic** produced by an app.

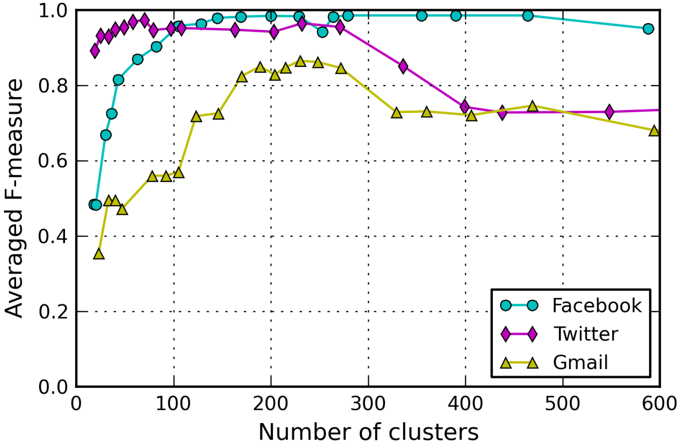
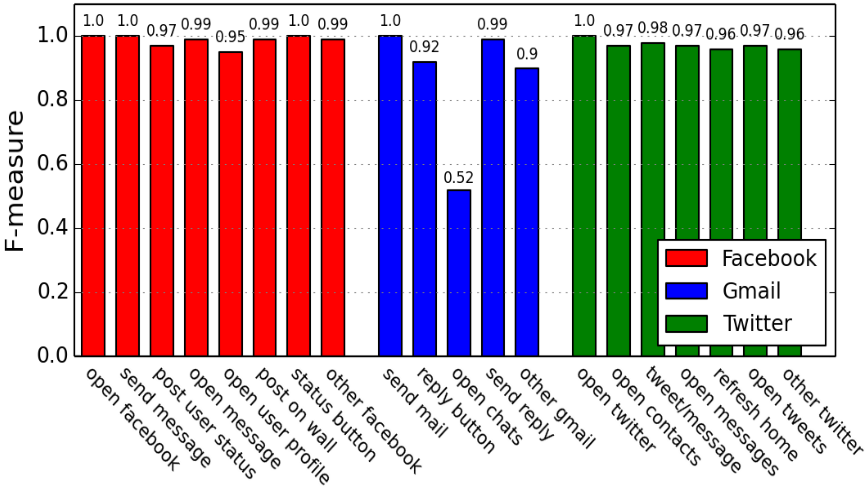


Figure : Accuracy obtained with Random forest classifier.

Performance



Conclusions

- We proposed a framework for the identification of **user actions** on Android apps.
- An eavesdropper can leverage this tool to **undermine the privacy** of mobile users.
- **Learn habits** of the target users, in order to gain some **commercial** or **intelligence** advantage.
- We believe that this work will **stimulate researchers** to work on countermeasures on the possible attacks.

[ACM WiSec 2014, 7th ACM Conference on Security and Privacy in Wireless and Mobile Networks (submitted)]

Thanks for your attention.

Do you have any questions?

References I

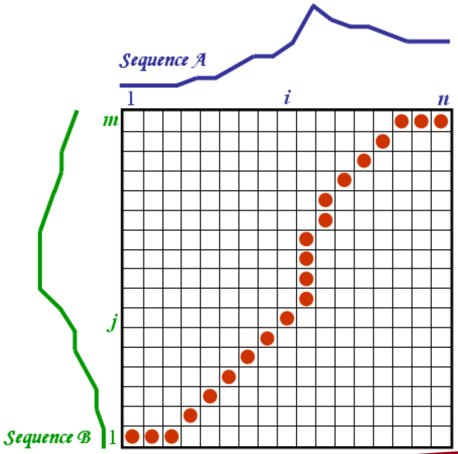
Müller, M. (2007).
Information Retrieval for Music and Motion.
Springer-Verlag New York, Inc.

Stöber, T., Frank, M., Schmitt, J., and Martinovic, I. (2013).
Who do you sync you are?: Smartphone fingerprinting via application behaviour.
WiSec '13.

