ANDROMAK:
State of the art Android ML based detection systems at your fingertips

Matthieu Simonin (SED Rennes)
David Bromberg (WIDE Team)
Interested in the experimentation practices in various contexts

- **Distributed systems**\(^1\)
- **Organizer / participant of meetings around experimentations**
  - XUG meetings: https://xug.gitlabpages.inria.fr/meetings/ (Check out the upcoming sessions)
  - AskTheSED: coming soon

Recently interested in Android malware classifications methodology

- Static analysis with scale requirements
- Environment local machine, Grid’5000

\(^1\)Cherrueau et al. EnosLib: A Library for Experiment-Driven Research in Distributed Computing. 10.1109/TPDS.2021.3111159
Some contributions focus on building $C$
  ▶ "Here is a new $C$ and I’ll show you how good it is"

Some contributions focus on evading an existing $C$
  ▶ Black/white box adversarial / poisoning
  ▶ "This dataset evades this $C$, here’s how we made it."

Some (less) contributions focus on the overall methodology
  ▶ Reproducibility
  ▶ Bias identification and evaluation
  ▶ Brings nuances: "yes your $C$ is better in this setting BUT…"

The three kinds of contributions are complementary$^2$

---

In a supervised context:

\[
D = \{APK\} \xrightarrow{fem} X \xrightarrow{gt} y \xrightarrow{train} C \xrightarrow{test} scores
\]

\[Eval : (D, fem, gt, C, S) \rightarrow scores\]

Some baseline:
- **Drebin paper:**
  - \[Eval(120\,K + 5\,K(drebin), fem_{drebin}, VirusTotal*, SVC, 66/33)\]
  - comparison with other related fem on the same dataset

- **Mamadroid paper:**
  - \(D\): product of GW (oldbenign/new benign) + MW(drebin/ V.Share)
  - \(C\): Random Forest / k-NN
  - \(S\): 10-fold CV + mean f1 scores
You need Variations\textsuperscript{3} into the pipeline == parameter change

- \( \mathcal{D} \): dataset
- fem: feature extraction method (core of many contributions)
- gt: ground truth (there isn’t one true ground-truth)
- \( C \): model / hyperparameters (including DNN models...)
- selection procedure (5/10 fold-CV, time-aware split . . .)
- and the associated statistical analysis (hypothesis testing . . .)

\textsuperscript{3} Feitelson, D.: From repeatability to reproducibility and corroboration. ACMSIGOPS Oper. Syst. Rev.49, 3–11 (2015)
Example of variations in time series classification

2 classical dimensions to fiddle with: datasets(x85) and models(x10)

---


Matthieu Simonin
Question

Are we there in the Android Malware classification context?

Not quite

\[ \text{Eval: } (D, \text{ fem, gt, C, S}) \rightarrow \text{scores} \]

- hard to vary
  - No reference datasets
  - Unavailability / deprecated
  - Tight coupling between \( D \) and \( \text{fem/gt} \)

- easy to vary
  - Nice APIs (sklearn / tensorflow . . .)

---

Walkthrough Andromak

A library to play with: "Eval : (D, fem, gt, C, S) → scores" in the Android context

- Goal 1: Allow experimenters to build easily various evaluations (write their own Eval functions)
- Goal 2: Knowledge base around classical techniques around Android malware classification

Design principles:
- Provide an implementation compatible for minimalistic infrastructure (single machine or cluster of machines with a minimal shared storage)
- Favor(abuse of?) functionnal aspects of the language
Knowledge base (datasets)

The library has built-in support for

<table>
<thead>
<tr>
<th>Dataset</th>
<th>#sample</th>
<th>MW%</th>
<th>Ref</th>
</tr>
</thead>
<tbody>
<tr>
<td>CICMalAnal17</td>
<td>4K</td>
<td>50%</td>
<td>10.1016/j.cose.2011.12.012</td>
</tr>
<tr>
<td>GM19</td>
<td>10K</td>
<td>50%</td>
<td>hal-02288116</td>
</tr>
<tr>
<td>Maldroid20</td>
<td>17K</td>
<td>75%*</td>
<td>10.1109/DASC-PICom-...</td>
</tr>
<tr>
<td>Malscan19</td>
<td>30K</td>
<td>50%</td>
<td>10.1109/ASE.2019.00023</td>
</tr>
<tr>
<td>Tess19</td>
<td>120K</td>
<td>10%</td>
<td>usenix/Pendlebury19</td>
</tr>
</tbody>
</table>

