A New Approach to Content-based File Type Detection

Presented at the 13th IEEE Symposium on Computers and Communications (ISCC'08)

by:

Mohsen Toorani

(ResearcherID: A-9528-2009)

Why the File Type Detection?

File type detection has the most usage in proper functionality of:

- Operating systems
- Firewalls
- Intrusion Detection Systems (IDS)
- Virus scanning and malware detection
- Filtering email attachments
- Steganalysis detectors
- Analyzing networks traffics
- Computer forensics
- Any other application regarding 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 detection

- 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 the contents of files.
- It can be easily spoofed, even by a child.

File Type Detection Methods…

2. Magic bytes-based detection

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…

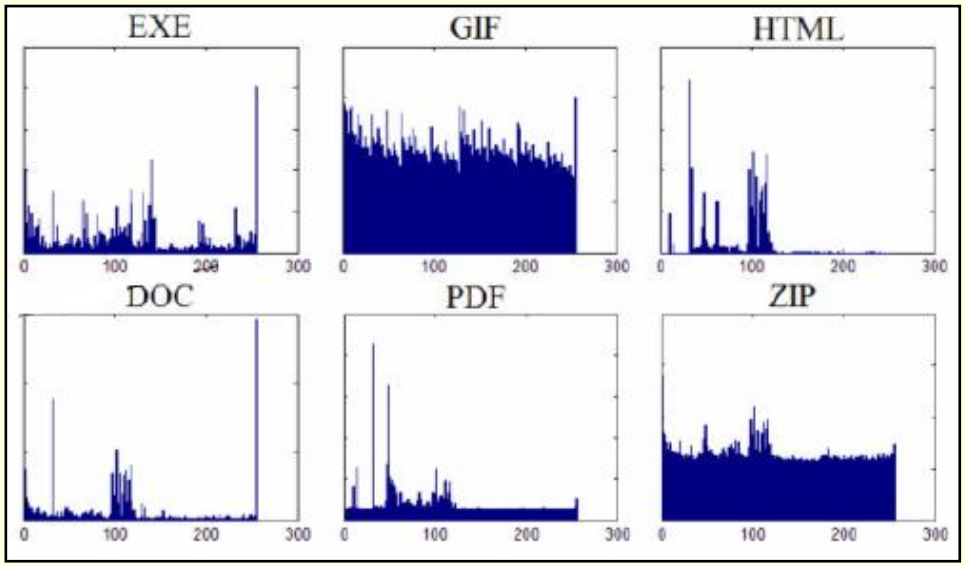
2. Magic bytes-based detection (Cont.)

