On the Lack of Consensus in Anti-Virus Decisions ; Metrics and Insights on Building Ground Truths of Android Malware

Médéric Hurier, Kévin Allix, Jacques Klein, Tegawende Bissyande, Yves Le Traon

University of Luxembourg

About me (@Freaxmind)

Who am I ?

Fresh PhD Student from Luxembourg

loves Open Source, Automation, Clojure and Learning

http://wwwen.uni.lu/snt/people/mederic_hurier

Member of the SerVal Team

http://wwwfr.uni.lu/snt/research/serval

on Social Networks:

https://github.com/freaxmind

https://twitter.com/Freaxmind

Outline

- Problem Introduction
- Metrics and Insights
- Conclusion
- Questions

Link to the paper

http://hdl.handle.net/10993/27845

Problem Introduction

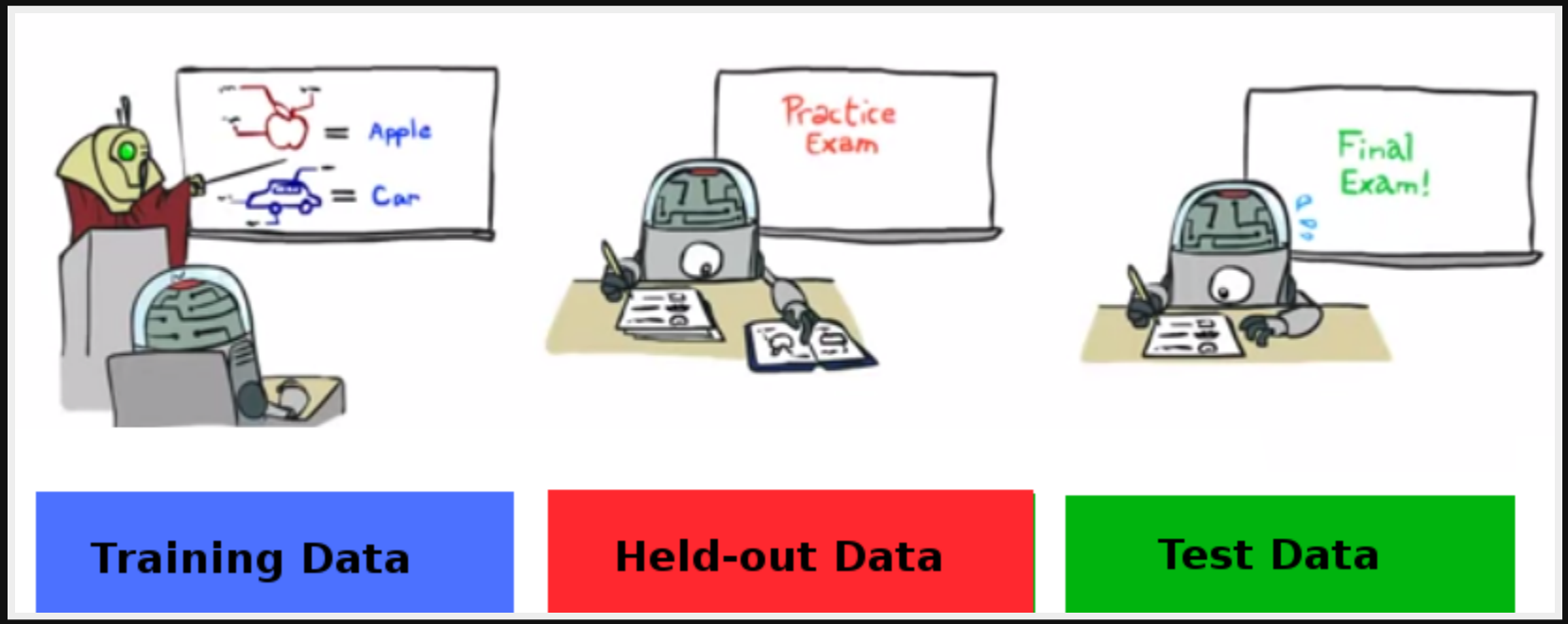
Automated Approaches

Automation is key to classify and detect malware in the large

1. We need to face a **rising number** of malware created every year
   - 575 000 new samples between July and Sept. 2015 (@GData)
2. Besides, industries report a **shortage of cybersecurity skills**
   - more than 1M unfilled security jobs worldwide (@CISCO)
3. We also need to encourage Machine to Machine expansions
   - Machine Learning, Internet of Things, Cloud, DevOps ...