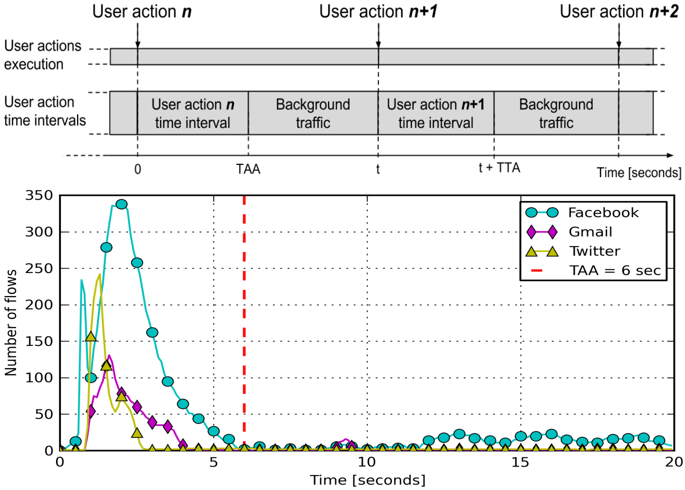
Tongaonkar, A., Dai, S., Nucci, A., and Song, D. (2013).
Understanding mobile app usage patterns using in-app advertisements.
PAM '13.

Dynamic Time Warping

Dynamic Time Warping (DTW) is an alignment method for time series [Müller, 2007], also used in speech recognition.



User action time interval



Clustering parameters

Apps	Sets	Weights	Incoming	Outgoing	Complete
Gmail	Conf. 1	0.80	[1,4]	[1,2]	[1,6]
		0.20	[1,6]	[1,3]	[1,9]
	Conf. 2	0.66	[1,4]	[1,2]	[1,6]
		0.33	[1,6]	[1,3]	[1,9]
	Conf. 3	0.33	[1,4]	[1,2]	[1,6]
		0.66	[1,6]	[1,3]	[1,9]
Facebook	Conf. 1	0.66	[1,3]	[1,5]	[1,7]
		0.33	[1,6]	[1,7]	[1,12]
	Conf. 2	0.33	[1,3]	[1,5]	[1,7]
		0.66	[1,6]	[1,7]	[1,12]
	Conf. 3	0.20	[1,3]	[1,5]	[1,7]
		0.80	[1,6]	[1,7]	[1,12]
Twitter	Conf. 1	0.95	-	-	[7,10]
		0.05	-	-	[1,10]
	Conf. 2	0.95	-	-	[8,11]
		0.05	-	-	[1,11]
	Conf. 3	0.95	-	-	[8,10]
		0.05	-	-	[1,10]

Table : Weights set configurations and packets intervals for Gmail, Facebook and Twitter apps.

Gmail flows

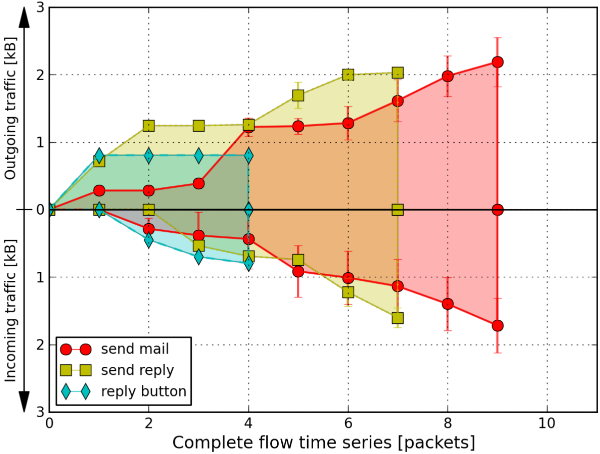


Figure : Flow representation for Gmail actions.

