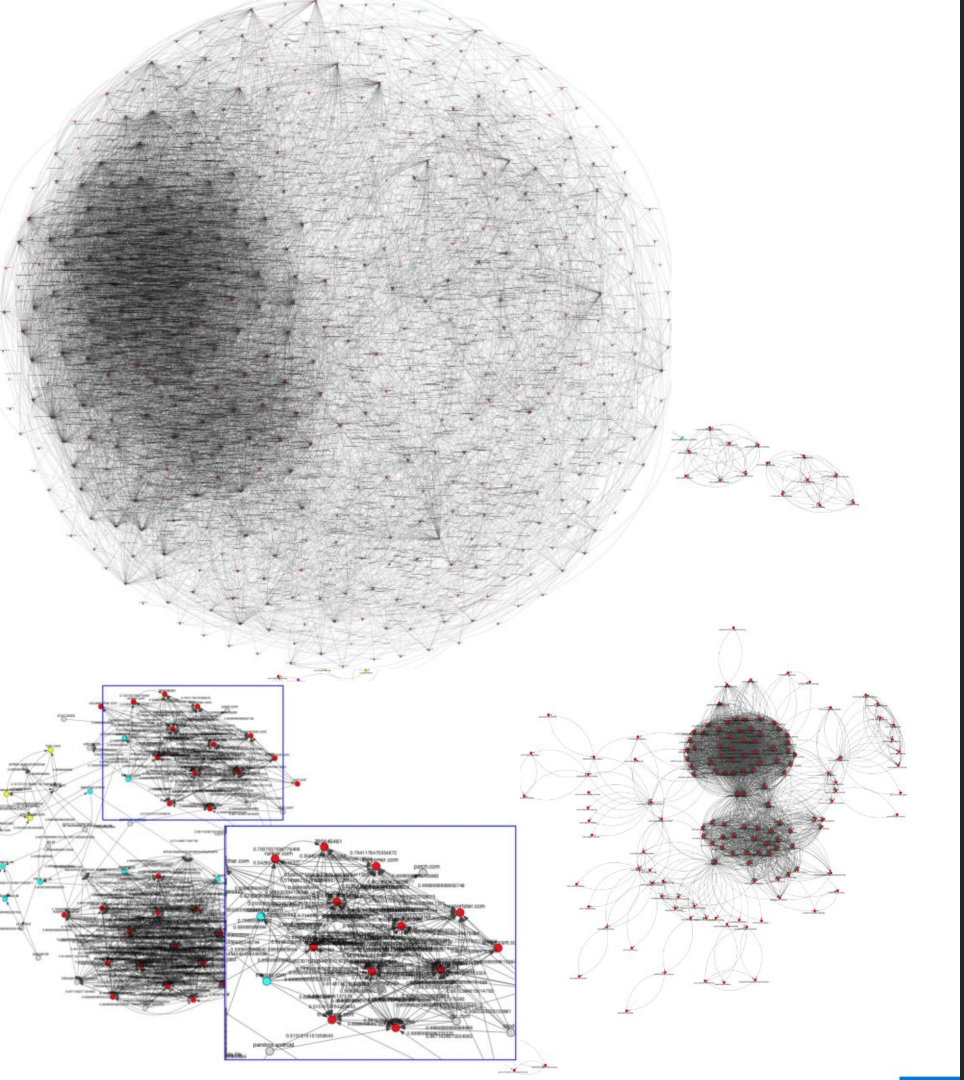
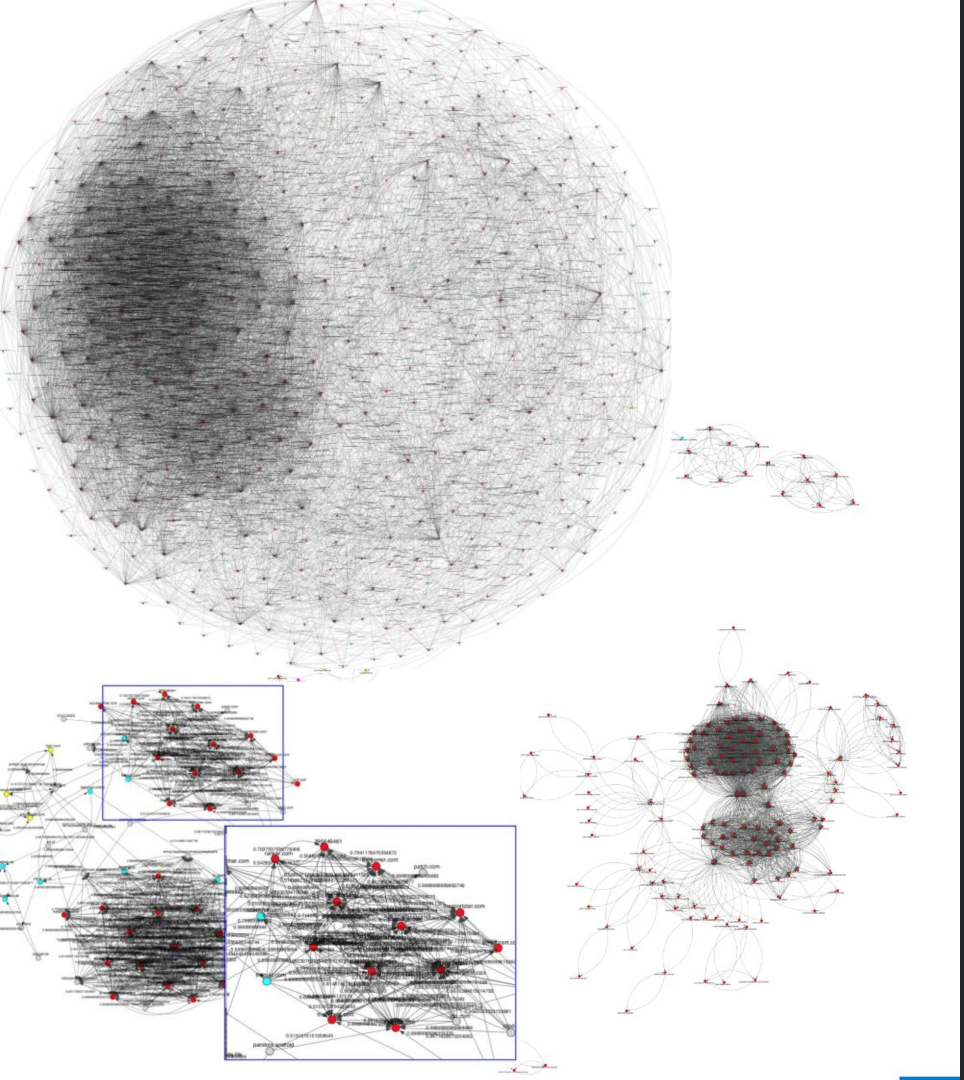


