Name: Burman Noviansyah
AndrewID: bnovians
Program: MSISPM '14

Thesis

95-765 Information Security Project

(version: Final)

"Increasing Security Awareness in Enterprise Using Automated Feature Extraction with Long n-gram Analysis for File Type Identification"

Submitted in partial fulfillment of the requirements for the degree of
Master of Science in Information Security Policy and Management

Burman Noviansyah
MSISPM Class of 2014

Pittsburgh, Pennsylvania, US
2013

Carnegie Mellon University
Heinz School (MSISPM)

Approved and recommended for acceptance as a thesis in partial fulfillment of the requirements for the degree of

Master of Science in Information Security Policy and Management

Title: *Increasing Security Awareness in Enterprise Using Automated Feature Extraction with Long n-gram Analysis for File Type Identification*
Project Members: Burman Noviansyah

Theses Advisor 1:	Timothy J. Shimeall **Print Name**	Software Engineering Institute **CMU Dept/Company**
	[redacted] **e-mail**	4500 Fifth Avenue Pittsburgh, PA 15213-2612 **Address**
	_____ **Signature**	December 19, 2013 **Date**
Theses Advisor 2:	Sidney Lloyd Faber **Print Name**	Software Engineering Institute **CMU Dept/Company**
	[redacted] **e-mail**	4500 Fifth Avenue Pittsburgh, PA 15213-2612 **Address**
	_____ **Signature**	December 19, 2013 **Date**
MSISPM Program Director:	Sean R Beggs **Print Name**	
	_____ **Signature**	**Date**

Table of Content

Table of Figures

Table of Tables

Abstract

File Type Identification (FTI) is a challenging task for IT department in Enterprise. In order to prevent confidential data leak, Information Security Team in IT department wants to recognize the true file type for each file leaving from or coming to their network. Previous works show great techniques and algorithms on how to identify the true file type. These include file extension, magic numbers within the specific part of file, statistical and machine learning approach. This Master Thesis proposes a combination technique from several applications and also suggests an improvement of algorithm and selection of training data set from previous works using automated feature extraction of long n-gram.

Keywords: file type identification; automated feature extraction; n-gram analysis

Acknowledgements

All praise is due to God, May He be Glorified and Exalted, Lord of the worlds with His Entirely Merciful, the Especially Merciful for guiding me to the straight path of those upon whom He has bestowed favor. His never-ending presence always reminds me to strive to be a great leader for my family, a best father of my children, a devoted son of my parents and a person that can be counted on by my community.

My beloved wife, Emilda, who always there to supports me for everything we planned before and while we are here in Pittsburgh. Your dedication as a wife and mother of our children cannot be expressed in words. It takes a great woman to do that, and I am grateful to have you beside me.

My lovely children, Amanda and Raihan, who always barge into my room just want to tell that I still have them here while I was sinking into stacks of papers and books.

My family in Pulo Gadung and PHDM, Kakek, Nenek, and Ungku, for your support and prayers.

Central Bank of Indonesia for the scholarship and all of support for me while attending CMU Heinz College.

Timothy Shimeall and Sidney Faber from CMU SEI's CERT Cyber Threat and Vulnerability Analysis. Thank you for approving my master thesis topics and supervised my works. I am very grateful and appreciate your support and your expertise while doing this thesis.

Thank you for Heinz College staffs and members for all helps for these two wonderful years. My friends in MSISPM 2014 program that shares idea, insight, and great discussion in Information Security area.

Last but not least, Intercultural Communication Center and Global Communication Center at CMU for workshops and tutors that help me to finish this thesis.

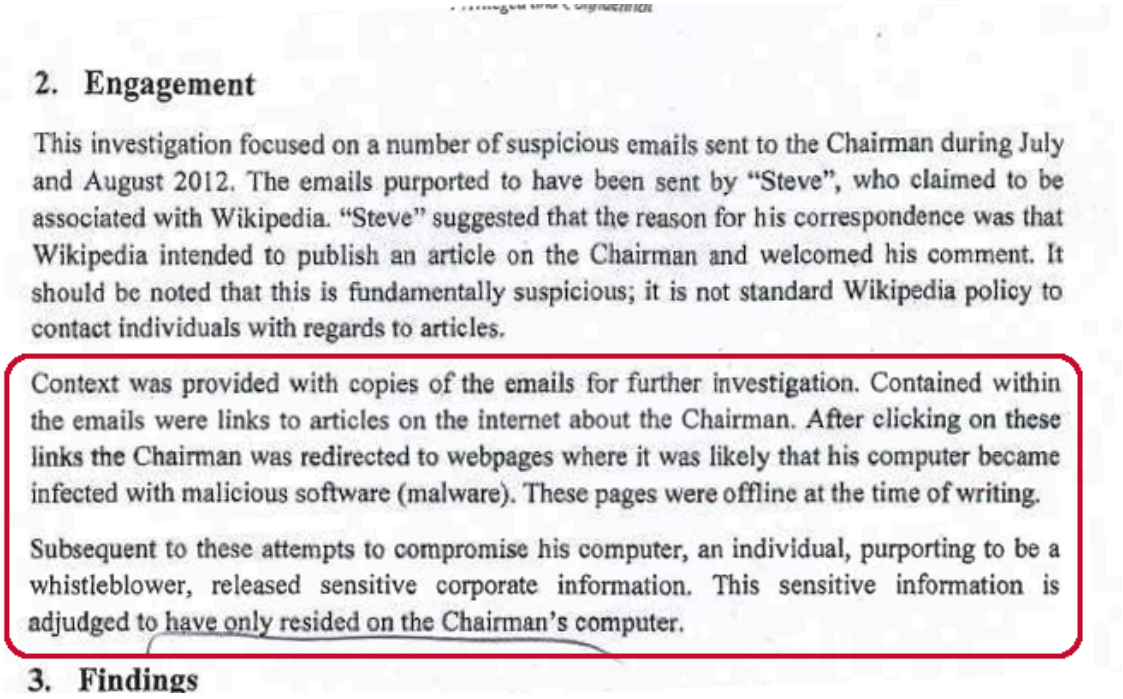
Chapter 1: Introduction

This chapter discusses the background of problem on how determining true file type along with its threat to Enterprise. In this chapter also discusses the scope of work in this thesis also the organization about this document.

1.1 Background

In Enterprise-scale Company, competition relies on the works and researches. The result of these sometimes become the thin line that can determine whether a company still be able to stay in its business or not. But sometimes, Company does not realize the magnitude of the insider threats and spy industries. The adversaries could steal or make the research papers left the Company without being detected. Recent technology such as Data Loss Prevention (DLP) was believed could stop this threat. But certain products on this technology only rely on the specific properties of files, and can be relatively easy to manipulate that.