Twitter flows

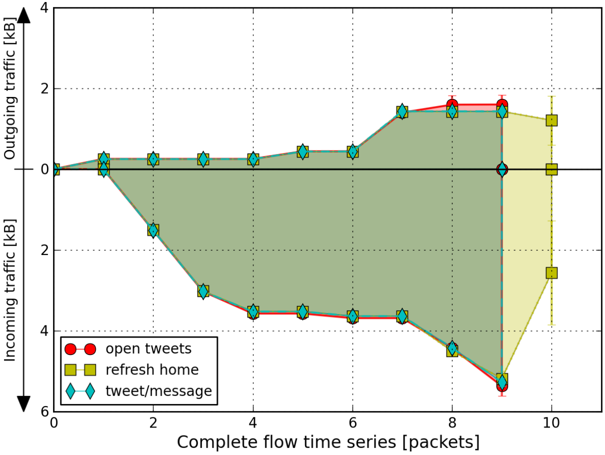


Figure : Flow representation for Twitter actions.

Significant user actions

Facebook

send message	send a direct message to a friend
post on wall	post a content on a friend's wall
post user status	post a status on user's wall
open user profile	select user profile page from menu
open message	select a conversation on messages page
status button	select "write a post" on user's wall
open facebook	Facebook app execution start

Gmail

send mail	send a new email
send reply	send a reply to an email
reply button	button to reply selection
open chats	select chats page from menu

Twitter

tweet/message	publish a tweet or send a message
refresh home	request for refresh the home page
open contact	select contacts page on menu
open messages	select direct messages page
open tweets	select tweets page on menu
open twitter	Twitter app execution start

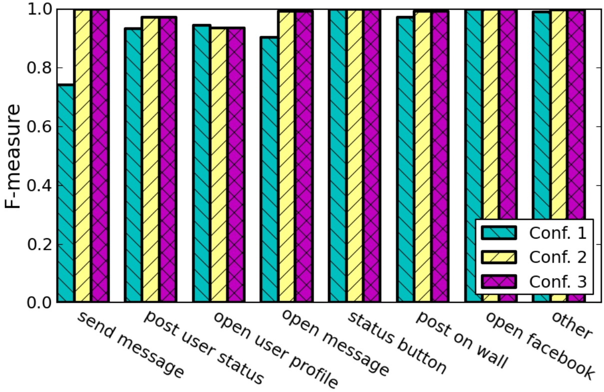


Figure : Classification accuracy of the Facebook user actions.

Facebook II

User actions	Precision	Recall	F-measure
send message	1.00	1.00	1.00
post user status	1.00	0.95	0.97
open user profile	0.96	0.91	0.94
open message	0.98	1.00	0.99
status button	1.00	1.00	1.00
post on wall	1.00	0.98	0.99
open facebook	1.00	1.00	1.00
other	0.99	1.00	0.99
Average	0.99	0.98	0.99

Table : Classification results of Facebook user actions reached using Configuration 3.

Facebook III

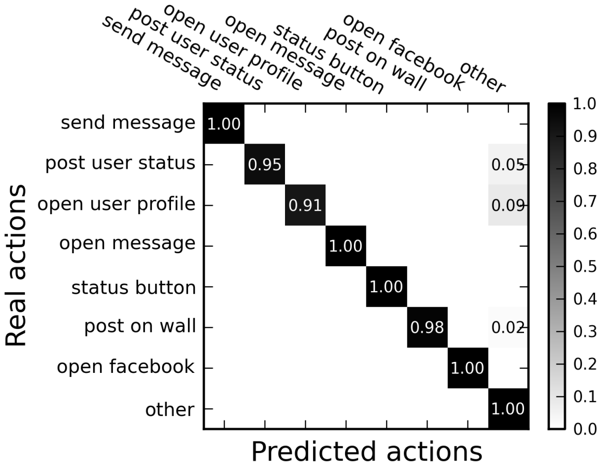


Figure : Facebook user actions confusion matrix for Configuration 3.

Gmail I

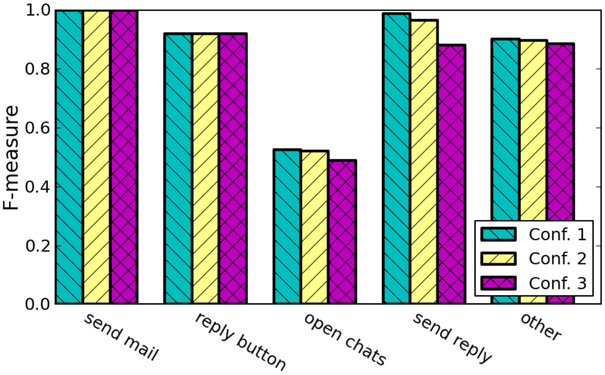


Figure : Classification accuracy of the Gmail user actions.

