Feature-based Type Identification of Computer Data

Mohsen Toorani

Selmer Center, Department of Informatics, University of Bergen

FRISC Winter School at Finse

May 2012

A joint work with M.C. Amirani and S. Mihandoost

Why Type Identification?

Type identification of computer data is a building block in:

- Operating systems
- Firewalls
- Intrusion Detection Systems (IDS)
- Virus scanning and malware detection
- Analyzing networks traffics
- Filtering email attachments
- Digital Forensics and Digital Investigation
- Steganalysis detectors
- Any other application concerning computer files and computer security …

File Type Detection Methods

1. Extension-based: Windows OS
2. Magic bytes-based: Unix-based OS
3. Content-based

File Type Detection Methods…

1. Extension-based method

- It is the fastest, easiest, and most common method of file type identification.
- At least in windows-based systems, all file types are generally accompanied by an extension.
- It is applicable to both binary and text files.
- No need for opening and reading contents of files.
- It can be easily spoofed, even by a child.

File Type Detection Methods…

2. Magic bytes-based method

What are the magic bytes?

- The magic bytes are some predefined signatures in the header or trailer of binary files.

File Type	Header Magic Bytes	Footer Magic Bytes
RTF	"{\rtf1\"	"}}"
PDF	"%PDF-<version>"	"%%EOF" plus optional CR/LF
JPG	FF D8	None
GIF	"GIF87a" or "GIF89a"	None
PNG	89 50 4E 47 0D 0A 1A 0A	None
WAV	"RIFF" plus "WAVE" at offset 0x08	None
ZIP	"PK"	None
EXE/DLL/SCR etc.	4D 5A	None

File Type Detection Methods…

2. Magic bytes-based method (Cont.)

- Only applicable to the binary files.
- Some binary files do not have magic bytes.
- There is not any worldwide standard for magic bytes.
- Available references do not provide the same information on magic bytes.
- Lengths of magic bytes varies for different file types.
- Spoofing is feasible and needs a little technical knowledge.