In August 2012, one enterprise in Indonesia focusing in Coal Energy, Bumi PLC, became a victim of malware that extracted the confidential documents that only reside on the Chairman's computer. This documents affected company valuation dramatically. Bumi PLC appointed Context Information Security, a security consultant company based in London, UK. In Project MASON: Incident Response by (McKenzie), the preliminary investigation concluded that the Chairman clicked the link from malicious email and installed a malware from the link. There were no reports or notification in Bumi PLC network towards this extraction activity. A snippet document relates with this report can be observed Figure 1 below.



2. Engagement

This investigation focused on a number of suspicious emails sent to the Chairman during July and August 2012. The emails purported to have been sent by "Steve", who claimed to be associated with Wikipedia. "Steve" suggested that the reason for his correspondence was that Wikipedia intended to publish an article on the Chairman and welcomed his comment. It should be noted that this is fundamentally suspicious; it is not standard Wikipedia policy to contact individuals with regards to articles.

Context was provided with copies of the emails for further investigation. Contained within the emails were links to articles on the internet about the Chairman. After clicking on these links the Chairman was redirected to webpages where it was likely that his computer became infected with malicious software (malware). These pages were offline at the time of writing.

Subsequent to these attempts to compromise his computer, an individual, purporting to be a whistleblower, released sensitive corporate information. This sensitive information is adjudged to have only resided on the Chairman's computer.

3. Findings

Figure 1 Incident Response in (McKenzie) case

Detecting the data leak just by recognizing the file naming pattern can be simply deceived by changing the file name or even the extension itself. Specific black list keywords on file naming pattern such as "confidential", or "research", cannot prevent the file from leaving the company if the file is renamed into something different. A more sophisticated evasion technique is by manipulating the file header. A picture file format (e.g. JPG, BMP, etc.) can be recognized by Operating System using its file extension and also its file header and footer. An evasion technique to manipulating this metadata is a little bit hard to do, but still do able. If an enterprise will block all files that is recognized as Document file, one adversary can change this metadata into a picture file such JPG but the file body remains intact. Most likely, knowledgeable technical employees can do this manipulation. On (Lewellen, Moore and Cappelli), it is said that technical employees were more likely committed IT sabotage and also Intellectual Property Theft. With a sufficient knowledge in IT, this

type of employee could be an insider threat for the enterprise. Enterprise needs proper and practical technique to detect the real file type in order to prevent files leaving the company. The same technique can also be applied to prevent unwanted file such malware going inside the network company. It is common nowadays to see malware in mail attachment use extension of JPG but it is actually executable file.

1.2 Problems in identifying file type

Determining the true file type is a challenging task. An IT Department of Enterprise might want to know what is the real file type of any given file. The purpose of this identification process could be a part of their policy to block any restricted file types from leaving their network. This restricted file type might contain a confidential data, including the Intellectual Property Rights from Research and Development Department.

There are already various implementations on File Type Identification (FTI) task. Most of them are already embedded or bundled on Operating System functionality. Operating System such Microsoft Windows recognized the file type using file extension. With this file extension, Microsoft Windows internal system associates it with a proper application to handle this file type. Users can see the icon that is associated with a particular file extension. This is a simple way to perform the FTI task. Unfortunately, the accuracy of this technique is very low. Even in an unintentional way, a user can change the file extension and makes the Operating System wrongfully recognized the file type. For example, one original file has .PDF extension and by default it will be recognized by Microsoft Windows as a PDF document that is associated with Adobe Reader application to handle it. When user changes the file extension from .PDF to .JPG, Operating System will recognize this PDF document file as JPG image file. Hence, a default application named "Windows

Picture and Fax Viewer" is launched but gives the error message since this application cannot understand the file format.

To increase the accuracy of FTI task, researchers see what is inside the file itself. A practical implementation is observing specific part of files and then matching it to a prior known pattern called *magic number*. This specific part could be several bytes on the beginning of the file, which is known as file header. Also, most of file types have end marker of the file as trailer or file footer. This method can be used to identify a full complete file that has proper header and footer. It will fail when testing a fragmented file that is not a complete file. In certain circumstances, a file can be deliberately split into several parts using free tools that are available on Internet. Moreover, for a knowledgeable technical employee, altering file header and footer of specific file might not a big problem. By altering magic number in file header and footer, this technique also fails to identify the true file type. In an advanced technique, another file could be appended at the end of image file and append it with the original footer. Operating System will only see the whole image file, and application handler also will only process the original file. Meanwhile, the appended file is undetected. This evasion technique is known as one of steganography technique.

With another approach, researchers then use the statistics to see what characteristics inside the file that can be used to determine and to differentiate a file uniquely. These characteristics will be used as *features* on how to determine and classify the file type using statistical machine learning methodology. The common feature in statistical classification approach called n-gram—that is an array of bytes with the size of n. In other words, 8-gram data means 8 bytes of adjacent data block inside a file. With this technique, any n-grams can be extracted anywhere inside a

file, not necessarily in a file header or in file footer. Relying on this feature can be useful in determining a specific file type.

1.3 Scope of work

This thesis discusses the use of various FTI techniques with emphasize of n-gram analysis to minimize the insider threats in Enterprise that try to steal the confidential data by altering the specific part of the data itself. This deliberate alteration is an effort to conceal the true file type, so it will not be detected easily by certain technology such DLP. The alteration attempt could be: (i) file extension alteration, (ii) file header and footer alteration. To limits the works, this thesis will not discuss on how to detect file that is encrypted (e.g. PDF file with password) or compressed with any algorithm (e.g. ZIP, TAR, etc.) or hidden file using any techniques in steganography. This thesis also limits the discussion only on how to identify a full complete file, not for identifying an incomplete nor a fragment of file.

As a problem set, this thesis only recognized the common file format that can be considered as confidential documents in most enterprise. These file types include:

Format	File Type
Pictures File	JPG, PNG, BMP, GIF
Documents File	PDF Office 2003 (DOC, XLS, PPT) Office 2007 (DOCX, XLSX, PPTX)

To simulate the alteration, this thesis will test the files in several ways:

(i) testing the proposed solution towards the same data set with previous work to compare its effectiveness;

(ii) testing with file extension alteration (e.g. DOCX to JPG, PDF to GIF);

(iii) testing with file header alteration (e.g. from JPG image file header

changed to PDF file header); and

(iv) altered both file extension and file header.

1.4 Document organization

In this document, there are several chapters to discuss on how the idea of using n-gram to identify the file type.

Chapter 2 discusses about the previous works on how to perform File Type Identification (FTI) task, along with their advantages and caveats. This chapter also discusses the proposed solution for a problem in File Type Identification task.

Chapter 3 presents the design of proposed solution on how to utilized the automated extracted long n-gram and use the to predict a set of files.

Chapter 4 exposes the result of extraction process and prediction process.

Chapter 5 discusses the conclusion and further works for the proposed solution.

