Malware family classification via efficient Huffman features

By:
Stephen O Shaughnessy (Technological University Dublin) and Frank Breitinger (University of Lausanne)

From the proceedings of
The Digital Forensic Research Conference
DFRWS USA 2021
July 12-15, 2021

DFRWS

Malware family classification via efficient Huffman features

Stephen O'Shaughnessy (TU Dublin)

Frank Breitinger (UNIL)

OLLSCOIL TEICNEOLAÍOCHTA BHAILE ÁTHA CLIATH
TU DUBLIN
TECHNOLOGICAL UNIVERSITY DUBLIN

UNIL | Université de Lausanne

DFRWS Background

- Malware is constantly on the rise.
- Classification is important to group according to similar traits, behaviours etc.
- Malware feature extraction approaches:
  - can be labour-intensive so may not scale well.
  - require knowledge of malware's internal binary structure.
- Feature selection crucial to classification performance.
  - *Malware analysts are not data scientists*
- Need for an "automated" solution.

DFRWS Compression

- Think `7-zip`, `Bzip`, `LZMA` etc.
- Encodes data to a reduced representation (fewer bits)
  - *reduced storage space or bandwidth for transmission*
- For our purposes, we can represent a full binary file with less data.
- Compression can be applied to data in any format – wide range of possible domains.

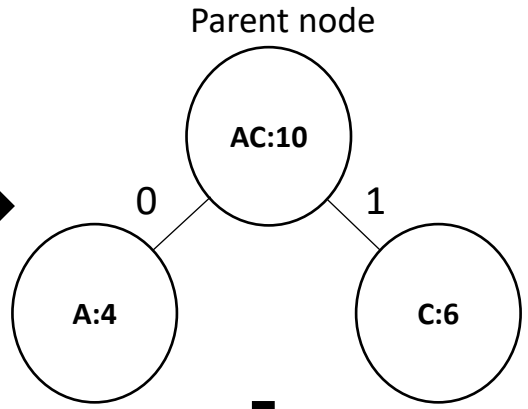
DFRWS Related Work

- **Distance Metrics**
  - *Normalized Compression Distance (NCD) (Cilibrasi & Vitanyi, 2005).*
  - *LZJD and SHWel (Raff & Nicholas, 2017)*
- **Feature space**
- *Dictionaries of substrings can be engineered as feature vectors (Sculley & Brodley, 2006).*
- *Applied to text classification problems (Paskov et al., 2013).*
- **Limitations:**
  - *NCD computationally inefficient*
  - *LZ variants feature space infeasibly large*

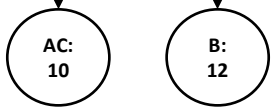
DFRWS Huffman Coding

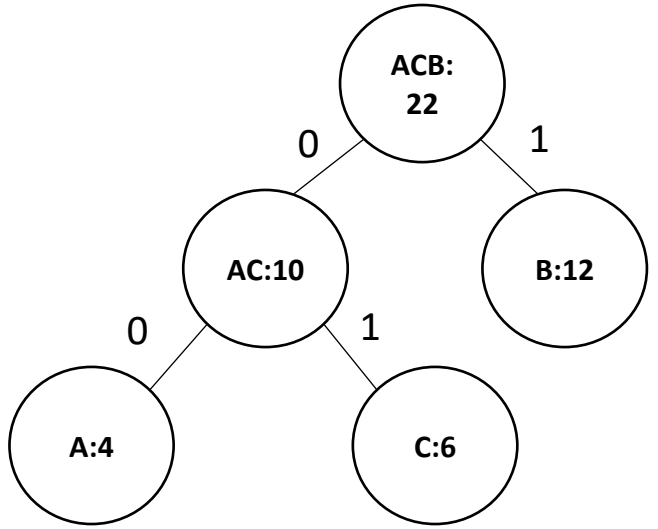
- Prefix codes generated from a set of symbol (character) frequencies.
- Example: input string: AAAABBBBBBBBBBBBBCCCCCC