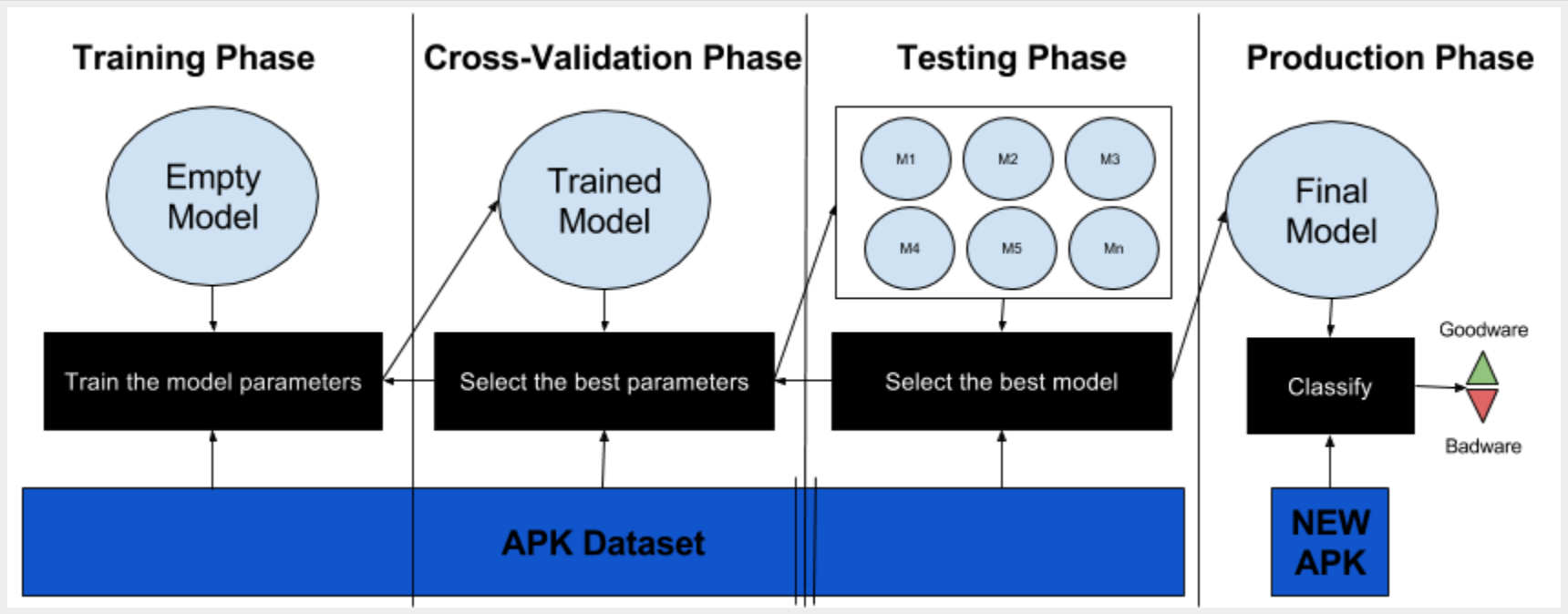
The importance of Malware Ground-Truths

Ground-Truths are key to automation and good results



Source: UC Berkeley CS188 (Intro to AI)

Why does a ML algorithm learn to recognize good/badware instead of apples and cars ?



It only depends on your input data and their quality !

GIGO: "Garbage in, garbage out"

Are you confident in your malware ground-truth ?