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

File Type Detection Methods…

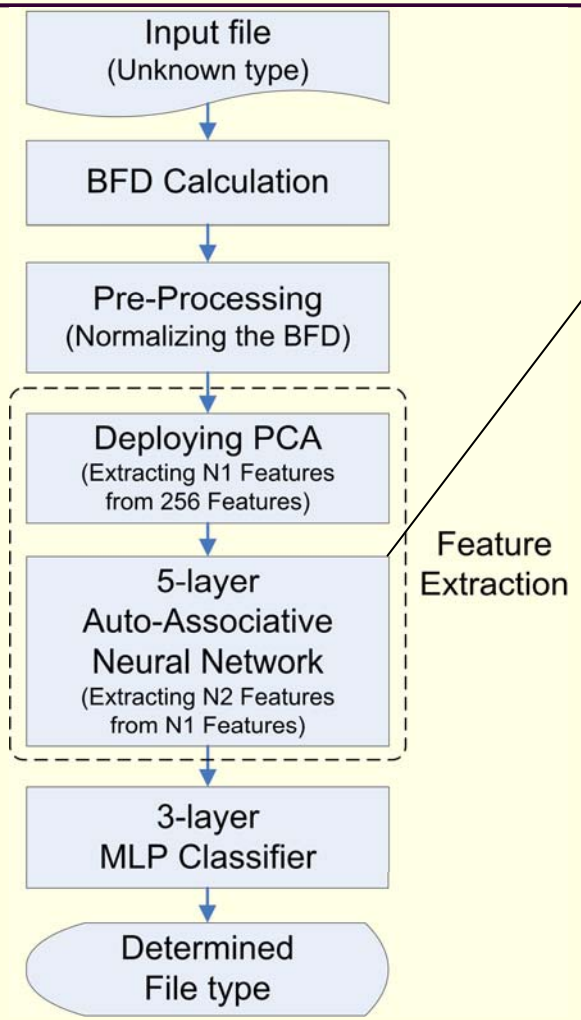
3. Content-based detection

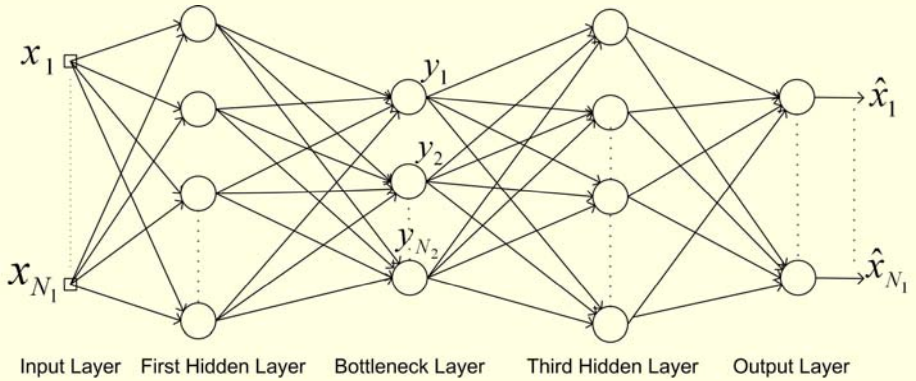
- Subject to further researches
- Slower than two previously-mentioned methods
- Based on file contents and its BFD (*Byte Frequency Distribution*), and uses statistical modeling techniques



BFD of some common file types

Our Proposed Method





The deployed 5-layer auto-associative neural network

- In the training phase, the outputs are taken from the *output layer* (5th layer of the above figure).
- In the detection phase, the outputs are taken from the *bottleneck layer* (3rd layer of the above figure).
- Values of N_1 and N_2 will be experimentally determined so that the introduced error of dimensionality reduction is minimized.

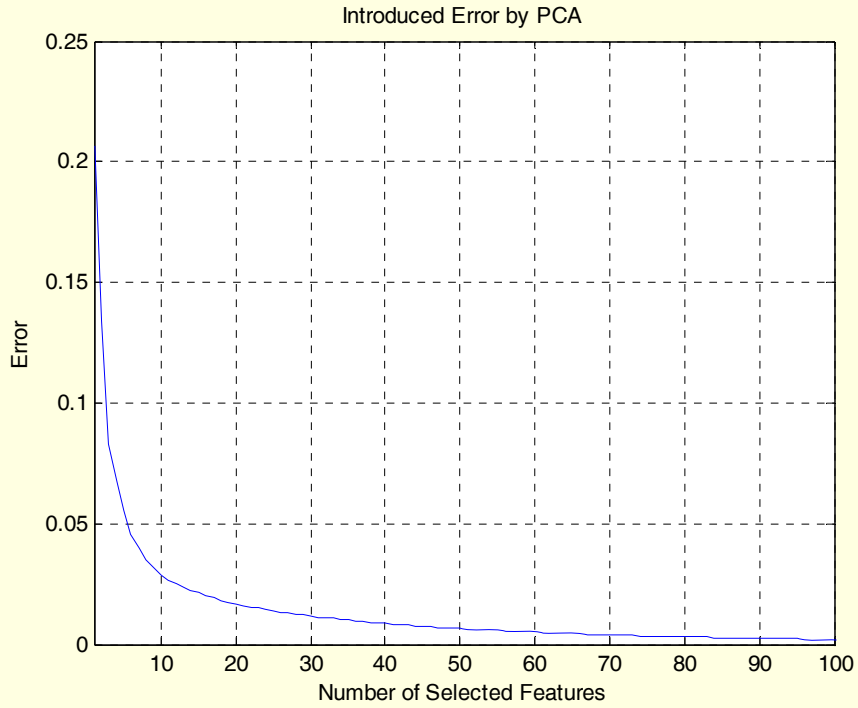
Experimental Results
(On the selected sample files)

- 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.
- Unlike other literatures, our experiments were not size specific.