Gmail II

User actions	**Precision**	**Recall**	**F-measure**
send mail	1.00	1.00	1.00
reply button	0.85	1.00	0.92
open chats	0.36	0.94	0.52
send reply	0.98	1.00	0.99
other	0.99	0.82	0.90
Average	0.83	0.85	0.86

Table : Classification results of Gmail user actions reached using Configuration 1.

Gmail III

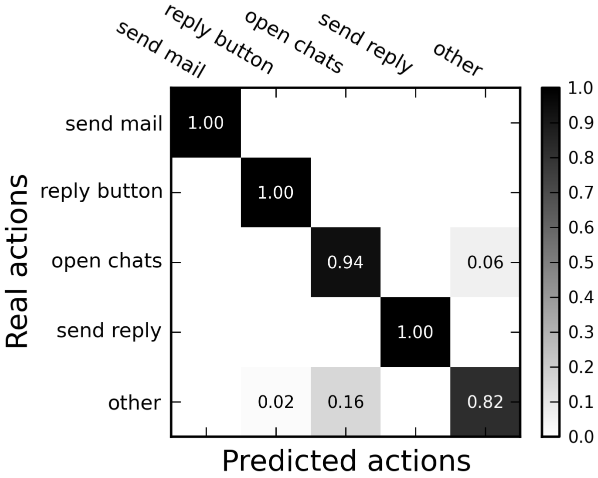


Figure : Gmail user actions confusion matrix for Configuration 1.

Twitter I

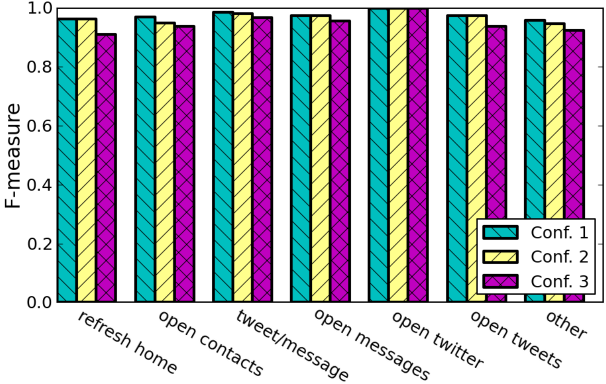


Figure : Classification accuracy of the Twitter user actions.

User actions	**Precision**	**Recall**	**F-measure**
refresh home	0.94	0.99	0.96
open contacts	0.97	0.96	0.97
tweet/message	0.97	1.00	0.98
open messages	1.00	0.95	0.97
open twitter	1.00	1.00	1.00
open tweets	1.00	0.95	0.97
other	0.96	0.96	0.96
Average	0.98	0.97	0.97

Table : Classification results of Twitter user actions reached using the Configuration 1.

Twitter III

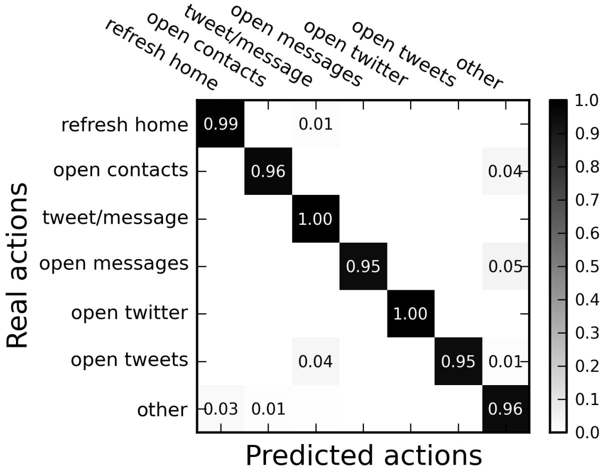


Figure : Twitter user actions confusion matrix for Configuration 1.