Check, Correlate and Convey

Why do we need metrics to evaluate our ground-truths ?

- **Validate** the properties of a dataset against some requirements
- **Observe** the consensus and conflicts between antivirus engines
- **Provide** a high-level picture that can be shared between peers

How can this work help you in this regard ?

- We defined a set of 9 metrics to examine dataset of labels
- We discuss the pros/cons of popular filtering approaches
- We provide an open-source tool written in Python 3 (MIT)

Metrics and Insights

Specifications

- Dataset of Android Apps and Antivirus: **Androzoo** / **VirusTotal**
  - $\mathscr{A}$: Set of Antivirus engines # n = 66 (anonymized)
  - $\mathscr{P}$: Set of Android applications # m = 689 209
  - $\mathcal{B}$: Matrix of Binary decisions # 689 209 × 66
  - $\mathcal{L}$: Matrix of Label strings # 689 209 × 66
  - R_i, C_j: Row Vector i and Column Vector j

- Reused functions through the presentation/paper:
  - ***positives***: number of positive detections of a vector/matrix
  - ***exclusives***: number of exclusive detections of a matrix
  - ***distincts***: number of distinct labels of a vector/matrix
  - ***freqmax***: maximum absolute frequency of a vector
  - ***clusters***: grouped elements of a matrix
  - ***PairwiseSimilarity***: Overlap or Jaccard
  - ***Ouroboros***: statistical dispersion

Binary Decisions ($\mathcal{B}$)

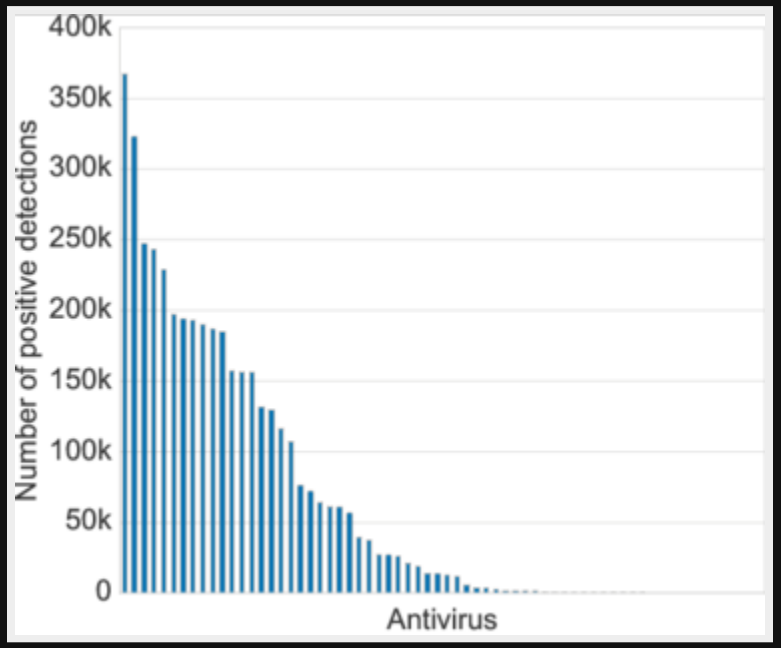
F	AV1	AV2	AV3
F1	1	0	1
F2	1	0	1
F3	1	1	1
F4	1	0	1
F5	1	1	1
F6	0	1	1

Label Strings ($\mathcal{L}$)

F	AV1	AV2	AV3
F1	Android/Deng.FV		Trojan.AndroidOS.Generic.A
F2	AndroidOS.AdWare.Bankun		Artemis!D8536179E295
F3	Android.Adware.Dowgin.AW	Adware/ANDR.Kuguo.K.Gen	TROJ_GEN.F47V0123
F4	Android.Adware.Adware/Umeng.W		Trojan/AndroidOS.eee
F5	AndroidOS/Gappusin.A	Android.Adware.Wapsx.A	Adware.AndroidOS.AirPush.a
F6		Trojan.Dex.Secapk.cussul	Android/Secapk.A!tr