Size variation among the 120 sample files of each type (Proofs on not being size-specific)

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
(On the selected value of N_1)



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

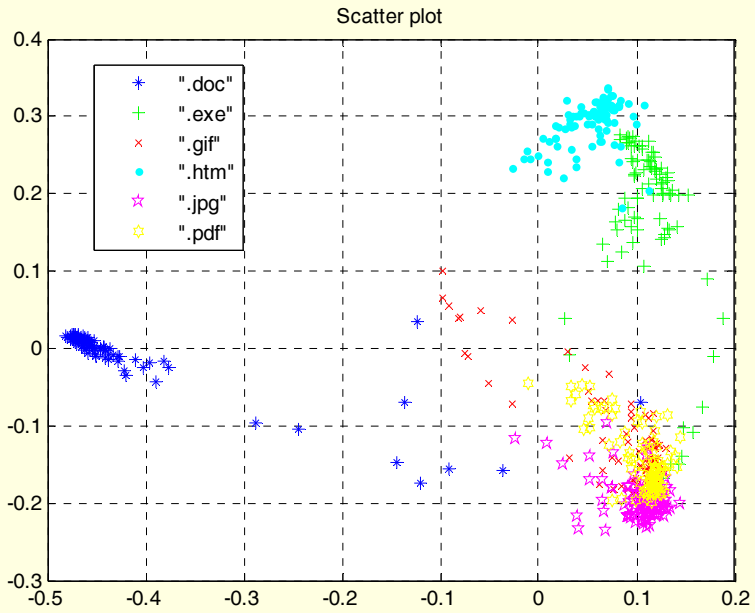
(*k* is the number of selected features)

The above diagram is obtained from our set of training files.

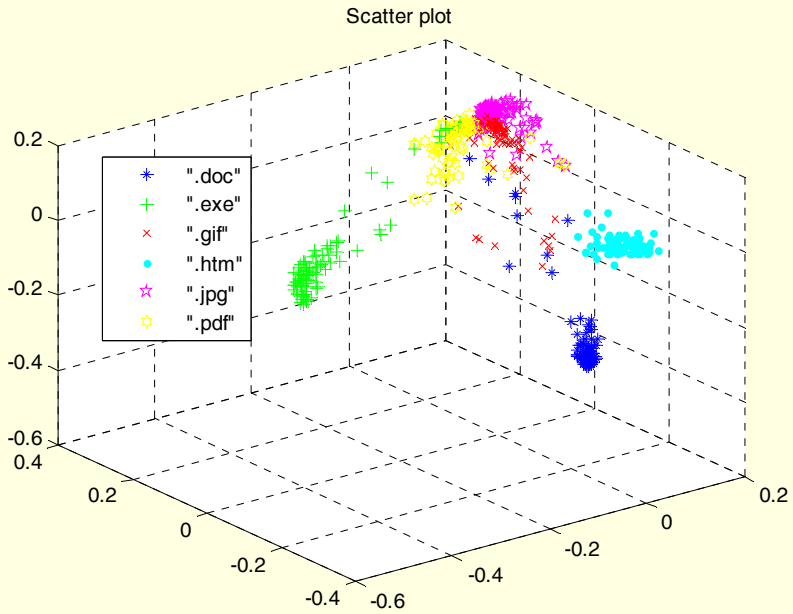
N_1=60 seems to be a a good trade-off between accuracy and speed.

Experimental Results
(On the selected value of N_2)

With a try-and-error experiment, we reached to N_2=15 as an optimum choice. This means that our automatic feature extraction system finally extracts 15 features from each examined file type.



Two Dimensional Scatter Plot
N_2=2



Three Dimensional Scatter Plot
N_2=3

Experimental Results (Cont.)

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 (Cont.)

A comparison with the other content-based approaches

Method	Total Correct Classification Rate
McDaniel and Heydari (2003)	27.5%
Li et. al (2005)	82%
Karrasand and Shahmehri (2006)	92.1%
Our approach	98.33%

Conclusions

- A new content-based file type detection method was introduced that uses PCA and unsupervised neural networks for the automatic feature extraction.
- It is completely header-independent and uses the whole contents of files. It does not depend on the positions of data fragments of files and can detect the file type even if the file header is changed or corrupted.
- The proposed method has a very good accuracy and is fast enough to be used in the real-time applications.
- A total correct classification rate of 98.33% is obtained when considering the whole contents of files, and without being file size specific.

Future works

The proposed method can be optimized by taking several improvements:

- The accuracy can be improved by taking the multi-centroid models when dealing with the huge number of file types.
- Truncation can also improve the accuracy and the speed but it can make the method header-dependent.
- A file size categorization can also improve the accuracy.

We are considering such improvements in our next paper.

Thanks

Thank you for your attention!