Chapter 2: Motivation of Thesis

There are several previous works that have been done and published as computer security journal. Most of these works are performed and tested with unaltered file. Most of the authors assumed that files remain intact, so there is no alteration in the binary level of the file that being tested.

2.1 Previous works

2.1.1 Magic Number

In 1973, in his website (Darwin), Ian Darwin developed a `file` command to do a best guessing about what kind of file type for a given tested file. This technique actually reads several first bytes of the file to find 'magic number' and compares it with the database of a known 'magic number' for a certain file type. This technique still developed actively by a group to refine the database of 'magic number'. When this report is written, the current version of file is 5.16 with compiled date December 1, 2013.

This technique still have flaw to recognize the file. In special circumstances, a modified file with altered file header and file footer can conceal its true file type. Adversaries can modify the file header of JPG image file format into PNG image file format. This `file` command will recognize this file as PNG file instead of JPG file.

On more advance implementation, `ExifTool` can read all metadata associated with specific file type (Harvey). Instead of simply read the first few bytes of file header, this tool can read the metadata based on known file structure recognized at the first header. If one certain file identified as JPG file, this tool can extract the information such image width, height, resolution, and if available, it can also extract the GPS information where did the picture taken. On a different file type, for example

Microsoft Word Document, this tool can extract the metadata information such as: total editing time, username that created the document, number of words and characters, etc. This information is very useful and complete enough to conduct surface analysis towards the file without opening the file. The adversary needs to delete all of information relates to this file type in order to conceal the true file type.

2.1.2 File Extension

This method was commonly used as file association techniques on popular Operating System such Microsoft Windows. In 1996, Microsoft developed this technique that let operating system build a proprietary hierarchical database called System Registry as stated in (Hiller). This registry has capability to associate a file extension as a known file for a known application to handles it.

Although this is the fast way to differentiate a file type, in other documentation (Support) said that this method had a certain limitation. It is said that Operating System couldn't associate the file without extension. Operating System simply trust the file extension. If one executable file with EXE extension is renamed with PDF extension, Operating System will open it with associated application that handles PDF format. The accuracy of this technique is very low, since it is simply rely on the file name extension.

2.1.3 Byte Frequency Analysis (BFA) Algorithm

This algorithm was introduced in (McDaniel) that enables a 'fingerprint' of a file can be created based on the occurrence of bytes that is ranging from 0x00 (0 in decimal) to 0xFF (255 in decimal). This occurrence then normalized to provide the equal weight, no matter how big the file size is. With a predefined fingerprint of several known unaltered files, a comparison can be made with an unknown file to determine its file type.

The author explains that this algorithm had a very low accuracy. Among 120 files that have been tested, only 33 files can be guessed correctly. That means this algorithm yields the accuracy of only 27.50%. The author did not specify any details why this method failed.

2.1.4 Byte Frequency Cross-Correlation (BFC) Algorithm

This algorithm was also introduced in (McDaniel). The difference between this method and BFA methods is the author analyzed the average frequency difference between known file and unknown file. Then calculate the correlation strength between those.

This method had a slightly better level of accuracy to 45.83%. But the author also suggested that this method was not accurate enough for real world implementation.

2.1.5 File Header/Trailer (FHT) Algorithm

The same author wrote on (McDaniel) said that instead of using BFA and BFC which make use of all byte value in one file to identify the real file type, there is a specific way to increase the accuracy. The author used the file header and trailers to generate fingerprint of header and footer. Different with `file` from (Darwin) that only looking for the occurrence of known 'magic number' in header and footer, this FHT algorithm performed the same approach with BFA and BFC, but only for a H bytes of header and T bytes of Trailer.

The result of this algorithm was significantly increased to 95.83% (115 files recognized out of 120 files). There is no data of accuracy when the file header and footer were maliciously altered into different file type.

2.1.6 Fileprints with 1-gram Analysis

This method was introduced as file type identification technique at (Li, Wang and Stolfo). The authors suggest to use different approach with (McDaniel) in order to build a model invariant of frequency normalization toward file size. The technical approach itself is called n-gram analysis that was introduced at (Damashek). Further, these authors state on their journal that n-gram analysis was applied to binary content of file data set and produced a normalized n-gram distribution that represent all tested file type.

The terminology n-gram refers to n-byte token that is used as identifier. In (Li, Wang and Stolfo), the authors use 1-gram analysis as the improvement to increase the accuracy from (McDaniel). With this approach, a string of data is converted to 1-grams as a data reference model. From this model, the authors considered to perform comparison of an unknown file with this model using Mahalanobis Distance.

The accuracy of this method is almost 100%. But surprisingly, the bigger fragment size of a sample file, the accuracy is decreased. And the lowest level of accuracy was hit when all bytes in a full complete file was sampled.

2.1.7 Sliding Windows

Instead of relying on short gram (i.e. 1-gram or 2-gram) using fixed size matrix, (Hall and Davis) introduce the technique to file type identification using sliding window to measure the entropy of internal structure of the file. The term of entropy itself was used by (M. M. Shannon) which was originally introduced by (C. E. Shannon) as the randomness of internal data in file. In 2004, (M. M. Shannon) used this entropy number to identify a compressed file or encrypted file.

With sliding windows, (Hall and Davis) used a fix size that is smaller than the full file size itself. Within this window, the process of calculation took place. Then the window slides to exactly next one byte while keep maintaining the window size. With this movement, the authors eliminate the influence of the first byte by adding the next byte into the end of window position. The authors claim that the result of this technique were promising for several file types even with low rate of true positive.

2.1.8 Long summarized n-grams

On (Mayer), the author uses different approach and suggests that using short n-grams of one or two bytes will not be an excellent indicator to identify the real file type. In addition, author convinces the reader that short n-grams have a high possibility to appear in any random binary data. Meanwhile, the probability of longer n-grams appears in random data decreased dramatically.

On his Master Thesis, the author creates an algorithm to find long n-grams from certain file set, up to 20-grams. These long n-grams were tested on the specific file set to generate common long n-grams. Each of common long n-gram found will be summarized. This means that n-gram will be converted such that all upper case ASCII characters will be converted into lower case ASCII characters. For any occurrences of ASCII integer value, will be converted to ASCII integer '5'. With this substitution, the author eliminates the stiffness of string found and claimed that more common long n-grams can be found. From summarized n-gram, the author with the past experiences and knowledge manually defines the predictability of each n-gram whether it can be used to as a feature in file classification.

2.2 Proposed Solution

Among all of those techniques, most of them are trained or used to identify unaltered complete file. Moreover, one technique only yields one result. Without having any comparison whether the result of individual technique is reliable or not. If these techniques were implemented in enterprise, it is not practical for the security operator to determine whether one file is truly classifies as a certain file type or the result was false positive.

In (Mayer), the author introduces how to perform n-gram analysis and extract the long n-gram then summarized it. Unfortunately, when the author used the standard corpus data from (Garfinkel, Farrell and Roussev), he does not state whether he performed a md5 hash check first to guarantee that only unique file that can be consider his data set.