Development of novel algorithms for fraud detection in online advertising

Olaya García Fernández

Master Thesis Cybersecurity 18-19



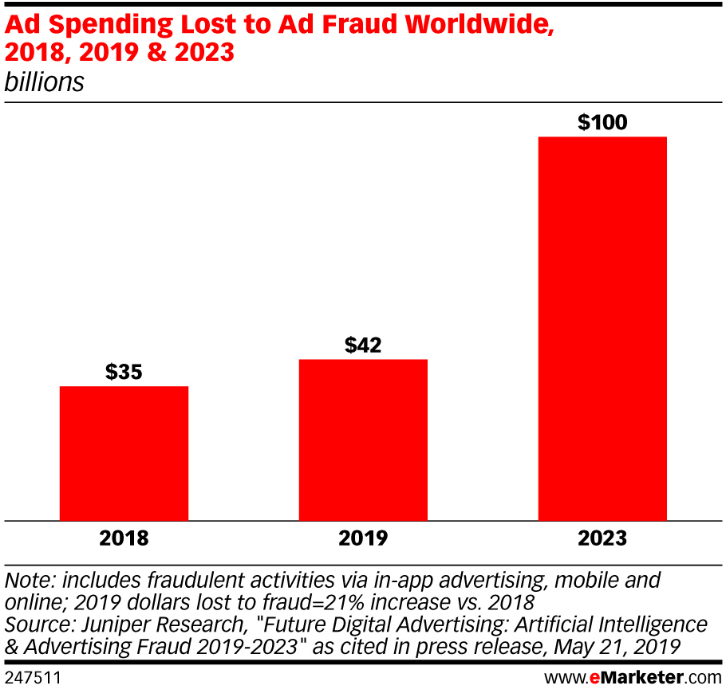
Development of novel algorithms for fraud detection in online advertising