Symbol	A	C	B
Frequency	4	6	12



Symbol	AC	B
Frequency	10	12





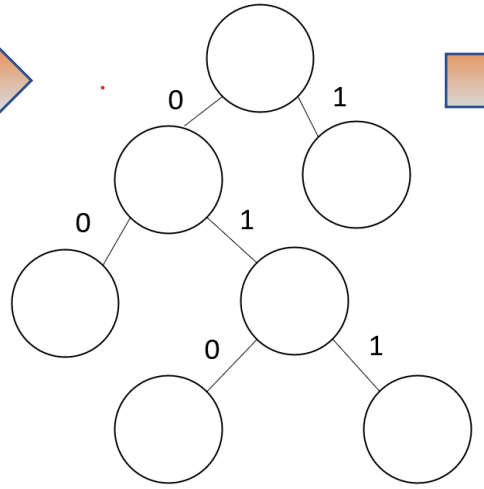
Symbol	A	C	B
Frequency	4	6	12
Codeword	00	01	1

DFRWS eHf algorithm

Malware sample

$h\ \{\ [v, [\ \kappa, c = \{\emptyset\}\ ]]\ \}$

$h = heap; v = frequency; \kappa = symbol, c = codeword$



Huffman tree

$sorted\ h\ \{[v, [\kappa, c]]\}$

$foreach\ i\ in\ h:$
$eHf_i = \kappa_i + v_i + int(c_i)$

$eHf\ feature\ vector = [69, 74, 79]$

Symbol converted to decimal:

- A = 65
- B = 66
- C = 67

Initial heap

[4, [65, Ø]]
[6, [67, Ø]]
[12, [66, Ø]]

Heap after processing

[22, [66, '1'],[65,'00'], [67, '01']]

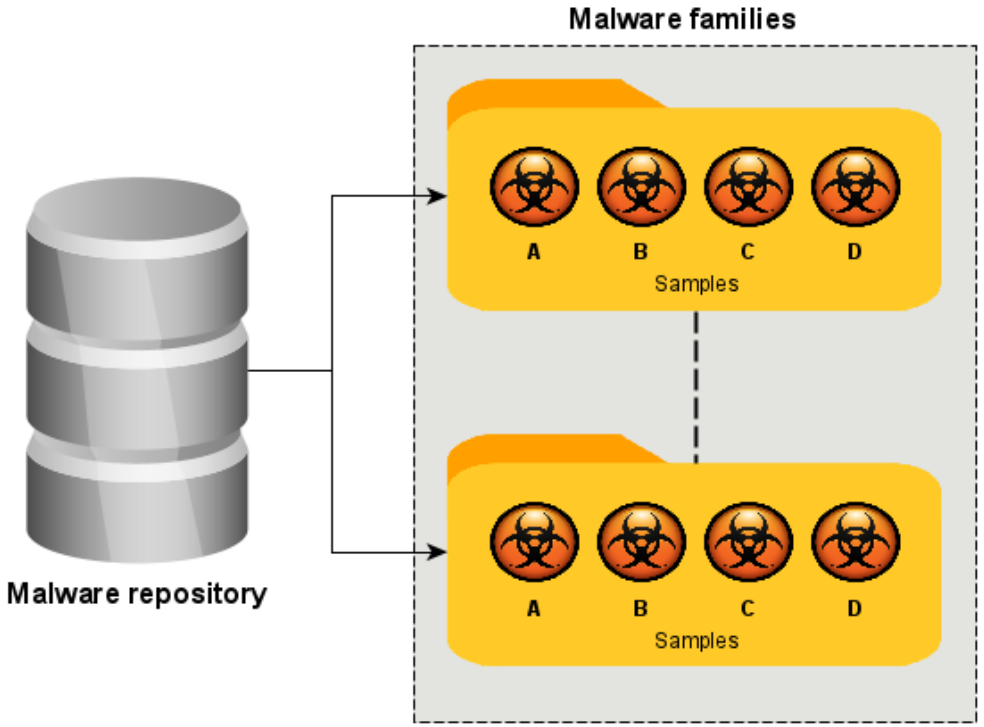
Note: eHf may produce variable length vectors. For classification, resize to k-smallest dimension vector.