In this thesis, the proposed solution is to combine several techniques and compare each result to inform the user of this application about the inconsistency between findings. The methods are: (i) file extension detection, (ii) magic number, (iii) Metadata extraction, and (iv) long n-gram analysis with sliding window. Those techniques will be combined in a single process to predict the file type of a given file as depicted in Figure 2 below. Meanwhile, the learning process is straightforward from extracting the n-grams from given data set and save those grams as common grams without any manual intervention from operator.

As data set for learning process, this thesis only uses the file that has a unique md5 hash value. When two files or more have the same value, the author of this thesis assumes that these files are identic, even though (Daum and Lucks) demonstrate their ability to create two different PostScript files with the same md5 hash value by using technique (Wang and Yu) journal. The author of this thesis also performed

manual checking of the files that has identic md5 value to see whether those files are really identic or not. All of files that have same md5 hash value in corpus standard data are identic.

This selection guarantees that only a unique file is learned. This uniqueness could prevent the learning algorithm to wrongfully consider one n-gram as a common gram because of its occurrence in multiple files but with the same md5 hash value.

From the Figure 2 below, the process requires a file as input. The file could be coming from any processes. If enterprise wants to detect all document files that will leave the network, these files can be extracted either from Mail Server or using Internet Content Adaptation Protocol (ICAP) that acts as transparent proxy. The extraction process yields a full, non-fragmented, and complete file. Each of these files can be processed to this workflow.

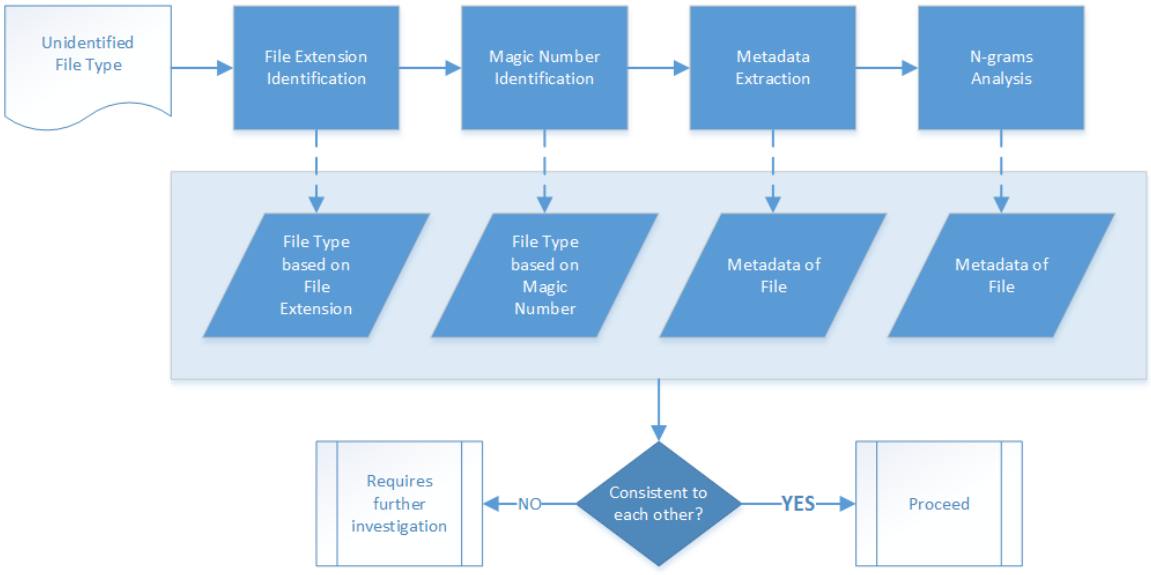


Figure 2 Workflow to predict file type as proposed solution

Each rectangle represents a prediction module that performs a specific prediction task as implementation of each technique stated as proposed solution. The output of each module is represented as parallelogram. The file will be processed by all individual module and the results will be concluded either the file is consistent with its file type or the file requires further investigation since the result is inconsistent between each module.

Chapter 3: Design of Proposed Solution

This chapter presents the idea on how proposed solution is designed an implemented into a practical solution for an enterprise.

3.1 Overview

Implementing a process to perform an automated extraction feature using long n-gram analysis requires a careful design and strategy. A naïve implementation using a fix multiple dimension of matrix can consume great resource. This fix size of matrix data representation was implemented in prior works such Byte Frequency Analysis (BFA), Byte Frequency Cross-Correlation (BFC), etc. For 1-gram data, the process requires a matrix that can hold binary representation from 00 until FF, which means 256 elements. When implementing 2-gram data with this data representation, a matrix with element of 256 x 256 required to store each 2-gram. It means that for n-gram data, the storage required for this representation is 256^n, which is not practically effective.

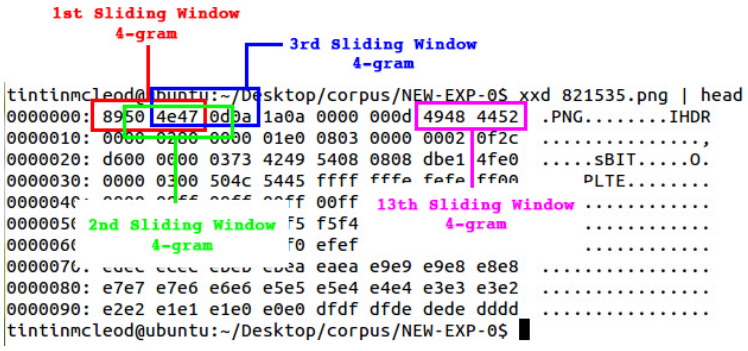


Figure 3 Sliding Window in binary representation of PNG file

Instead of using predefined matrix, a sliding window will be used to index the n-gram that only occurred in the file as depicted at Figure 3 above. Each data in this window will be processed both in learning mode and prediction mode. Hence, only n-gram data that exists will be considered in analysis process. This design can reduce the usage in compiler memory.

The implementation for this application will utilize Java programming language with the use of PostgreSQL database to store the data. There is no particular reason to choose those technologies. The author of this thesis solely determined the use of Java 7 and PostgreSQL 9.1.10 by the familiarities and preferences to use them. The Integrated Development Environment used is NetBeans IDE 7.4 that runs on Linux Ubuntu 13.10 (saucy) 64-bit with kernel version 3.11.0-13-generic. The Java VM in NetBeans is set to use all resources in order to speed up the process with VM options –Xms8192m –Xmx8192m. This operating system runs inside virtual machine with 8GB memory, 2 Processors Intel Core i7-3720QM 2.60GHz with 120GB hard drive. The virtual machine was created with VMware Fusion 6.0.2 (1398658) for OS X Mavericks 10.9 on MacBook Pro Retina Display 15-inch Mid 2012).