Analysis of Antivirus Decisions

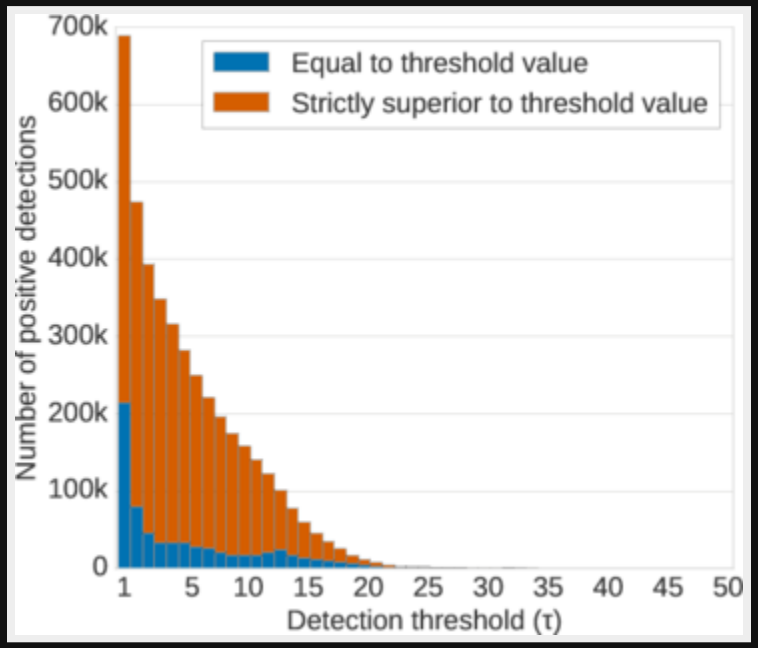
Measure of Participation: Equiponderance