OLLSCOIL TEICNEOLAÍOCHTA BHAILE ÁTHA CLIATH
DUBLIN
TECHNOLOGICAL UNIVERSITY DUBLIN

UNIL | Université de Lausanne

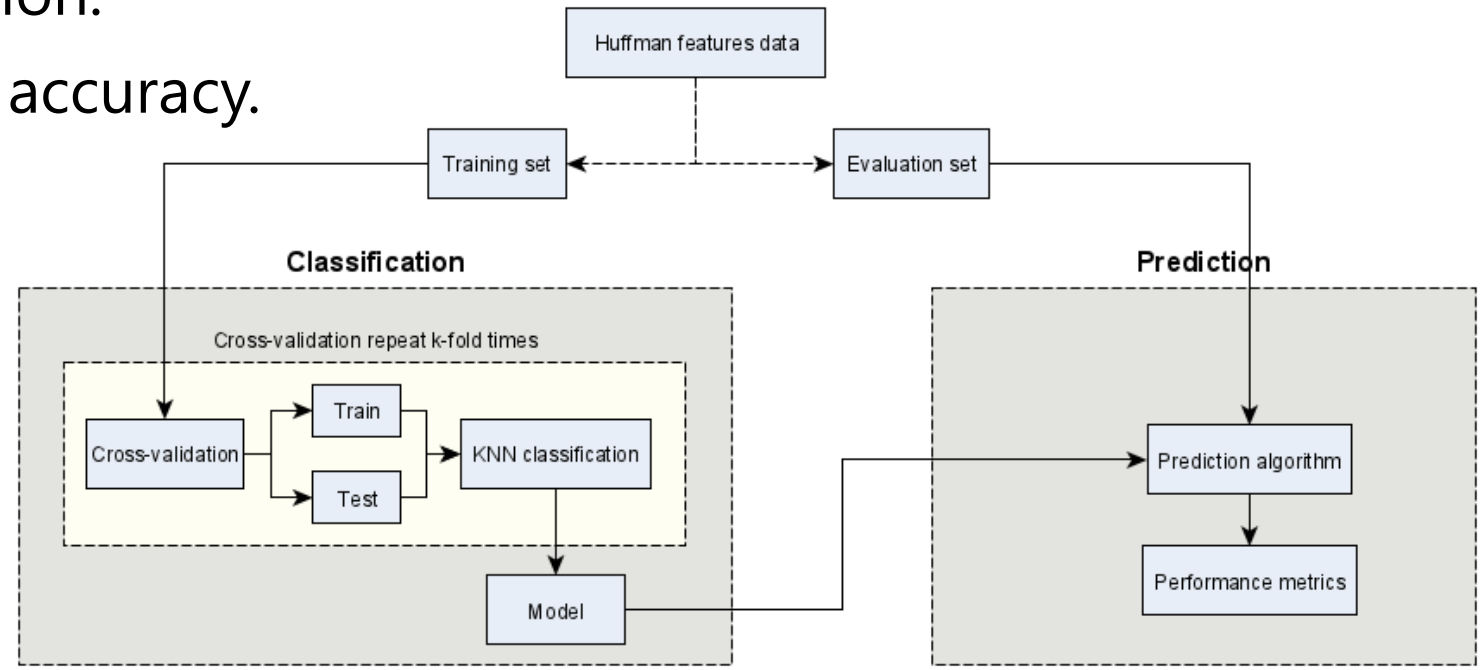
DFRWS Data

- Win32 portable executable files.
- **VirusTotal** academic share (~2018 – present).
- **AV Class Labeller** *(Sebastian et al., 2016).*
- Initial training dataset *8,232* from 12 families.
- Extended to *14,694* from 23 families.



DFRWS Classification

- Data and label sets split *90:10* for training and evaluation.
- k-nearest neighbour algorithm (*k = 3*, Minkowski).
- Parameter tuning using GridSearchCV.
- 5-fold stratified cross-validation.
- Metrics: precision, recall and accuracy.