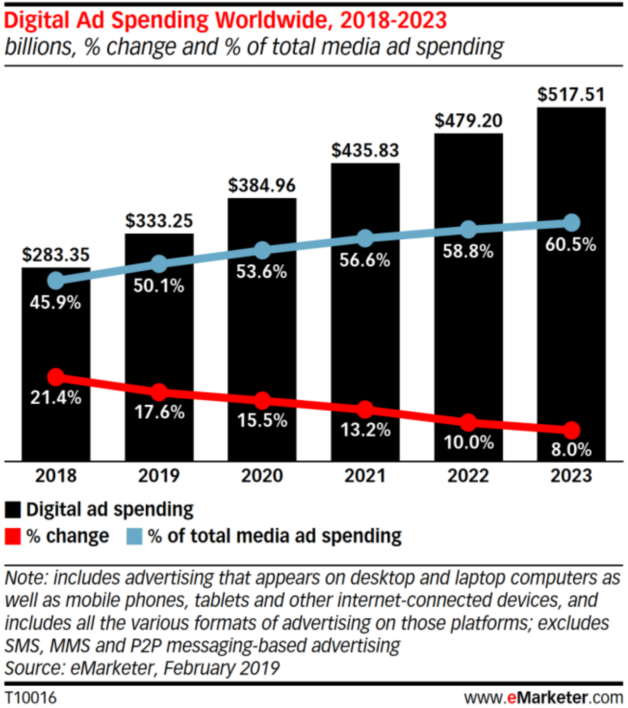
Olaya García Fernández

Master Thesis Cybersecurity 18-19

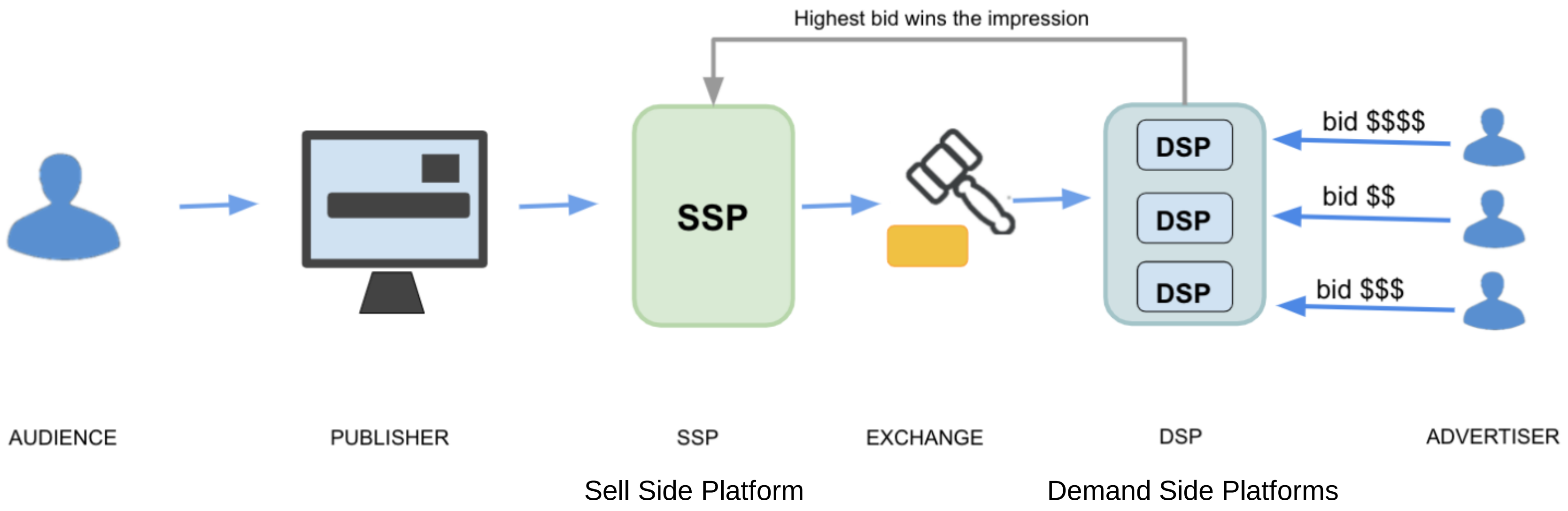
Introduction

- **Web Advertising makes money**
- Fraud Losses increasing





Lifecycle of an Ad



The fraud problem

What is fraud?