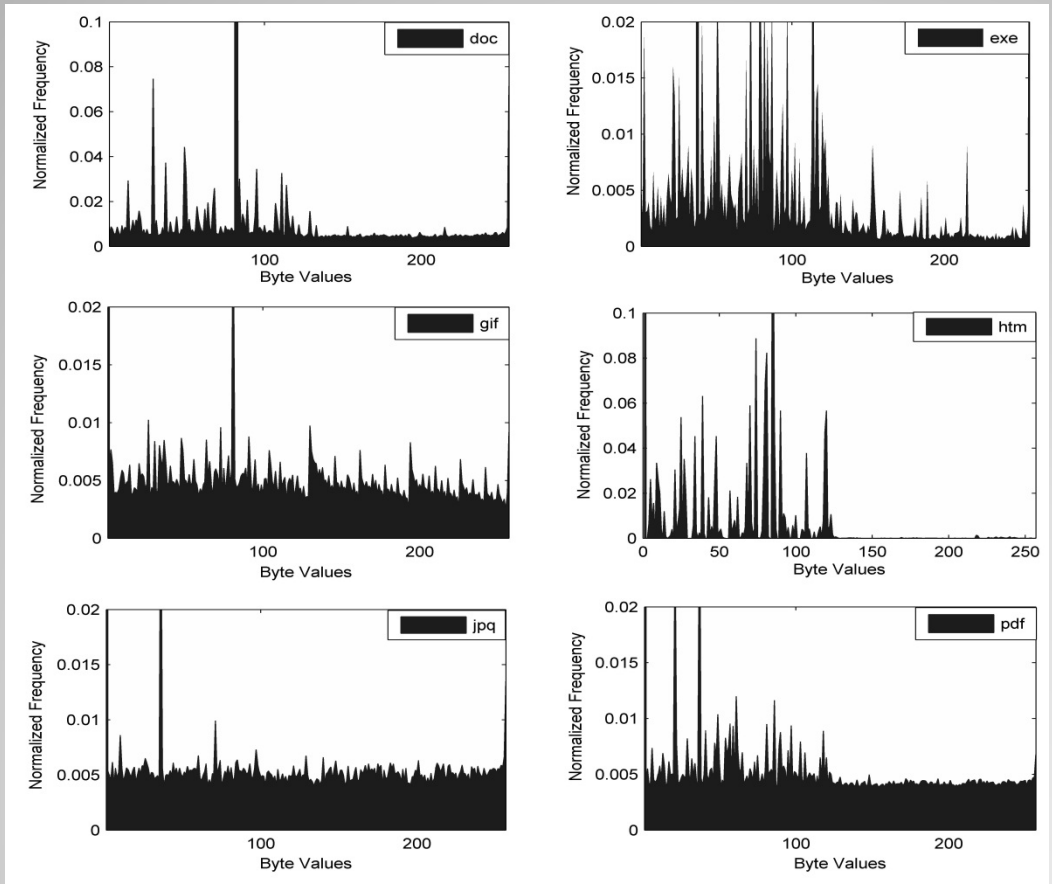
File Type Detection Methods…

3. Content-based method

- Presented for the first time in 2003
- Slower than two previous methods
- Subject to further research for increasing speed and accuracy
- Based on file contents and their BFD (*Byte Frequency Distribution*)
- Uses statistical modeling or feature-extraction techniques