DFRWS Results

Family	Precision	Recall	Accuracy
Agent.BDMJ	0.989	1.000	0.994
Autoit	0.961	0.976	0.969
Berbew	0.993	0.986	0.990
Dinwod	0.994	0.983	0.988
Dorkbot	0.977	0.988	0.982
Dridex	**1.000**	**1.000**	**1.000**
Oberal	0.976	1.000	0.988
Scar	0.857	0.854	0.855
Sfone	0.987	0.996	0.991
Socks	0.991	0.980	0.986
Sytro	0.994	0.999	0.997
Vilsel	0.985	0.971	0.978
Weighted avg.	**0.982**	**0.982**	**0.982**

- Initial dataset - *8,232* samples.
- 11 out of 12 class prediction true positive rates (TPR) of 97% or above.
- Scar family poorest performer – some mislabelling discovered on VT.
- 10% evaluation testing returned ~97% precision, recall and accuracy.

DFRWS Comparison with NCD

Dist. metric	Runtime (secs)	Prec.	Recall	Acc.
eHf-Jaccard	1.42×10^{-3}	0.969	0.968	0.968
eHf-Minkowski	$\mathbf{1.02 \times 10^{-3}}$	**0.972**	**0.973**	**0.972**
eHf-Euclidean	1.06×10^{-3}	0.970	0.971	0.969
NCD	1.2	0.782	0.774	0.772

- Compare `eHf` + standard distance metrics with NCD

DFRWS Comparison with LZJD[1]

	Run-time efficiency (ms)		Size
	feat. gen.	training	feat. dims.
LZJD-sh	5.15×10^{-1}	9.92×10^{-2}	1024
LZJD	1.21×10^{-1}	1.86×10^{-1}	1024
eHf	$\mathbf{5.0 \times 10^{-2}}$	$\mathbf{2.71 \times 10^{-2}}$	**229**

- Comparison of time complexities on larger dataset.

DFRWS Comparison with LZJD[2]