$$Equiponderance(\mathcal{B}) = Ouroboros(X)$$

$$\text{with } X = \{positives(C_j) : C_j \in \mathcal{B}, 1 \leq j \leq n\}$$

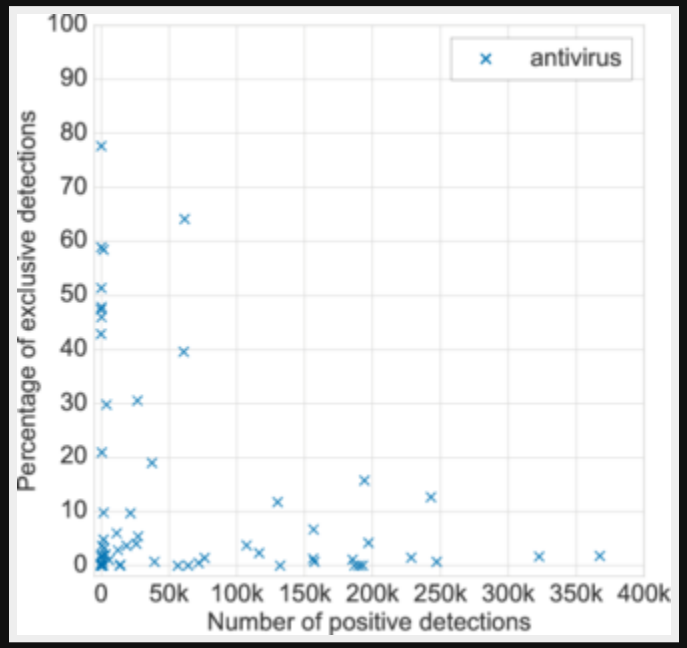
Measure of Detection: Recognition



$$Recognition(\mathcal{B}) = \frac{\sum_{i=1}^{m} X_i}{n \times m}$$

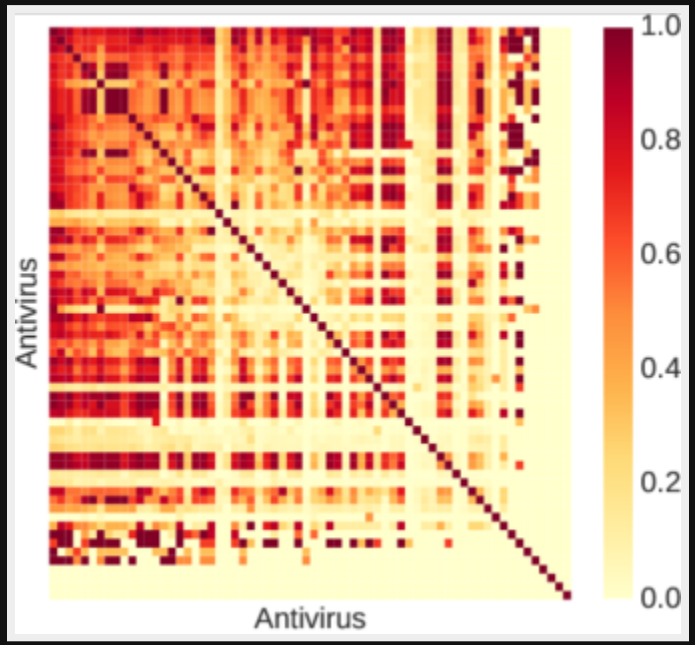
$$\text{with } X = \{positives(R_i) : R_i \in \mathcal{B}, 1 \leq i \leq m\}$$

Measure of Specificity: Exclusivity



$$Exclusivity(\mathcal{B}) = \frac{exclusives(\mathcal{B})}{m}$$

Measure of Similarity: Synchronicity

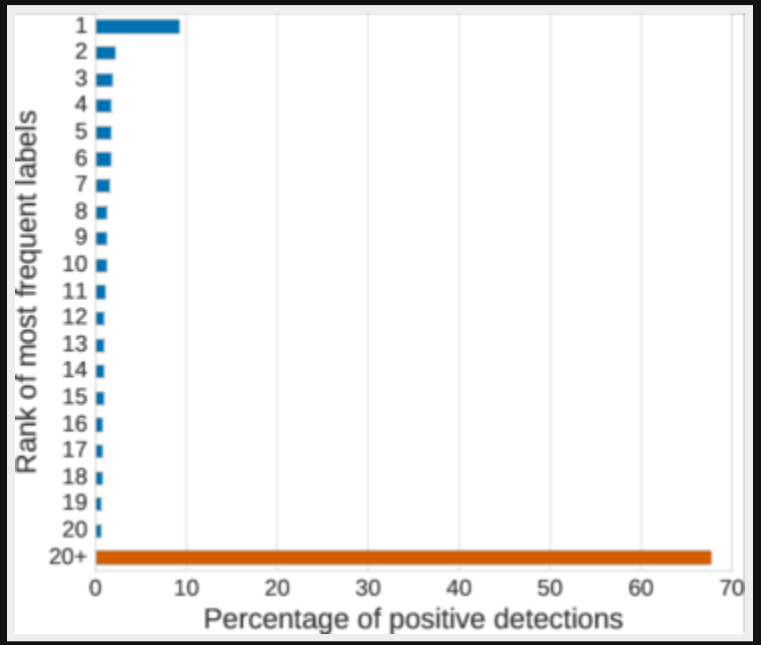


$$Synchronicity(\mathcal{B}) = \frac{\sum_{j=1}^{n} \sum_{j'=1}^{n} PairwiseSimilarity(C_j, C_{j'})}{n(n-1)}$$