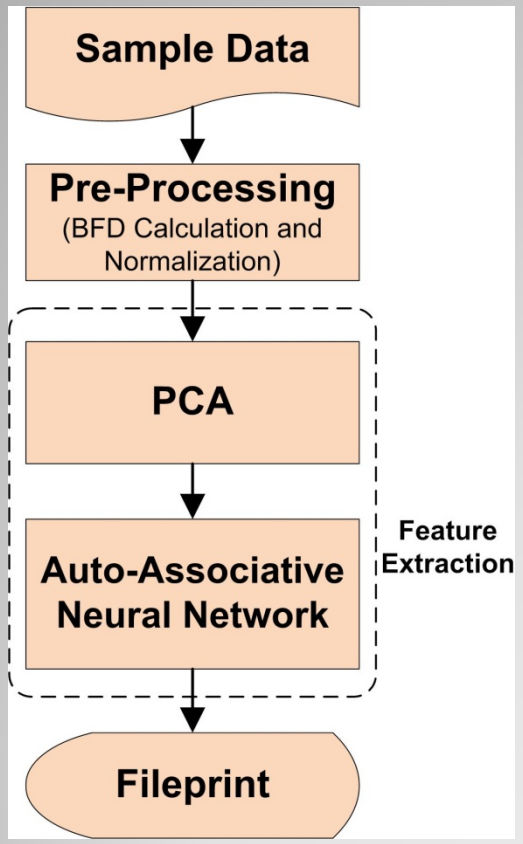
File Type Detection Methods…

3. Content-based method

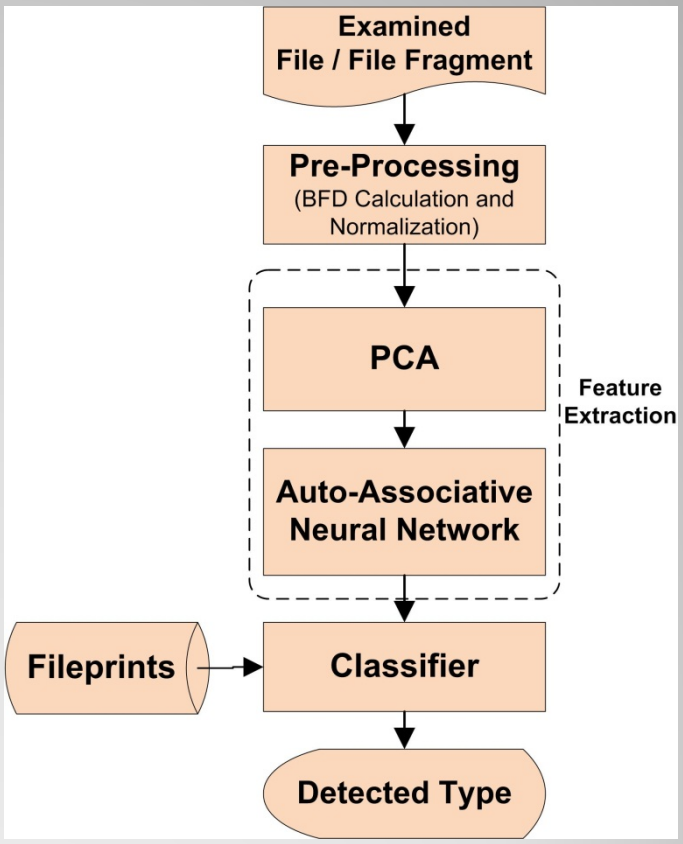


BFD of some common file types

The Proposed Method

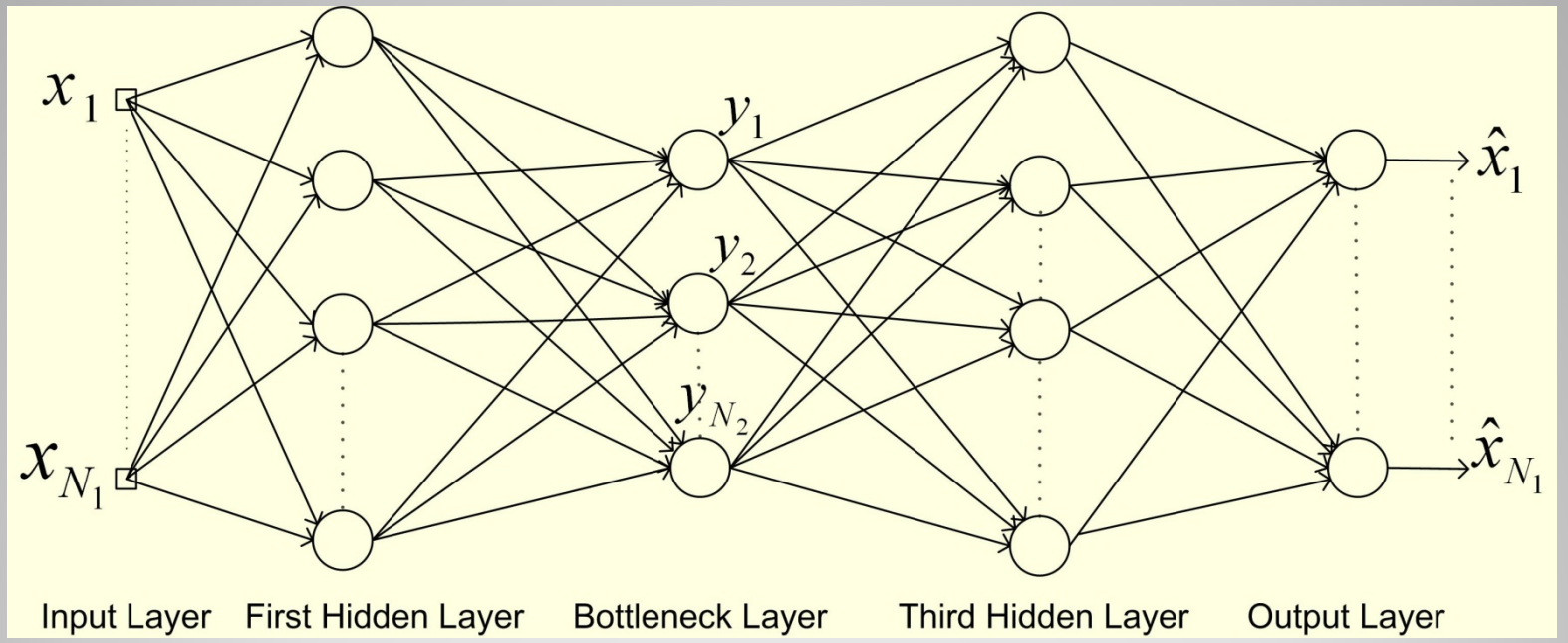


Training Phase



Detection Phase