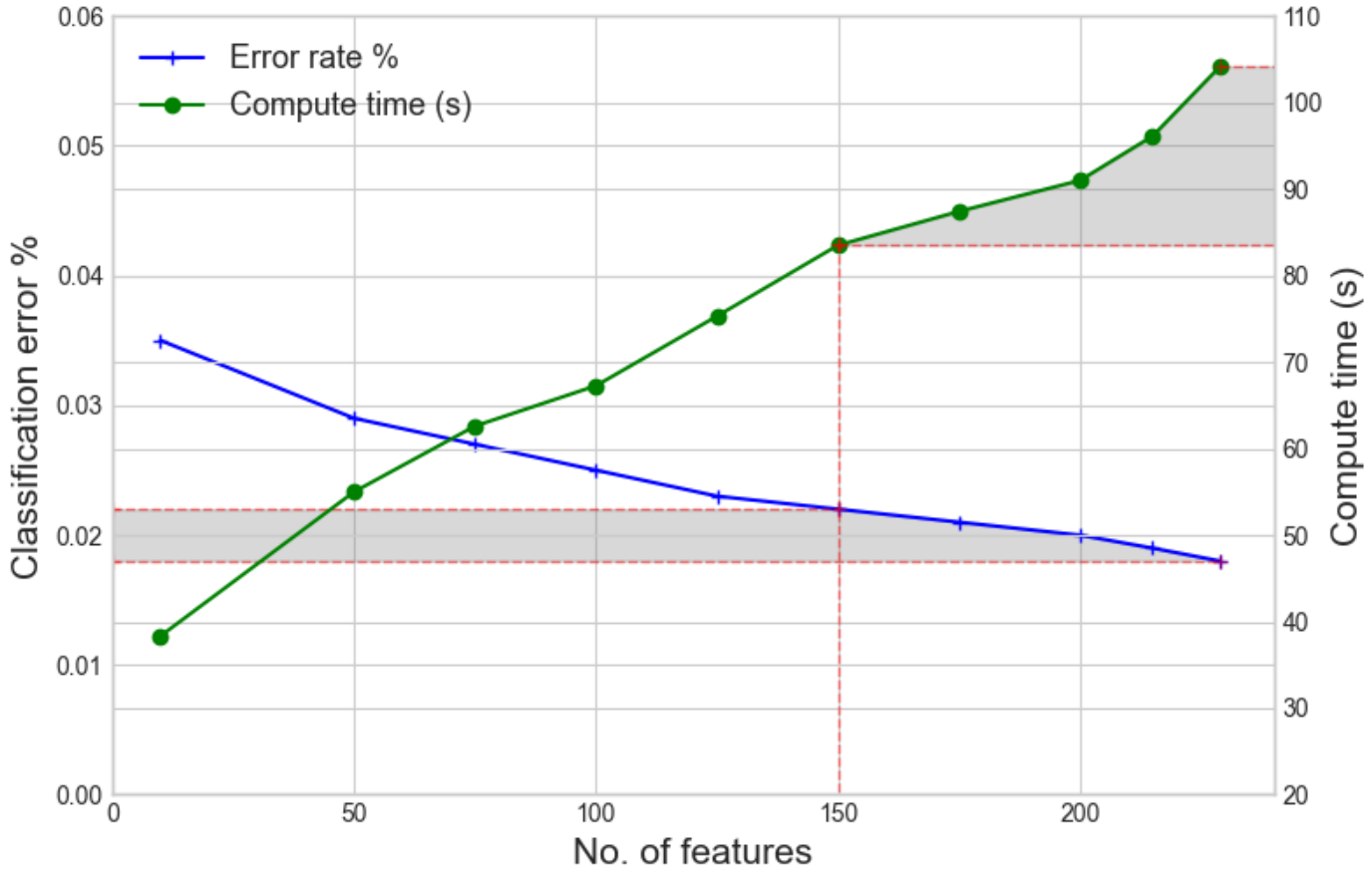
	Training			Validation		
	precision	recall	accuracy	precision	recall	accuracy
LZJD-sh	**0.977**	**0.973**	**0.974**	0.882	0.878	0.873
LZJD	0.951	0.950	0.950	0.745	0.752	0.746
eHf	0.972	0.973	0.972	**0.974**	**0.974**	**0.974**

- Comparison of classification performance.

DFRWS Code reordering obfuscation

- Import tables extracted from all samples in the VT dataset.
- KNN classifier trained as previously using the eHf import table vectors.
- `SCYTHE` tool used to reorder ~3k sub-sample import tables.
- KNN model: precision, recall and accuracy of 99.8%, 99.7% and 99.7% respectively.
- *Ordering of the input sequence is not a consideration in the generation of the codewords as data is stored according to frequency of symbols.*

DFRWS Feature optimization



- Features can be optimized.
- Reduction from 229 dimensions to 150 = 20% less compute time.
- Error rate increase of 0.4%.
- Dependent on data.

DFRWS Summary of contributions

- Novel method of representing binary features.
- Negates the need for invasive analysis techniques.
- Does not require intricate knowledge of binary structures.
- (Quite) Fast and scalable.
- FOSS
- Potential to apply to other domains.
- Outputs can be "plugged in" to other ML algorithms.

DFRWS Study limitations

- `eHf` developed in January 2021.
  - *Limited to malware executables and import table dumps.*
  - *Only shows results from KNN classifier.*
  - *Other forms of obfuscation not tested.*

DFRWS Future work

- Test at scale (Sorel 20M dataset)[1]
- Improve processing speeds.
- Other forensic scenarios.
- Further obfuscation testing.

[1] https://github.com/sophos-ai/SOREL-20M

DFRWS Thank you!

TU DUBLIN
OLLSCOIL TEICNEOLAÍOCHTA BHAILE ÁTHA CLIATH
TECHNOLOGICAL UNIVERSITY DUBLIN

UNIL | Université de Lausanne