$$\text{with } j \neq j', C_j \in \mathcal{B}, C_{j'} \in \mathcal{B}$$

Analysis of Malware Labels

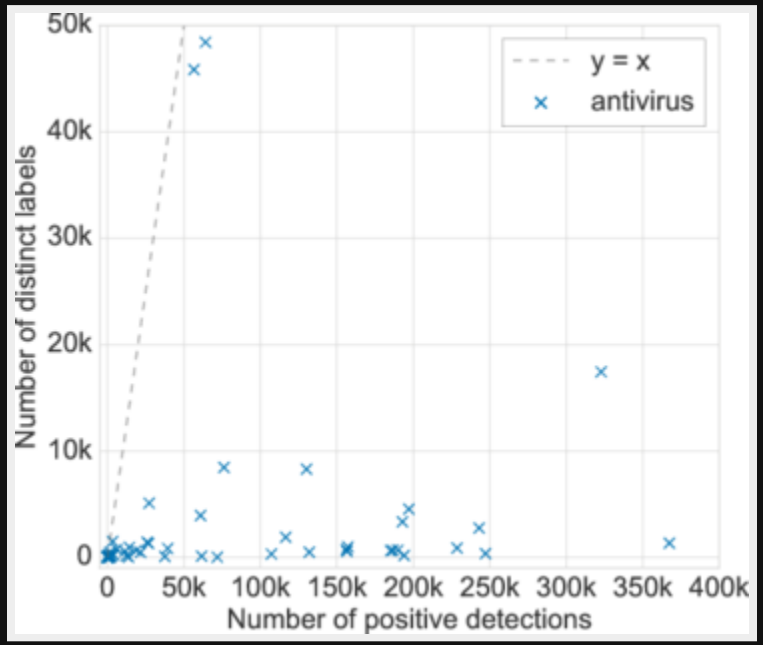
Measure of Distribution: Uniformity



$$Uniformity(\mathcal{L}) = Ouroboros(X)$$

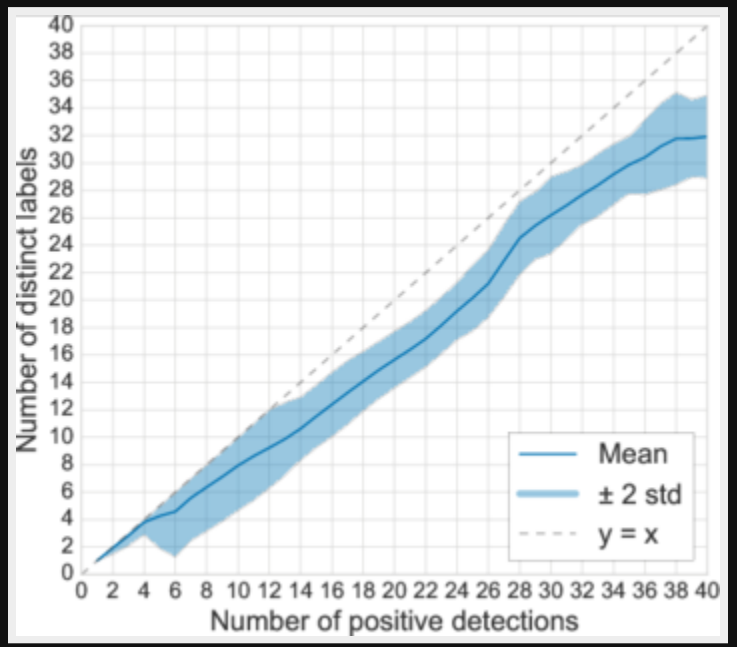
$$\text{with } X = \{|l_k| : l_k \in clusters(\mathcal{L})\}$$

Measure of Information: Genericity



$$Genericity(\mathcal{L}) = 1 - \frac{distincts(\mathcal{L}) - 1}{positives(\mathcal{L}) - 1}$$