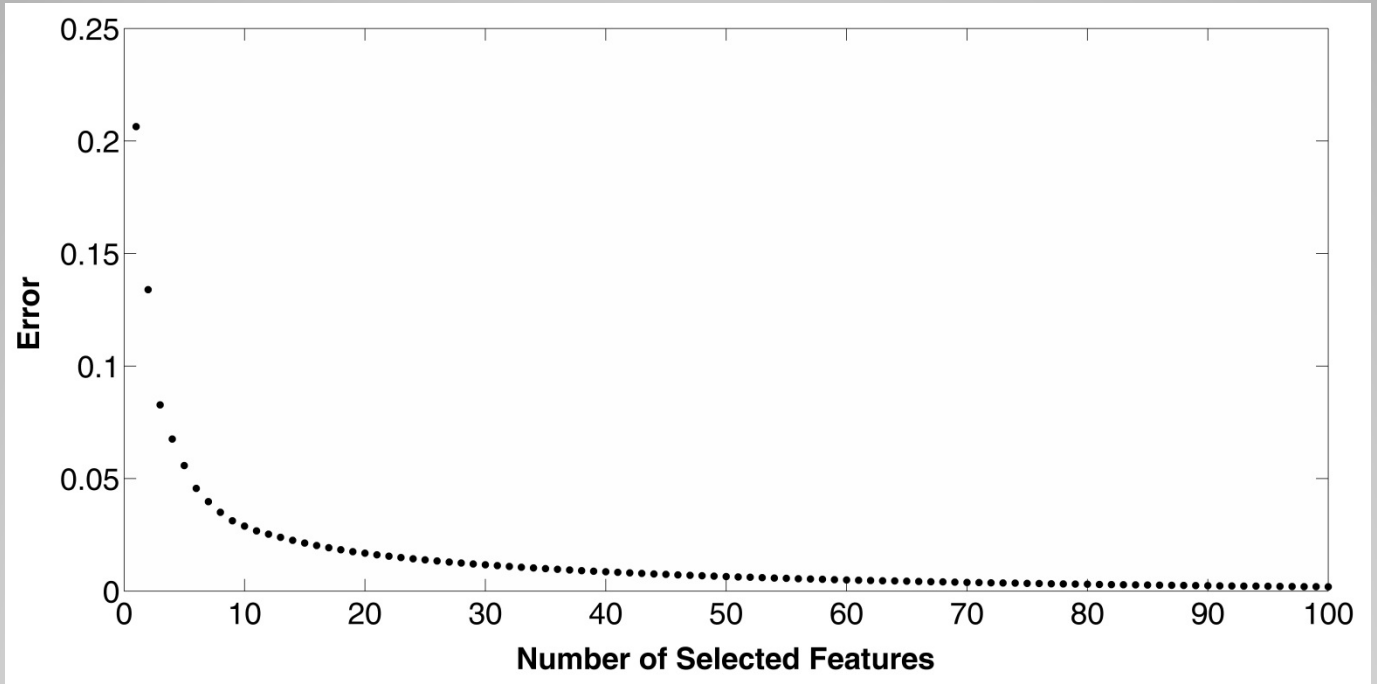
(The Proposed Method) Auto-associative Neural Network



The deployed 5-layer auto-associative neural network

Values of N_1 and N_2 are determined experimentally. The aim is minimize the introduced error of dimensionality reduction.