3.2 Learning Process

Learning process is the core module to extract the common features from file types. The process should be designed specifically to find specific characteristics of n-gram that occurred on all learning data set for the same file types. The first step is handled by sub module to extract all n-grams that occurred on the given files. After all learning file set was extracted, the next sub module will normalize the extracted n-grams. For each extracted n-gram, the process automatically selects which n-gram exists in all same file type but does not exist on the other file type. The following is the explanation about sub modules.

3.2.1 Extraction

In this sub module, each file will be read and copied into data representation as array of bytes. This array of bytes will be calculated its md5 hash value to determine whether this current file has been learned before or not. This process guarantees

that all files in learning set are unique to each other. If an identic file learned more than once, the Learning process will consider all n-gram in these files become a common feature since those n-grams occurred on more than one file.

After the file was checked against its similarity with other file in data set, the md5 hash value stored to database for comparison with next file as information that this file has been learned. The next following steps are for the core process of extraction of each single file:

1. Determine the maximum size of n-gram that will be extracted from file. In this thesis, the author chooses the maximum size is 20-gram.
2. For each n-gram size, repeat these steps:
    a. Create sliding window with the size of n, started with value of n = 1.
    b. Copy the gram inside this current window to memory object as *extracted n-grams*. If this gram has not been stored in memory object, then simply put it on memory object. Otherwise, proceed to next step 2c. This step guarantees that memory object only contains *unique extracted n-grams* that exists on this current file and not repeated between other n-gram.
    c. Slide the window 1 byte away, then repeat the step 2b above until sliding window reach the end of the file.
    d. Write all *unique extracted n-grams* in memory object to database, along with its file identification such file name, file size, and file type. Until this step, the database contains all *unique extracted n-grams* data from size n = 1 up until n <= 20.
    e. Increase the size of n with 1, and then repeat from step 2a until the maximum size of n is reached.

Until this point, the database now contains all unique 1-gram to 20-gram within this file. To extract n-gram from other files, repeat from the step number 1 again.

3.2.2 Normalization

Sub-module Extraction yields a database with all *unique extracted 1-gram up to 20-gram data*. These data are not unique enough to be used in Prediction process. Thus, this sub-module requires normalization process for the *unique extracted n-gram* so that each *unique extracted n-gram* only occurred once regardless its file extension that is associated with it. These are the steps to normalize the *unique extracted n-gram* so that can be used as feature in Prediction process.

1. Lookup the database that contains all *unique extracted n-gram*. Count how many unique file types that have been populated by previous sub-process.
2. For each unique file types, find all of *unique extracted n-gram* that only exists on all files that share the same file type, from 1-gram until 20-gram. These n-grams then named as *common n-gram* for file type.
3. Normalize the *common n-gram* for each file type using these steps:
    a. For each size of *common n-gram*:
        i. Choose only *common n-gram* for file type (from process step 2) that only represent one file type. That is, if same common n-gram associated with more than 1 file type, then this common n-gram cannot be considered as a *temporary unique n-gram*.
        ii. Compare all of *temporary unique n-grams* from step 3.a.i above with *final unique n-grams* in database.
            1. If *temporary unique n-grams* already exist in *final unique n-grams* with the same file type, it means that these *final unique n-grams* still considered as feature.
            2. But if *temporary unique n-grams* already exist in *final unique n-grams* with the different file type, both

temporary unique n-grams and *final unique n-grams* should be removed, since two file types share the same unique n-gram.

3. If *temporary unique n-grams* not exist in database, it means that these *temporary unique n-grams* considered as *final unique n-grams* also known as *final features* that can be used to predict file type in next process.

b. Repeat the step 3.a above until n = 20.

4. Repeat step 2 above until all file types were processed.

3.3 Prediction Process

This module is more straightforward rather than Learning process. There are four sub modules as proposed solution. Each unknown file will be processed through these sub modules, and all results from each process will be aggregated by last sub module to generate a conclusion about file type.

3.3.1 File Extension Predictor

This sub module simply recognizes the file type by using its extension. Each file that becomes an input for Prediction process will be treated as file descriptor and string value of its file name will be processed. The last position of '.' (dot) character is the first position to substring the filename until the last character which become the file extension. This file extension is stored for the comparison purpose with other sub modules.

3.3.2 Libmagic Predictor

One of magic number implementation is `file` Linux command. This application searches a predefined string of bytes at certain location in the header or footer file section. For example, if the beginning of the file started with 0xDE 0xAD 0xBE 0xEF

(DEADBEEF), the file is recognized as SiLK network flow data. While 0xCA 0xFE 0xBA 0xBE (CAFEBABE) is the indicator of Java Bytecode. These magic numbers are maintained actively as its database to predict the file type.

Hence, to prevent reinvent the wheel, this module will invoke the `file` command that has been installed previously on Virtual Machine. The output from this execution will be parsed to determined what the file type is.

3.3.3 Metadata Predictor

Instead of reading only predefined magic code, `exiftool` reads more information rather than just identifying file type. With this information, the result becomes more comprehensive. For image file, this tool can read and shows the image dimension width and height, along with the bit depth, software name that creates the file, MIME type, etc.

Similar with Libmagic Predictor, this module also invoke the `exiftool` Linux command that has been installed on Virtual Machine and parse the output to determined what is file type for the input file.

3.3.4 N-gram Predictor

This sub module utilizes the result from 3.2 Learning Process. Similar with Extraction sub-module, each file will be read and copied into data representation as *`array of bytes`*. After converted as object in computer memory, these steps are executed as follows:

1. Determine the maximum size of n-gram will be compared with file. In this thesis, the author chooses the maximum size is 20-gram similar with Extraction process.
2. For each n-gram size, repeat these steps:

a. Get all *final features* from the database based on n-gram size, regardless what file type associated with it, along with its *total final features counter* for each file type.
b. For each file type in *final features* with current size of n:
    i. Create a *final features statistics container* to hold statistics of all matched *final features* per file type. This container will be used to count how many *final features* occurred in *array of bytes*. This *final features statistics container* contains file extension, size of n, *total final features counter*, and matched *final features counter*.
    ii. Repeat process 2.b.i until all of file types have each different *final features statistics container*.
c. Create sliding window with size is current n.
d. Check whether this current gram in *final features*. If current gram exists in *final features* then updates its statistics in *final features statistics container*. Otherwise, just skip it.
e. Slide the window 1 byte away from the original position.
f. Repeat from 2.d until the window reaches the end of file.

3. Calculate the percentage match based on value in *final features statistics container*.
4. Determine the acceptance threshold toward percentage match so that this matching feature can be used to determine whether this is accepted as specific file type. For this thesis, author set the threshold of acceptance is 95%. Hence, if a set of features in size of n for file type X matches more than 95% of total features, than this file is classified as file type X.