How ? Store abstraction

- Generic API for getting/storing APK, metadata, any transformation of APK/metadata
- Can be populated from an existing dataset (on disk or dl from androzoo)
- Can be backed in a FS (or a DB – theoretically)
Knowledge base (fem)

Various *fem* (decoupled from the datasets):

<table>
<thead>
<tr>
<th>fem</th>
<th>Réf</th>
</tr>
</thead>
<tbody>
<tr>
<td>Drebin</td>
<td>ndss/ArpSHGR14</td>
</tr>
<tr>
<td>Mamadroid</td>
<td>10.1145/3313391</td>
</tr>
<tr>
<td>Malscan</td>
<td>10.1109/ASE.2019.00023</td>
</tr>
<tr>
<td>Revealdroid</td>
<td>10.1145/3162625</td>
</tr>
<tr>
<td>Hindroid</td>
<td>0.1145/3097983.3098026</td>
</tr>
</tbody>
</table>

How ? Rely on the definition of

- **extract**: $APK \rightarrow Feature$
  - can be full python or wrap a docker execution
  - **serializable** $\Rightarrow$ implicit parallelism/distribution on a cluster

- **dump**: $Feature \rightarrow bytes$
  - compression / sparsification
  - storage optimization
Example: massively parallel fem execution

extract serializable ⇒ implicit parallelism/distribution on a cluster

- screenshot of $fem_{drebin}$ execution on 1M apk
- Andromak uses Dask\(^6\) transparently (here on Grid5000 using 50 nodes / 250 workers)

\(^6\)https://docs.dask.org/en/latest/
<table>
<thead>
<tr>
<th>FEM</th>
<th>MalAnal17</th>
<th>GM19</th>
<th>MalDroid20</th>
<th>Malscan17</th>
<th>Tess19</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>reference</strong></td>
<td>apps</td>
<td>2.126</td>
<td>10.000</td>
<td>17.187</td>
<td>30.715</td>
</tr>
<tr>
<td></td>
<td>mw</td>
<td>426</td>
<td>5.000</td>
<td>13.150</td>
<td>15.430</td>
</tr>
<tr>
<td></td>
<td>gw</td>
<td>1.700</td>
<td>5.000</td>
<td>4.037</td>
<td>15.285</td>
</tr>
<tr>
<td><strong>common</strong></td>
<td>apps</td>
<td>1.940</td>
<td>9.158</td>
<td>15.781</td>
<td>28.856</td>
</tr>
<tr>
<td></td>
<td>gw</td>
<td>1.533</td>
<td>4.835</td>
<td>3.609</td>
<td>14.748</td>
</tr>
<tr>
<td><strong>drebin</strong></td>
<td>apps</td>
<td>2.116</td>
<td>10.000</td>
<td>17.186</td>
<td>30.715</td>
</tr>
<tr>
<td></td>
<td>mw</td>
<td>426</td>
<td>5.000</td>
<td>13.149</td>
<td>15.430</td>
</tr>
<tr>
<td></td>
<td>gw</td>
<td>1.690</td>
<td>5.000</td>
<td>4.037</td>
<td>15.285</td>
</tr>
<tr>
<td><strong>hindroid</strong></td>
<td>apps</td>
<td>2.111</td>
<td>9.930</td>
<td>16.661</td>
<td>30.690</td>
</tr>
<tr>
<td></td>
<td>mw</td>
<td>425</td>
<td>4.941</td>
<td>12.630</td>
<td>15.430</td>
</tr>
<tr>
<td></td>
<td>gw</td>
<td>1.686</td>
<td>4.989</td>
<td>4.031</td>
<td>15.260</td>
</tr>
<tr>
<td><strong>malscan</strong></td>
<td>apps</td>
<td>2.121</td>
<td>9.947</td>
<td>16.702</td>
<td>30.715</td>
</tr>
<tr>
<td></td>
<td>mw</td>
<td>425</td>
<td>4.947</td>
<td>12.671</td>
<td>15.430</td>
</tr>
<tr>
<td></td>
<td>gw</td>
<td>1.696</td>
<td>5.000</td>
<td>4.031</td>
<td>15.285</td>
</tr>
<tr>
<td><strong>mamadroguard</strong></td>
<td>apps</td>
<td>2.121</td>
<td>9.945</td>
<td>16.661</td>
<td>30.715</td>
</tr>
<tr>
<td></td>
<td>mw</td>
<td>425</td>
<td>4.945</td>
<td>12.630</td>
<td>15.430</td>
</tr>
<tr>
<td></td>
<td>gw</td>
<td>1.696</td>
<td>5.000</td>
<td>4.031</td>
<td>15.285</td>
</tr>
<tr>
<td><strong>mamadroid</strong></td>
<td>apps</td>
<td>1.978</td>
<td>9.288</td>
<td>15.881</td>
<td>29.333</td>
</tr>
<tr>
<td></td>
<td>mw</td>
<td>408</td>
<td>4.430</td>
<td>12.211</td>
<td>14.527</td>
</tr>
<tr>
<td></td>
<td>gw</td>
<td>1.570</td>
<td>4.858</td>
<td>3.670</td>
<td>14.806</td>
</tr>
<tr>
<td><strong>revealroid</strong></td>
<td>apps</td>
<td>2.100</td>
<td>9.764</td>
<td>16.325</td>
<td>30.142</td>
</tr>
<tr>
<td></td>
<td>mw</td>
<td>420</td>
<td>4.781</td>
<td>12.355</td>
<td>14.897</td>
</tr>
<tr>
<td></td>
<td>gw</td>
<td>1.680</td>
<td>4.983</td>
<td>3.970</td>
<td>15.245</td>
</tr>
</tbody>
</table>