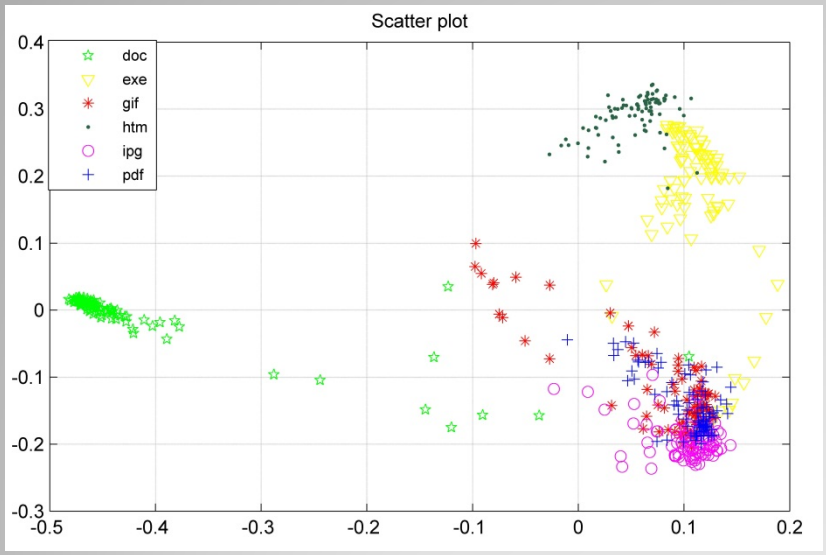
Experimental Results
(Value of N_1)



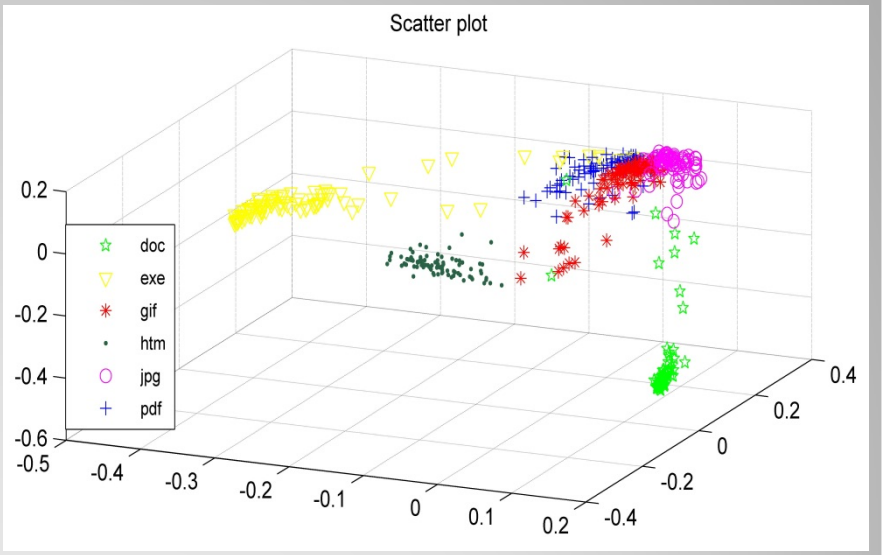
Introduced error of the PCA: $E_k = \frac{1}{2} \sum_{i=k+1}^{256} \lambda_i$

N_1=60 seems to be a good trade-off between accuracy and speed.

Experimental Results (Value of N_2)



Two-dimensional Scatter Plot (N_2=2)



Three-dimensional Scatter Plot (N_2=3)

With a trial-and-error approach, we selected N_2=15.

Experimental Results (First experiments)

- The test files were collected from the Internet by a general search on the *Google* search engine.
- 120 files of each type were randomly collected. We used 90 files out of them for training and the remained 30 files for testing the results.

Type of sample files	Maximum Size (Bytes)	Minimum Size (Bytes)
doc	6906880	15360
exe	24265736	882
gif	298235	43
htm	705230	1866
jpg	946098	481
pdf	10397799	12280

Experimental Results
(First experiments)

The resulted confusion matrix for 180 examined files of 6 types

	doc	exe	gif	htm	jpg	pdf
doc	30	0	0	0	0	0
exe	0	28	0	0	0	0
gif	0	0	29	0	0	0
htm	0	0	0	30	0	0
jpg	0	0	0	0	30	0
pdf	0	2	1	0	0	30

Total correct classification rate = 98.33%

(Considering the whole contents of files)

Experimental Results (Second experiments)

- The test files were collected from the Internet by a general search on the *Google* search engine.
- 200 files of each type were randomly collected. We used half of them for training and the remaining half for testing the results.