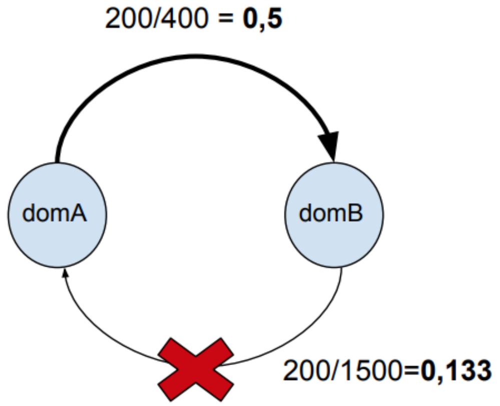
Focus:

- Programmatic Advertising
- Invalid Traffic
- Non Intentional Traffic (NIT)

Why?

- open ecosystem
- Ad Fraud is not illegal
- monetary reward based on the volume of transactions

Co-visitation Networks



$$
{}^{n}_{d}\mathrm{G} = (V_d \subseteq D, E = \{(domA, domB) : domA, domB \in D,
\left[\Gamma_G(domA) \cap \Gamma_G(domB)\right] / \Gamma_G(domA))
$$

Using Co-Visitation Networks For Detecting Large Scale Online Display Advertising Exchange Fraud

Ori Stitelman,
Claudia Perlich
m6d Research
37 E. 18th Street
New York, NY
claudia@m6d.com

Brian Dalessandro,
Rod Hook, Troy Raeder
m6d Research
37 E. 18th Street
New York, NY

Foster Provost
NYU/Stern School
& m6d Research
44 W. 4th Street
New York, NY

Algorithm

Dataset

Logs from incoming requests that the DSPs exchange with the AdEchange.

user_ip	uuid_hashed	useragent	referrer_domain	ssp_domain	date_time

user_ip : IP addr of the user that creates de Ad-request.

referrer_domain : publishers ad-request referrer domain.

300 logs/200MB per day csv.gzip format ; 13000000 TOTAL rows /per log

Solution

Development Framework

- Apache Spark
- Python
- GraphFrames
- Jupyter Notebook

Why use these technologies ?