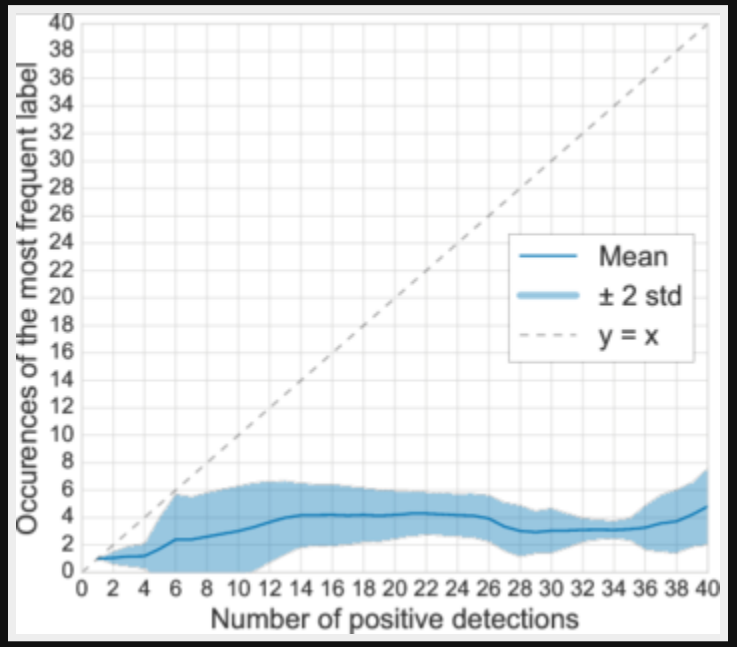
Measure of Opinions: Divergence



$$Divergence(\mathcal{L}) = \frac{(\sum_{i=1}^{m} X_i) - n}{positives(\mathcal{L}) - n}$$

$$\text{with } X = \{distincts(R_i) : R_i \in \mathcal{L}, 1 \leq i \leq m\}$$

Measure of Agreement: Consensuality



$$Consensuality(\mathcal{L}) = \frac{(\sum_{i=1}^{m} X_i) - n}{positives(\mathcal{L}) - n}$$

$$\text{with } X = \{freqmax(R_i) : R_i \in \mathcal{L}, 1 \leq i \leq m\}$$

Conclusion

Results

Dataset Evaluated (derived from Androzoo):

- **Baseline**: remove Android apps without any positive detections
- **Filtered**: apply heuristics from the literatture (Arp D. et al., 2014)
  1. use a set of 10 popular AVs
  2. select apps detected by at least two AVs
  3. remove apps where at least one AV label contains "adware"
- **Genome**: manually verified dataset (Zhou Y., Jiang X., 2012)

Baseline: 689 209 apps, Filtered = 44 615 apps, Genome: 1 248 apps

Dataset	Equiponderance	Exclusivity	Recognition	Synchronicity	Uniformity	Genericity	Divergence	Consensuality
Baseline	0.27	0.31	0.09	0.32	0.001	0.97	0.77	0.21
Filtered	0.59	0	0.36	0.75	0.01	0.87	0.95	0.05
Genome	0.48	0	0.48	0.41	0.04	0.82	0.87	0.06

Take Home Messages

What did we learn from these results ?

- Filtering processes have large impacts on the output dataset
- A handful of metrics can provide a high-level picture (mean, sd)

What did we learn from this project ?

- It is much harder to exploit label strings than binary decisions
- What are we **really** feeding to our machine-learning algorithms ?

Future Work

- Use these metrics to optimize the learning process of algorithms
- Develop a method to improve the comprehension of label strings

GitHub Repository:

https://github.com/freaxmind/STASE

- **Input**: a dataset of labels formatted as a CSV (or CSV.GZ) file
- **Output**: metrics introduced in the presentation/paper

```
python3 stase.py sample.csv.gz output.json

{
    "equiponderance": 0.2422919148,
    "equiponderance_idx":8.0,
    "exclusivity":0.2626262626,
    "recognition":0.1051423324,
    "synchronicity":0.1677210336,
    "genericity":0.5233236152,
    "uniformity":0.2926562999,
    "uniformity_idx":48.0,
    "divergence":0.7568027211,
    "consensuality":0.2227891156,
    "resemblance":0.6406466991,
    "labels":328.0,
    "apps":99.0,
    "avs":66.0,
}
```

Questions

Thank you for your attention