3.3.5 Aggregate Predictor

This predictor is only to summarize either all of the predictor sub modules above give the similar result or not. If all of predictors give the same result of file type, it means that the file is consistent to be classified a file type that can be validated via its file extension, its magic code, its metadata and also its body file using n-gram analysis.

Chapter 4: Result and Analysis

Before executing the algorithm, learning set and testing set should be defined first.

4.1 Data Set for Learning and Prediction

In order to compare the result between the algorithms written for this thesis towards n-gram analysis with analysis from (Mayer), this thesis will use the same corpus data. In (Mayer) also using the standardized data set proposed by (Garfinkel, Farrell and Roussev). The aim for this standardization is to make the result from digital forensics research can be reproduced by other researchers. This data divided into multiple content sensitivities that collected from disk images, memory images, network packets, and files that are available on this site[1]. High sensitive files requires a clearance such IRB Approval and any other paper works.

For the Learning purposes, this thesis uses files that are considered has low sensitivity content inside 10 compressed files named thread0.zip[2] sequentially until thread9.zip. These files are public available for free. The files were selected carefully so that only files that have "pure" structure and content can be used as data set. The purpose of this action is to maintain the "pristine" condition of extracted n-gram.

This thesis expects that the algorithm in this thesis can detect the file container structure along with information about what kind of file type inside the container. If a learning data file contains another file type, say JPG, this image file might be encapsulated and encoded using certain method such as base64. Or, this image file is simply attached on a binary format without any encapsulation or encoding at all. If this image file is simply attached on the binary file format, the learning process will

[1] Digital Corpora. http://digitalcorpora.org/corpora/files. Last Accessed: December 1st, 2013.

[2] Govdocs 1 Million Files Corpus. http://digitalcorpora.org/corp/nps/files/govdocs1/thread0.zip

consider all of n-grams in image files belong to file type of its container, which make the prediction process could go inaccurate. The advantage of providing a "pristine" document file separated with image file as learning data set will be shown up on the Prediction module. List of files that being used as Learning Data Set, PRISTINE-DS-0, can be seen on Appendix A.

Meanwhile, for testing data set to verify the algorithm, this thesis also using the same data set that being used by (Mayer) which is identified as NEW-EXP-0 that is introduced by (Axelsson). Both researchers used this data set to experiment towards file fragments. The data set contains 267 files of 28 different file types. Since this thesis only focus on 11 file types, the data set is reduced to only represent 11 file types. Total files for the Prediction data set is 90 files. The name of files that are used in this thesis as Prediction Data Set can be observed at Appendix D until Appendix G.

There are several different number of files that were used by Mayer as stated on his document, page 49 Table 4.1. All of the files that are used as experimental data set (NEW-EXP-0) was stated on Mayer's document page 73 Appendix A. These differences can be observed on Table 1 below.

In this thesis, the author searched on Mayer's document on Appendix A and can only found number of files that is less than total files that were used by Mayer in Table 4.1 on page 49. This number can affect the calculation rate such accuracy, recall, and precision. If there is a different amount of file that being use to test the prediction process, the result cannot be compared directly. In this document, all of available files that stated in Appendix A page 73 of Mayer's document were used as is. For example, BMP files that were used to test Mayer's algorithm in his Confusion Matrix at Table 4.1 page 49 were 10 files. Meanwhile, there are only 8 files of BMP in

Appendix A page 73. The author of this thesis uses 8 BMP files instead of 10 BMP files.

Table 1 Different Number of Sample for each file type

File type	Number of Sample Files	
	This Thesis	Mayer
BMP	8	10
PNG	10	10
JPG	10	10
GIF	10	10
PDF	10	10
DOC	5	7
PPT	9	9
XLS	5	9
DOCX	8	9
PPTX	9	10
XLSX	6	10
Total Files	90	104

4.2 Learning Process

Executing the algorithm for Learning process is very time consuming. The algorithm is still in standard performance code without having any optimization for achieving high performance. No multithreading task and hardware optimization were implemented. On the earlier version of application, to learn 37 files directly in one batch, this algorithm finished the process to automatically extract all common features and processed it as a unique feature in 1,373 minutes 17 seconds or more than 22 hours. After tweaking in the algorithm and make the application more effective, time to finished the same task was decreased to 1,073 minutes 33 seconds.

The time to complete the learning process is still not acceptable. After several cycles of optimizing data representation in programming, executing algorithm, and

examining the extracted features; the author find out that the file size of learning process data set determined the time of execution. Thus, the author manually handpicked which files are considered as learning data set while still maintaining the same corpus data set.

The goal of this algorithm is extracting the file features automatically by comparing the occurrence of the features on all similar file types. If one file type with small file size can be considered as a valid file type, it means that inside this file the minimum structure of file features were satisfied. By neglecting the file content, a small size of file type is enough to determine which file features can be extracted from a specific file type.

On (Mayer), the author did not specify the file size of each file that has been chosen as data set in feature generation process (similar with learning process in this thesis). Instead, Mayer stated that the number of files used in his learning process did not guarantee the features of certain file type could be found. Hence, Mayer limited his data set to use 20 files for each file types.

Using similar number of file for each file type, this thesis defines the maximum file size is 1 MB while keep maintaining its "pureness". All of files that chose as data set should not contain another file type. For example, a document file DOC should only contain text without any embedded image or picture. To satisfy this limitation, it is very difficult to file PowerPoint format (both PPT and PPTX) that does not contain any picture at all. Instead of collecting 20 files, only 11 files being used for each PowerPoint (Office 2003) and 3 files Power Point (Office 2007). These files still collected from standard corpus. All of files that are used as learning data set, named PRISTINE-DS-0, can be seen at Appendix A on this document.

Instead of learning all data set directly 194 files in one batch, the learning process was split into 11 batches. Each batch was specifically learned all files on the same file type. Details about each batch can be seen in Table 2 below which is ordered by time of execution. First row means the first batch executed, the last row means the last batch executed.

Table 2 Learning process for each file type data set

File types	Num Files	Time to finish	AVG File Size	Max File Size
BMP	20	9 minutes 2 seconds	27,708	78,654
JPG	20	23 minutes 48 seconds	77,421	553,031
GIF	20	12 minutes 26 seconds	42,439	200,822
PNG	20	129 minutes 28 seconds	276,372	896,967
PDF	20	19 minutes 10 seconds	77,874	432,424
DOC	20	11 minutes 22 seconds	107,648	551,936
XLS	20	6 minutes 10 seconds	34,227	158,720
DOCX	20	7 minutes 32 seconds	25,559	95,167
XLSX	20	10 minutes 51 seconds	41,299	218,613
PPT	11	5 minutes 32 seconds	68,282	283,136
PPTX	3	5 minutes 27 seconds	65,686	82,635
Total	194	240 minutes 48 seconds		

Number of extracted features per n-size and per file type can be seen at Appendix B. By splitting 194 files into 11 batches, the total time consumed in Learning process was significantly decreased from 1,073 minutes 33 seconds to 240 minutes 48 seconds.