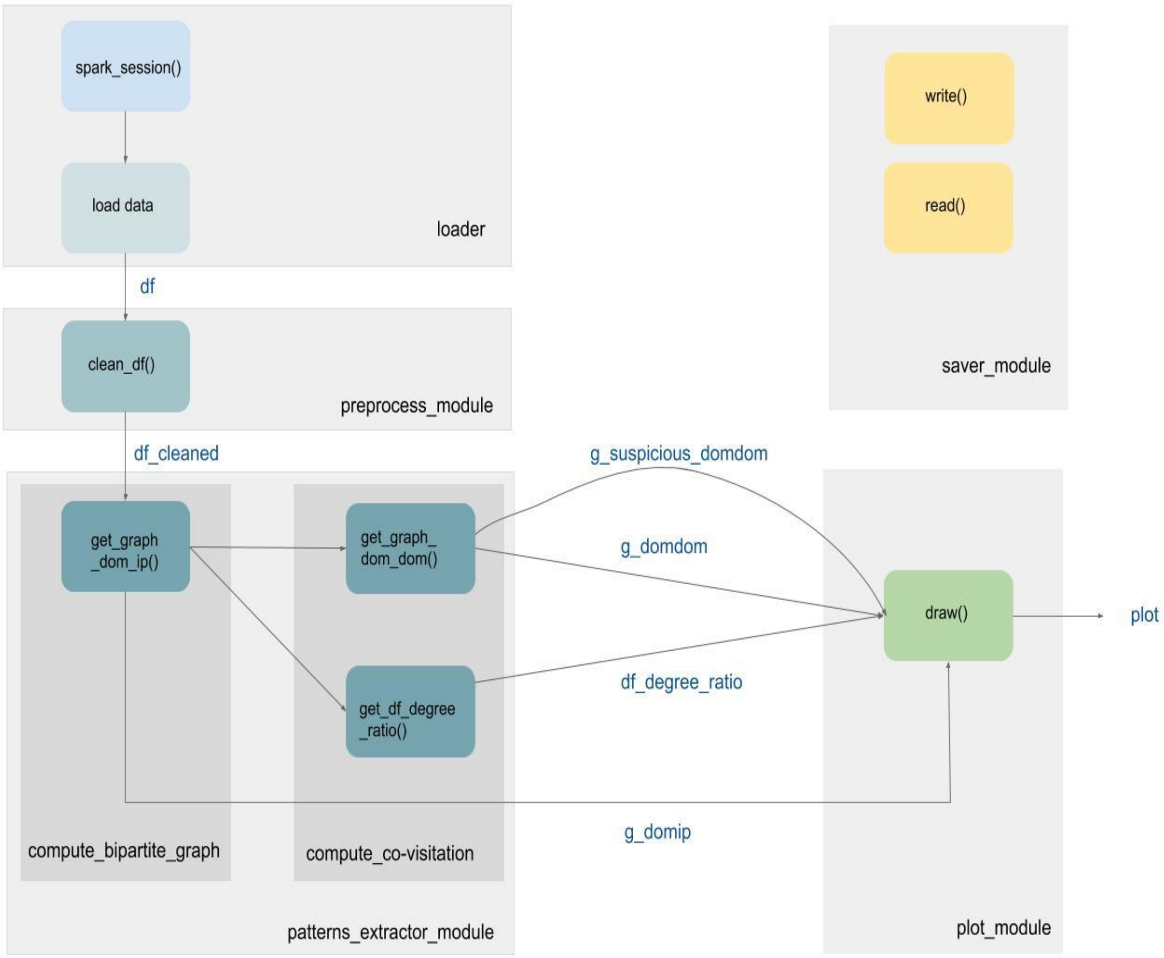
- most popular framework for bigdata
- high speed cluster computation
- data parallelism
- distributed environment

Implementation Design

Flow Chart

Utils library

df_utils.py
gf_utils.py
row_cleaner_utils.py
read_write_utils.py
draw_utils.py
spark_utils.py



Results

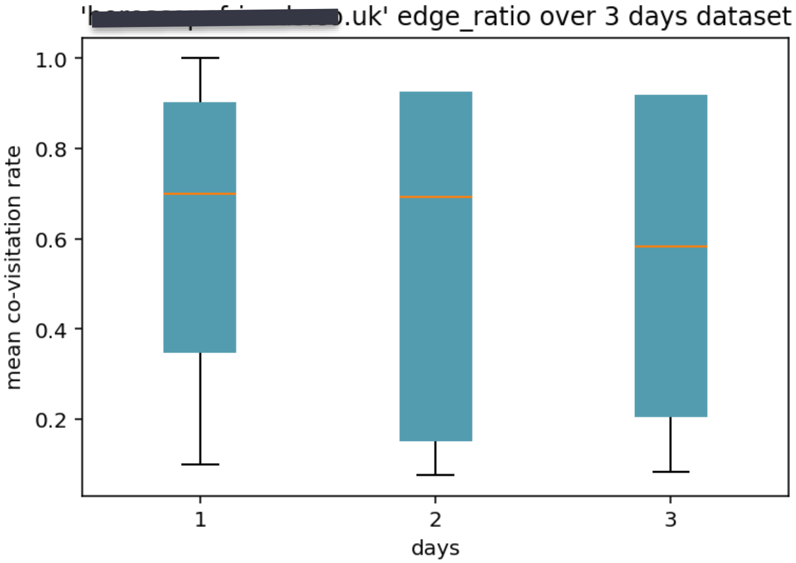
Comparing domains.