Type of sample files	Number of files	Minimum Size (Bytes)	Maximum Size (Bytes)	Average Size (Bytes)
doc	200	3 494	6 906 880	267 830
exe	200	1 884	21 715 672	7 376 930
gif	200	9 817	4 123 106	79 368
htm	200	232	418 819	42 268
jpg	200	3 717	6 674 957	395 386
pdf	200	12 231	9 025 199	734 380

Experimental Results
(Second experiments)

Summarized accuracies and Running-times for 100 examined data of each type

Status of data	Classifier	doc	exe	gif	htm	jpg	pdf	CCR	Running-time (Second)
Whole file contents	MLP	100	97	99	100	99	97	98.67	0.047
Whole file contents	SVM	100	98	100	100	99	98	99.16	0.032
File fragments of 1500 bytes length	MLP	88	83	78	95	74	85	83.83	0.013
File fragments of 1500 bytes length	SVM	89	85	80	95	75	89	85.50	0.009
File fragments of 1000 bytes length	MLP	83	80	72	90	71	84	80	0.007
File fragments of 1000 bytes length	SVM	85	81	76	91	73	86	82	0.006

Contributors	Approach	Header-dependent	Size Specific	# File/Data types	# Total Samples	Accuracy (%)	Running-time (Seconds)
McDaniel and Heydari (2003)	File	Yes	No	30	120	27.5 (BFA)	0.010
				30	120	45.83 (BFC)	1.19
				30	120	95.83 (FHT)	0.015
Li et al. (2005)	File	Yes	Yes	8 (5 classes)	800	82 (One-Centroid)	NA
						89.5 (Multi-Centroid)	
						93.8 (Exempler files)	
Dunham et al. (2005)	File	Yes	No	10	760	91.3	NA
Karresand and Shahmehri (2006)	File Fragment	No	No	49	53	97.9 (jpg)	NA
Karresand and Shahmehri (2006)	File Fragment	No	No	51	57	87.3 - 92.1 (jpg)	1.20 - 2.50
						46 - 84 (zip)	
						12.6 (exe)	
Zhang et al. (2007)	File Fragment	No	Yes	2	100	92.5	NA
Moody and Erbacher (2008)	File Fragment	No	No	8	200	74.2	NA
Calhoun and Coles (2008)	File Fragment	No	No	2	100	68.3-88.3 (bytes 129-1024)	NA
						60.3-86 (bytes 513-1024)	
Our previous work (2008)	File	No	No	6	720	98.33	NA
Cao et al. (2010)	File	No	No	4	1000	90.34	NA
Ahmed et al. (2010)	File	No	No	10	2000	90.19	NA
Ahmed et al. (2011)	File & File Fragment	No	No	10	5000	90.5 (using 40% of features)	0.077 (MP3 file)
						88.45 (using 20% of features)	0.007 (exe file)
Our recent work (2012)	File	No	No	6	1200	99.16	0.032
	File Fragment					85.5 (1500 bytes Fragments)	0.009
						82 (1000 bytes Fragments)	0.006

Conclusions

- The proposed content-based method can be used for type detection of computer files, file fragments, and data packets.
- 15 major features are automatically extracted from BFD of the sample data by using PCA and auto-associative neural network.
- We examined both SVM and MLP classifiers, but the SVM classifier provides better speed and accuracy.
- With the SVM classifiers, we obtained an average correct classification rate of 99.16%, 85.5%, and 82% when considering the complete file, file fragments of 1500 bytes lengths, and file fragments of 1000 bytes lengths, respectively.
- The results are significant in comparison with those of other literature.

Thanks

Questions?