Automatic Malware Signature Generation and Classification

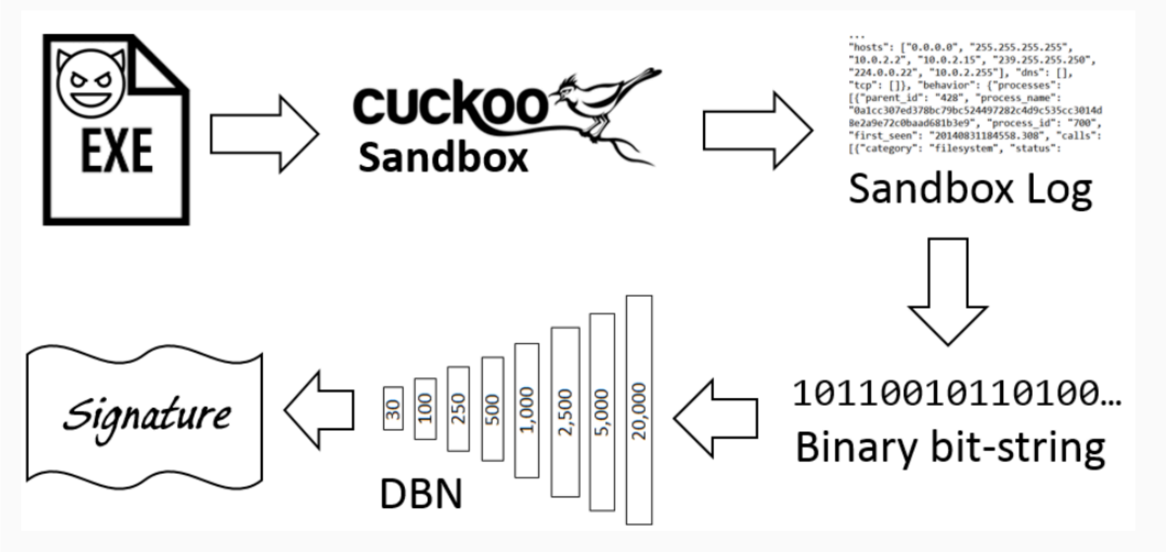
edited by GC

General Malware Detection Drawback

- Spend long time to analyze - manual analysis and handcrafted signature
- Easy to evade - just need a little modify can be undetected by anti-virus.

Is it possible to generate a signature for a program that represents its behavior, and is invariant to small scale changes?

Signature Generation Method



```
...
"hosts": ["0.0.0.0", "255.255.255.255",
"10.0.2.2", "10.0.2.15", "239.255.255.250",
"224.0.0.22", "10.0.2.255"], "dns": [],
"tcp": []}, "behavior": {"processes":
[{"parent_id": "428", "process_name":
"0a1cc307ed378bc79bc524497282c4d9c535cc3014d
8e2a9e72c0baad681b3e9", "process_id": "700",
"first_seen": "20140831184558.308", "calls":
[{"category": "filesystem", "status":
```

- Use sandbox to record the malware behavior.

- Convert the log of the malware into a 20000-dim vector.

- Use auto-encoder to convert the previous vector into 30-dim.

- The 30-dim vector is the signature of the malware.

Record malware behavior

- Which API functions have been called
- Files created or deleted
- IP addresses, URLs and ports accessed
- Registry keys written

Convert log to 20000-dim vector

- A method of natural language processing called unigram extraction is used to the whole dataset