		day 1			day 2			day 3		
		outDegree	co-visitation	common_ips	outDegree	co-visitation	common_ips	outDegree	co-visitation	common_ips
Suspicious Domains	[redacted].com	279	0.652	3.907	419	0.436	13.9476	320	0.610	7.0
	[redacted]s.co.uk	147	0.605	6.054	152	0.5706	7.4146	155	0.566	6.794
	[redacted]m.br	90	0.529	3.7	87	0.513	4.103	66	0.677	9.485
Worthy Domains	[redacted]r.com	2815	1.873E-4	9.390	2945	2.008E-4	7.962	2626	2.2996E-4	8.006
	[redacted].gg	4999	7.533E-5	13.146	5002	8.194E-5	14.357	4900	7.559E-5	12.391
	[redacted].com	3775	2.109E-4	13.957	3737	1.987E-4	14.066	3884	1.941E-4	13.525

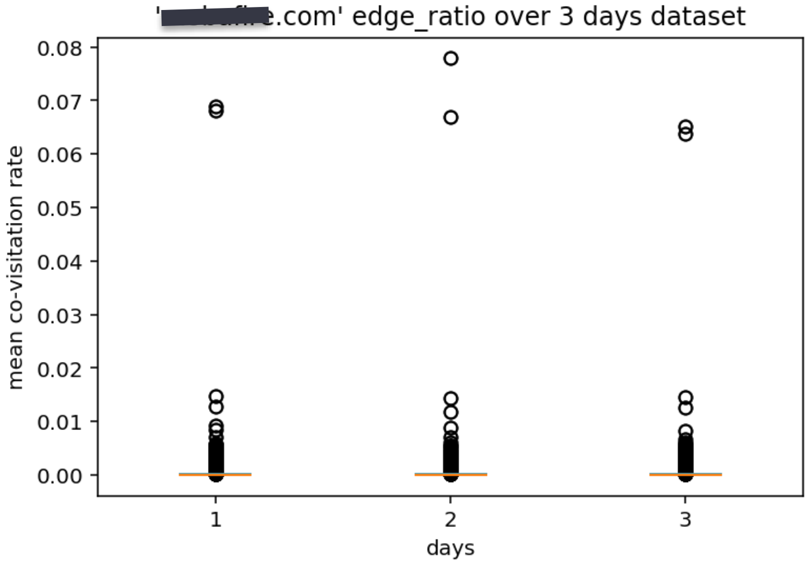
suspicious domain — non suspicious domain

"ratemyjob.com"		"yahoo.com"	
[redacted].com	0.743832	[redacted].com	0.045187
[redacted]org	0.673838	[redacted].com	0.029487
[redacted]com	0.668961	[redacted].com	0.028301
[redacted]er	0.640849	[redacted]r.com	0.027971
[redacted]com	0.595524	[redacted]r.com	0.027190
[redacted].com	0.583189	[redacted].com	0.025908
[redacted]com	0.578026	[redacted].org	0.022630
[redacted]at.co	0.567986	[redacted].com	0.022599
[redacted]com	0.567699	[redacted]com	0.017032
[redacted]s.com	0.563683	[redacted].com	0.016881