LinearSVC accuracy

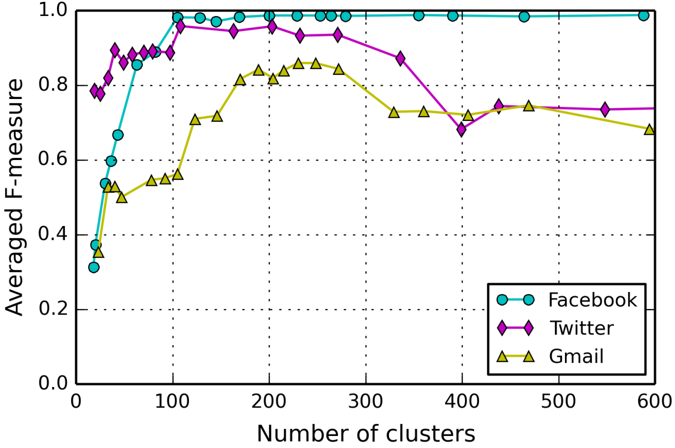


Figure : Accuracy obtained with LinearSVC classifier.

Random forest accuracy

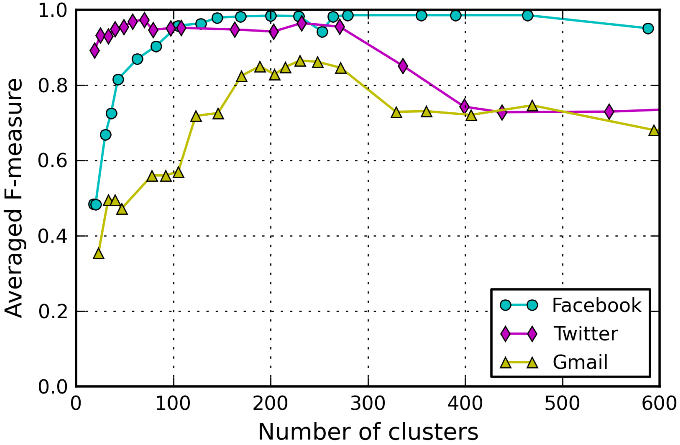


Figure : Accuracy obtained with Random forest classifier.

Gaussian naive Bayes accuracy

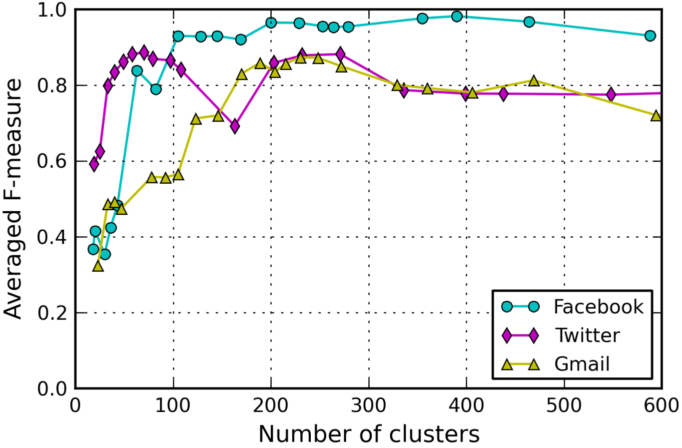


Figure : Accuracy obtained with Gaussian naive Bayes classifier.

Packets protocol (Complete)

Protocol	Facebook	Gmail	Twitter	Total
TLSv1	96.0% (33398)	90.9% (7205)	96.3% (13435)	95.4% (54038)
TCP	03.7% (1316)	00.8% (70)	03.7% (522)	03.4% (1908)
HTTP	00.0% (0)	08.1% (644)	00.0% (0)	01.1% (644)
SSLv2	00.1% (63)	00.0% (0)	00.0% (0)	00.1% (63)

Table : Protocol presence among all considered apps.

Payload analysis is not feasible because almost the 95% of captured packets are **encrypted with TLS protocol**.

Metrics

$$precision = \frac{TP}{TP + FP}.$$

$$recall = \frac{TP}{TP + FN}.$$

$$F_1 = 2 * \frac{precision * recall}{precision + recall} = \frac{2 * TP}{2 * TP + FP + FN}.$$