4.3 Prediction Process

Using extracted feature from Learning process above, 90 files that are part of NEW-EXP-0 data set were tested during prediction process. The sample output of Prediction Process can be seen Figure 4.

One file will be determined its file type using the classification result. This result of classification is based on the threshold of matching rate that can be accepted. For this thesis, the threshold was defined with 95%. Hence, if one file has 95% or more matching rates with features from file type, this file is classified as that file type. For example at Figure 4 Sample output of Prediction Process below. A PDF file from Axelsson's data set NEW-EXT-0, has 100% matched (160 features match out of 160 extracted feature) with JPG file type. Also, this PDF file matches with 98.87% PDF file (614 out of 621). This learning and prediction process enables multiple classification of one single file.

```
Processing 58 of 90 file /home/tintinmcleod/Desktop/corpus/NEW-EXP-0/184216.pdf
--- file /home/tintinmcleod/Desktop/corpus/NEW-EXP-0/184216.pdf is 0.0000% match with .bmp extension. (0 out of 7)
--- file /home/tintinmcleod/Desktop/corpus/NEW-EXP-0/184216.pdf is 1.0349% match with .doc extension. (48 out of 4638)
--- file /home/tintinmcleod/Desktop/corpus/NEW-EXP-0/184216.pdf is 7.2115% match with .png extension. (15 out of 208)
--- file /home/tintinmcleod/Desktop/corpus/NEW-EXP-0/184216.pdf is 23.9548% match with .docx extension. (636 out of 2655)
--- file /home/tintinmcleod/Desktop/corpus/NEW-EXP-0/184216.pdf is 2.4069% match with .ppt extension. (194 out of 8060)
--- file /home/tintinmcleod/Desktop/corpus/NEW-EXP-0/184216.pdf is 100.0000% match with .jpg extension. (160 out of 160)
--- file /home/tintinmcleod/Desktop/corpus/NEW-EXP-0/184216.pdf is 50.0000% match with .gif extension. (5 out of 10)
--- file /home/tintinmcleod/Desktop/corpus/NEW-EXP-0/184216.pdf is 1.0995% match with .xlsx extension. (19 out of 1728)
--- file /home/tintinmcleod/Desktop/corpus/NEW-EXP-0/184216.pdf is 98.8728% match with .pdf extension. (614 out of 621)
--- file /home/tintinmcleod/Desktop/corpus/NEW-EXP-0/184216.pdf is 0.9823% match with .xls extension. (86 out of 8755)
--- file /home/tintinmcleod/Desktop/corpus/NEW-EXP-0/184216.pdf is 20.7856% match with .pptx extension. (7424 out of 35717)
Processing 59 of 90 file /home/tintinmcleod/Desktop/corpus/NEW-EXP-0/714314.pdf
--- file /home/tintinmcleod/Desktop/corpus/NEW-EXP-0/714314.pdf is 0.0000% match with .bmp extension. (0 out of 7)
--- file /home/tintinmcleod/Desktop/corpus/NEW-EXP-0/714314.pdf is 0.7978% match with .doc extension. (37 out of 4638)
--- file /home/tintinmcleod/Desktop/corpus/NEW-EXP-0/714314.pdf is 4.8077% match with .png extension. (10 out of 208)
--- file /home/tintinmcleod/Desktop/corpus/NEW-EXP-0/714314.pdf is 10.5461% match with .docx extension. (280 out of 2655)
--- file /home/tintinmcleod/Desktop/corpus/NEW-EXP-0/714314.pdf is 1.3400% match with .ppt extension. (108 out of 8060)
--- file /home/tintinmcleod/Desktop/corpus/NEW-EXP-0/714314.pdf is 1.2500% match with .jpg extension. (2 out of 160)
--- file /home/tintinmcleod/Desktop/corpus/NEW-EXP-0/714314.pdf is 20.0000% match with .gif extension. (2 out of 10)
--- file /home/tintinmcleod/Desktop/corpus/NEW-EXP-0/714314.pdf is 0.8102% match with .xlsx extension. (14 out of 1728)
--- file /home/tintinmcleod/Desktop/corpus/NEW-EXP-0/714314.pdf is 100.0000% match with .pdf extension. (621 out of 621)
--- file /home/tintinmcleod/Desktop/corpus/NEW-EXP-0/714314.pdf is 0.3883% match with .xls extension. (34 out of 8755)
--- file /home/tintinmcleod/Desktop/corpus/NEW-EXP-0/714314.pdf is 8.8977% match with .pptx extension. (3178 out of 35717)
```

Figure 4 Sample output of Prediction Process

At the glance, this result seems invalid. But, after observing the file as mention above (that is, 184216.pdf), this PDF file only has 1 page but there are at least 3 image elements (marked with red line in Figure 6 below). The feature that was extracted from JPG file was found in PDF file. Using `foremost` a linux application from (US AFOSI and Naval Postgraduate School CISR) to carve out files within a file, the author found out that there are 3 complete JPG files inside this document. These

files were exactly the same as 3 images elements can be seen in folder extracted → output → JPG in Figure 5 below.

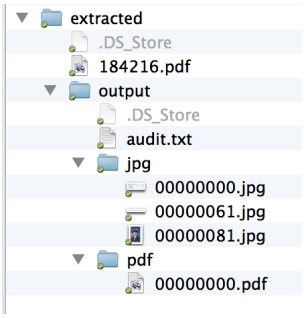


Figure 5 Carved out 3 JPG files inside PDF file

Figure 6 File PDF that fall into JPG and PDF classification

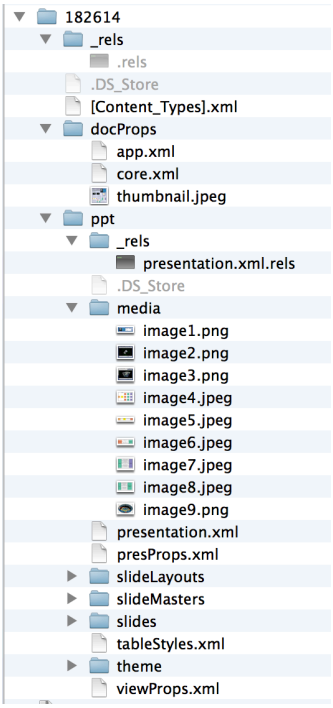


Figure 7 DOCX as A "container" file

If this PDF file was used as learning data set, this algorithm will assume that extracted features are strongly associated with PDF instead of JPG. Since both JPG and PDF have this feature, this algorithm will eliminate the features for both file type. This finding is not discussed on (Mayer), since the author uses different approach on how selecting the learning data set.