- After extraction, remove unigrams which appear in all data

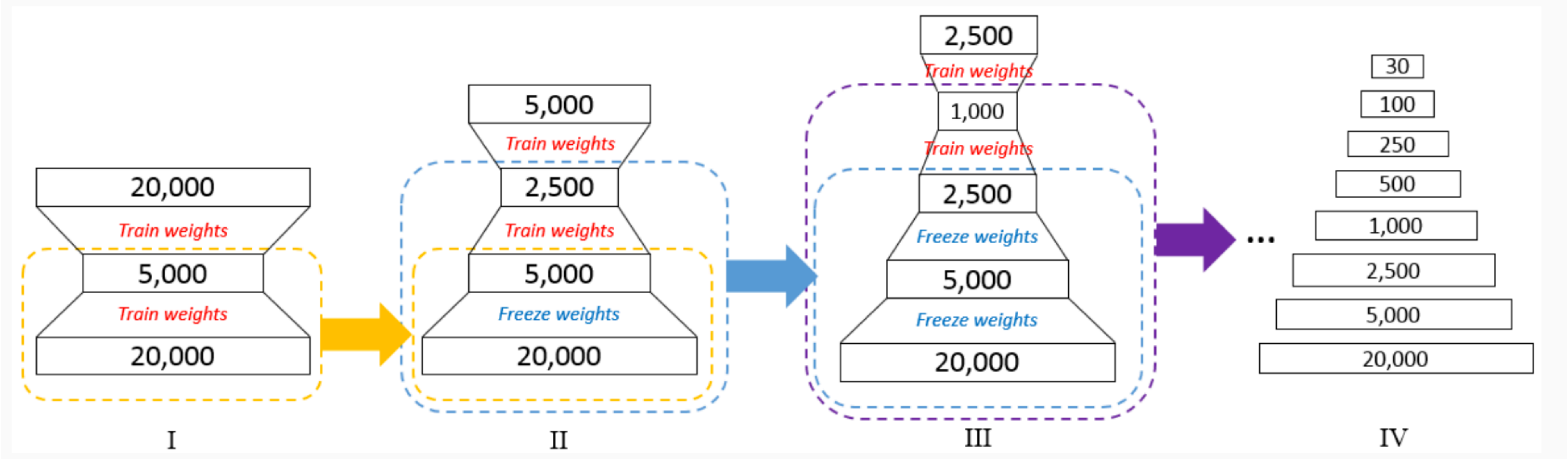
- Select the top 20000 highest frequency unigrams

- Convert each data to 20000-sized bit string by checking whether these 20000 unigrams is in it

Use DBN transfer vector to 30-dim

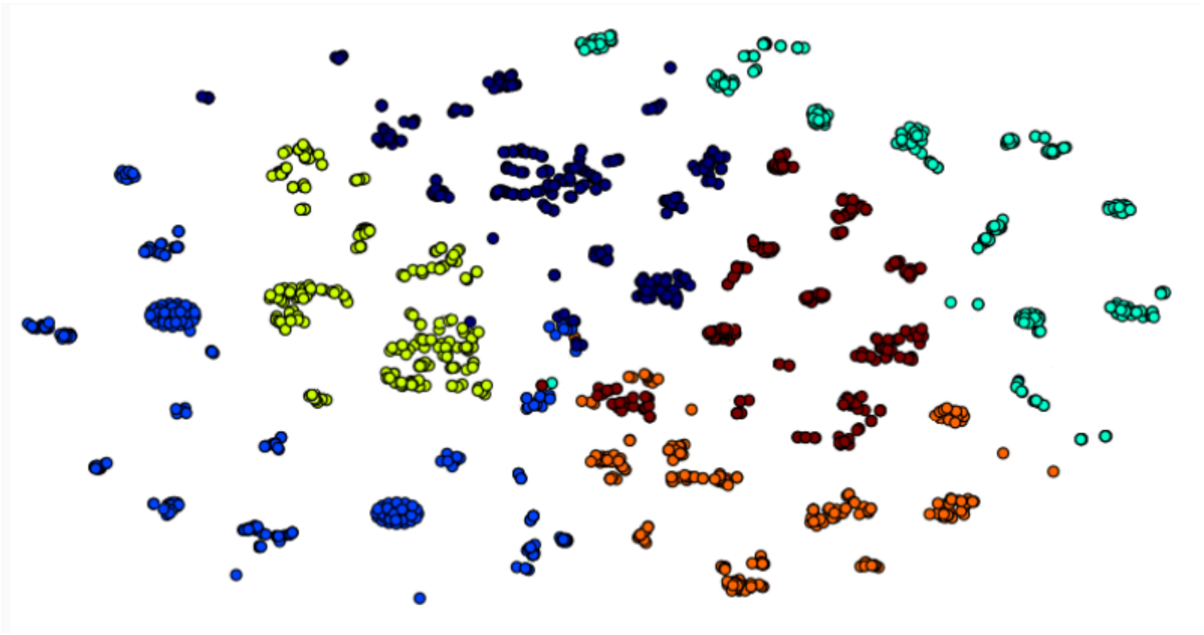
- In this case, DBN is implemented by deep stacked autoencoder with denoising
- The DBN input size is 20000 and output size is 30
- The output is called the signature of this malware

DBN training detail



batch size: 20   noise ratio: 0.2%   1000 training epochs

Classification



- Due to malware source code leaked, some version of one malware might have part of another one's code
- Use Kaspersky classify result as tag to train an SVM classifier

Conclusion

- They got 98.6% accuracy on test data, which is a relatively improvement to regular detection.