Results

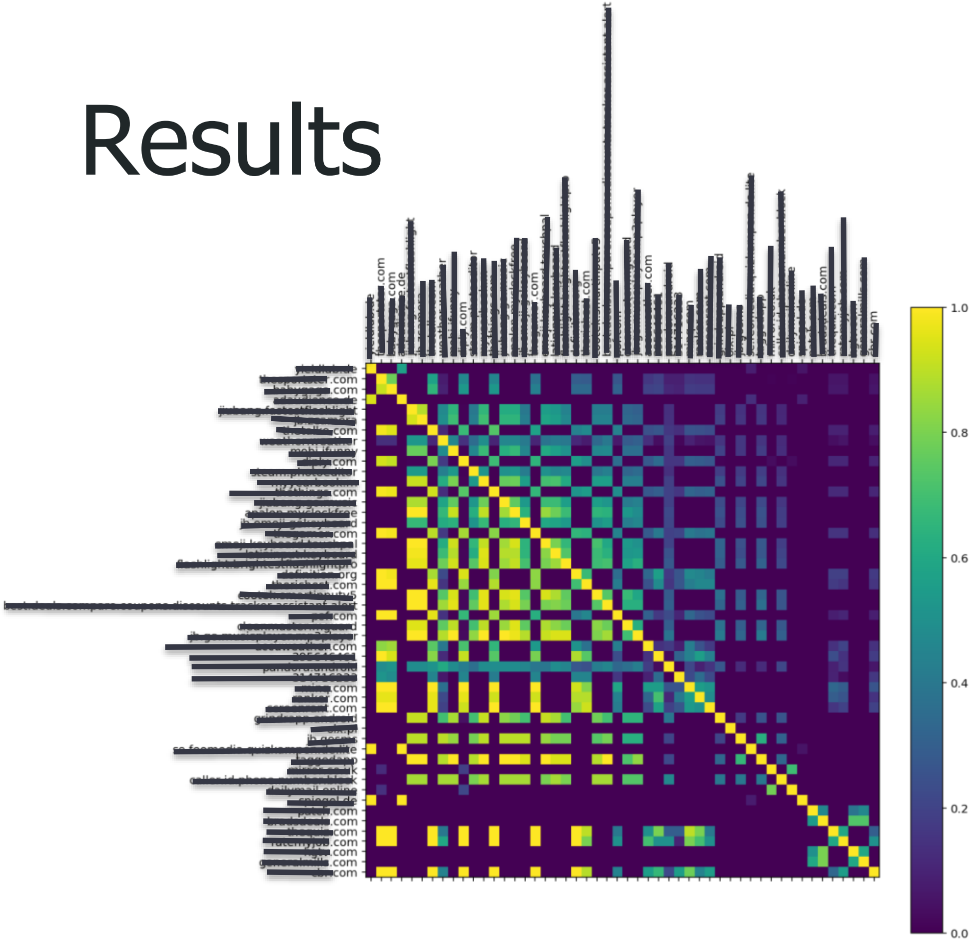


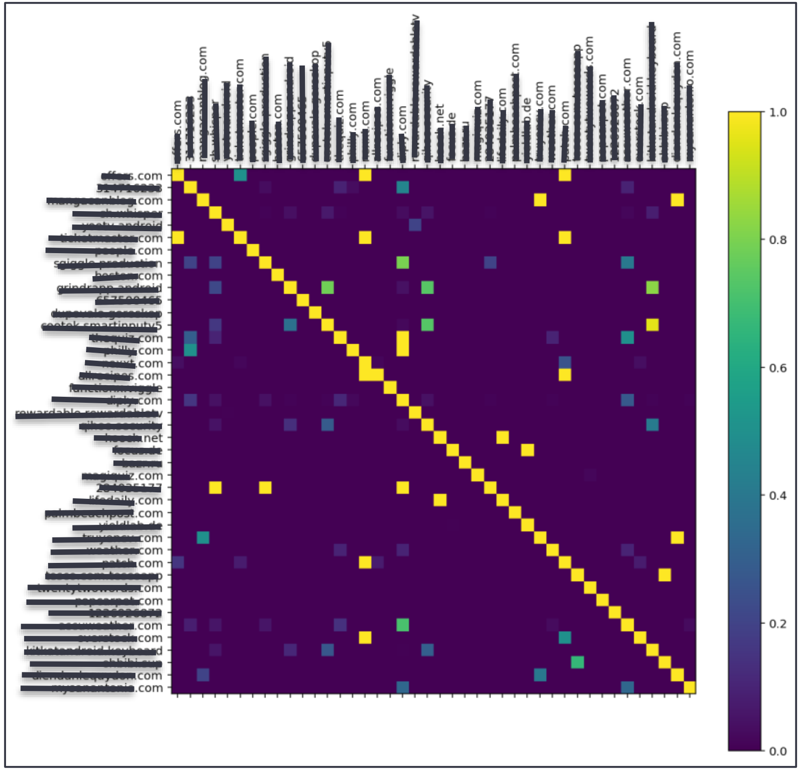
suspicious domain



legal domain

Results





Conclusion

- Algorithm implementation for automatic detection of malicious domains in large datasets with Pyspark.

- Utils library design : scalability , easiness of use, adaptability

  (Graph analysis: Graphframes + Distributed Environment: Pyspark)

- Validation of results Stitelman’s paper even on small sample datasets

Future Work

- Graph Embedding

- Evaluate the algorithm threshold over the time and train IA model.

Acknowledgments

Antonio Pastor

Patricia Callejo

Ruben Cuevas

Luis Peinado