MalPaCA: Malware behaviour analysis using machine learning

Which clustering algorithm has the best performance in terms of network behaviour discovery?

Hugo de Heer - h.j.deheer@student.tudelft.nl
Supervisors: Azqa Nadeem and Sicco Verwer

TU Delft

1. Background

MalPaCA is created to automate malware capability assessment by clustering the temporal behaviour in malware's network traces.

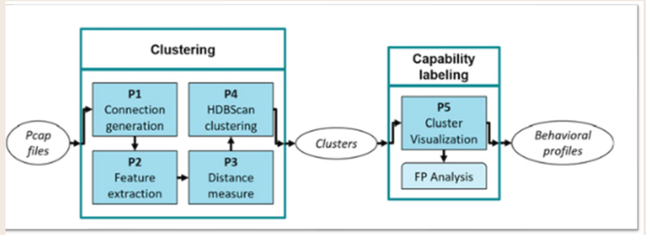

Figure 1: Pipeline of MalPaCA.

2. Methodology

Comparative analysis of the following clustering algorithms:

- HDBScan (baseline)
- OPTICS
- Hierarchical Agglomerative Clustering
- K-medoids

Aposemat IoT-23 labelled dataset was used.

Metrics used to analyse cluster results:

- *sErr* = Silhouette score error
- *cpErr* = Cluster purity error
- *cmpErr* = Cluster malicious purity error
- *nErr* = Noise error
- *ccErr* = Cluster completeness error

3. Experimental setup

All algorithms tested on 2 configurations:
a: with metric = 'precomputed' flag (baseline)
b: no metric = 'precomputed' flag

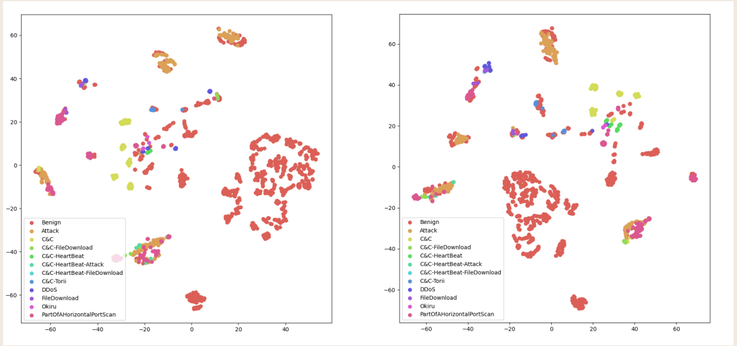

Figure 2: 2D t-SNE projection of the validation set (left) and test set (right).

4a. Results

Cluster configurations	clusters	*sErr*	*cpErr*	*cmpErr*	*nEr*	*ccErr*	*totErr*
HDBScan *a*	58	0.311	0.094	0.081	0.125	0.770	**1.381**
HDBScan *b*	33	0.249	0.179	0.110	0.051	0.483	**1.072**
OPTICS *a*	20	0.424	0.161	0.171	0.418	0.378	**1.553**
OPTICS *b*	13	0.273	0.340	0.254	0.198	0.330	**1.395**
AHC *a*	35	0.221	0.172	0.080	0.000	0.475	**0.950**
AHC *b*	13	0.328	0.316	0.429	0.000	0.308	**1.381**
Kmed *a*	10	0.516	0.435	0.629	0.000	0.458	**2.038**
Kmed *b*	54	0.274	0.169	0.075	0.000	0.717	**1.245**

Figure 3: Metric scores of configurations *a* and *b* for every clustering algorithm included in the comparative analysis.

4b. Results

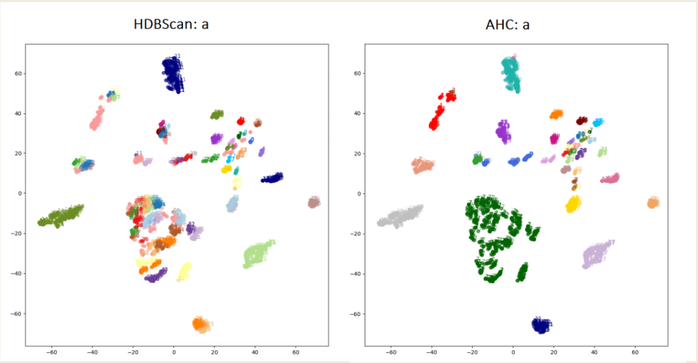

Figure 4: Clustering results of HDBScan *a* and Agglomerative Hierarchical Clustering *a*.

5. Conclusion and future work

- Agglomerative Hierarchical Clustering (AHC) scored best with a total error of 0.950.
- AHC achieves higher cluster separation and cohesion whilst not having a noise cluster, unlike the baseline HDBScan.

Future work:

- Label 'Benign' connections in a more specific way
- Test MalPaCA on more labelled datasets to prevent overfitting.
- Possible use of clustering error score from temporal heatmaps as unlabelled substitute for *cmpErr*.