**Table 3.** Extraction effectiveness. number of applications (resp. malware, resp. goodware) that are initially in the various datasets (reference), or successfully extracted for each extraction method. Applications that are successfully extracted by all the extraction techniques are accounted in the common bloc.
Knowledge base (Domain specific ML)

- Some default $C$ (for reproducibility purpose)
  - Related classifiers (best ones according to the related papers)
    - e.g. $C_{drebin} = SVC$
    - Multilayer Perceptron

- Specific selection method / Hypothesis testing
  - Time aware splits (usenix/Pendlebury19)
  - Wilcoxon, Ranks ...

---

7 Grosse et al. (2017) Adversarial Examples for Malware Detection. ESORICS 2017. https://doi.org/10.1007/978-3-319-66399-9_4
Example: comparison of 3 variants

Eval: \((D, \text{ fem}, \text{ gt}, C, S) \rightarrow \text{scores}\) with 3 different \text{ fem} and 5 different \text{ C}

- \(S\): 5x2-fold CV (mean) f1-scores

<table>
<thead>
<tr>
<th></th>
<th>MalAnal17</th>
<th>GM19</th>
<th>MalDroid20</th>
<th>Malscan17</th>
<th>Tess19</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>coupled</td>
<td>free</td>
<td>coupled</td>
<td>free</td>
<td>coupled</td>
</tr>
<tr>
<td>mamadroguard</td>
<td>0.82</td>
<td>0.82</td>
<td>0.92</td>
<td>0.92</td>
<td>0.95</td>
</tr>
<tr>
<td>mamadroid</td>
<td>0.82</td>
<td>0.81</td>
<td>0.92</td>
<td>0.92</td>
<td>0.96</td>
</tr>
<tr>
<td>mamadroid-fast</td>
<td>0.82</td>
<td>0.80</td>
<td>0.93</td>
<td>0.93</td>
<td>0.96</td>
</tr>
</tbody>
</table>

- \(S\): time-aware model selection with \text{AUT(f1)}\ score

![Graphs showing performance over time](image)
Conclusion

- **Andromak**
  - Let you introduce in your evaluation pipeline
    - variations
    - scale
  - Future:
    - research report / paper + release
    - sandboxed analysis, new datasets, visualisations . . .

- **Talk to SED people about your experimentations / problems**
  - XUG / AskTheSED events