The other file format that can be consider as a container format (such as DOC, DOCX, PPTX, PPT, etc.) also shown up in the matching results. To see what inside the container format, one of DOCX file (182614.docx) was renamed to 182614.zip. By default, in major Operating System, this compressed file was deflated to see the file structure (Figure 7 above). Inside this file contained several images file such PNG and JPG. This explains why there are classification for DOCX file as PNG and JPG. For overall file classification toward Prediction Data Set can be seen on Table 3 Confusion Matrix of Prediction Data Set below.

Table 3 Confusion Matrix of Prediction Data Set

95%	Total Files Match with Extracted Features											Num of
	BMP	PNG	JPG	GIF	PDF	DOC	PPT	XLS	DOCX	PPTX	XLSX	Samples
Sample FileType												
BMP	7	0	0	0	0	0	0	0	0	0	0	8
PNG	0	10	0	0	0	0	0	0	0	0	0	10
JPG	0	0	9	0	0	0	0	0	0	0	0	10
GIF	0	0	0	4	0	0	0	0	0	0	0	10
PDF	0	0	1	0	10	0	0	0	0	0	0	10
DOC	2	0	1	0	0	4	0	0	0	0	0	5
PPT	5	6	5	0	0	0	7	0	0	0	0	9
XLS	1	0	0	0	0	0	0	5	0	0	0	5
DOCX	0	4	4	1	0	0	0	0	8	0	0	8
PPTX	3	9	8	3	0	0	0	0	0	3	0	9
XLSX	1	0	1	0	0	0	0	1	0	0	5	6
												90
Classification	19	29	29	8	10	4	7	6	8	3	5	

Row BMP in Table 3 means the data set of BMP file type, while Column BMP means that classification method using extracted feature for BMP file type. The column BMP also means that how many files that intersected with row is classified as BMP.

For example, row PDF intersect with JPG column and there's a number '1'. It means that, among 10 files of total samples of PDF file that being tested (can be seen on row PDF, column Num of Samples), there is 1 file PDF with at least 95% confidence that is matched with JPG extracted features. While the very bottom of the row shows Classification, which means how many files can be detected and classified as each column features. Using this algorithm, there are 19 files can be classified as BMP file type, 29 files as PNG file type, so on and so forth. In order to compare the result of this algorithm with Mayer's, the calculation of the accuracy and precision for this algorithm can be seen at Table 4 below.

Table 4 Accuracy and Precision for Algorithm

File Extension	Num of Samples	Total Population	Num of Not Samples	True Positive	False Negativ	False Positive	True Negativ	Accuracy	Precision	Recall	True Neg
BMP	8	90	82	7	1	12	70	85.56%	36.84%	87.50%	85.37%
PNG	10	90	80	10	0	19	61	78.89%	34.48%	100.00%	76.25%
JPG	10	90	80	9	1	19	61	77.78%	32.14%	90.00%	76.25%
GIF	10	90	80	4	6	4	76	88.89%	50.00%	40.00%	95.00%
PDF	10	90	80	10	0	0	80	100.00%	100.00%	100.00%	100.00%
DOC	5	90	85	4	1	0	85	98.89%	100.00%	80.00%	100.00%
PPT	9	90	81	7	2	0	81	97.78%	100.00%	77.78%	100.00%
XLS	5	90	85	5	0	1	84	98.89%	83.33%	100.00%	98.82%
DOCX	8	90	82	8	0	0	82	100.00%	100.00%	100.00%	100.00%
PPTX	9	90	81	3	6	0	81	93.33%	100.00%	33.33%	100.00%
XLSX	6	90	84	5	1	0	84	98.89%	100.00%	83.33%	100.00%

For example, from BMP row, there are 8 BMP files that being predicted in NEW-EXP-0. Since there are 90 files from 11 file types, there are 82 files (90 subtracted by 8) that are not BMP files. The True Positive column is taken from the green cell at Table 3 above, which shows a correctly detected BMP files using extracted feature for BMP. False Negative column tells that how many file BMP that is failed to be classified using extracted feature for BMP. False Negative is calculated from number of BMP samples (i.e. 8 files) subtracted with true positive of BMP files (i.e. 7 files). In this case, False Negative for BMP file using BMP extracted feature is 1 file. Meanwhile, False Positive column tells that how many non-BMP testing dataset is recognized as BMP file type using BMP extracted features. To calculate this, subtract

the Classification value from Table 3 above for BMP classification (i.e. 19 files were classified as BMP files using BMP classification) with True Positive value of BMP (i.e. 7 files). Hence, the False Positive of BMP classification is 12 files. While True Negative means that how many files that is not BMP files and detected as not BMP.

Another dimension that being used in Mayer's documentation is is Accuracy, Precision, and Recall. In classification context, those three rates were defined in (Olson and Delen):

$$Accuracy = \frac{true\ positive + true\ negative}{true\ positive + false\ positive + false\ negative + true\ negative}$$

$$Precision = \frac{true\ positive}{true\ positive + false\ positive}$$

$$Recall\ (or\ True\ Positive\ Rate\ or\ Sensitivity) = \frac{true\ positive}{true\ positive + false\ negative}$$

$$True\ Negative\ Rate\ or\ Specificity = \frac{true\ negative}{true\ negative + false\ positive}$$

On average, this algorithm has 92.63% of Accuracy, 76.07% Precision, and 81.09% Recall. Compared to Mayer's algorithm that yields 96.33%, 51%, and 48% respectively, which is the result of algorithm in proposed solution has a slightly higher rate compares to Mayer's. As a side notes, Mayer did not use the same formula as pictured above to calculate his accuracy. His accuracy calculation simply calculated by total of True Positive divided by all samples files, which is only 49%. With this calculation, this thesis yields its 'accuracy' at 80%.

For final result of prediction process, all of result from file extension, libmagic, metadata and n-gram are combined as one aggregated conclusion. For example, one of PPT file was classified as more than one file type (Table 5 Matching rate for 948335.ppt), that is BMP file, GIF file, JPG file, and also PNG file. Using Aggregate

Prediction, the output of final prediction process is consistent file type of PPT (Figure 8 Final Result of Prediction Process). The file extension prediction module said that this file is PPT file, also the result of libmagic and metadata viewer, consistently classify this is PPT file. While the result of n-gram analysis says that this file is matched with multiple file type, which is defined as a container file. This PPT file acts as container file that suspected contains BMP file, JPG files, and PNG files.

Table 5 Matching rate for 948335.ppt

2	Filename	.bmp	.doc	.docx	.gif	.jpg	.pdf	.png	.ppt	.pptx	.xls	.xlsx	file type
64	948335.ppt	100.00%	37.04%	32.92%	80.00%	100.00%	30.92%	100.00%	98.62%	27.88%	5.53%	3.82%	.ppt

```
Output - FileAwareness (run)
run:
Predicting file /home/tintinmcleod/Desktop/corpus/NEW-EXP-0/948335.ppt using its extension.
*** Result: .ppt
Predicting file /home/tintinmcleod/Desktop/corpus/NEW-EXP-0/948335.ppt using libmagic 'file' command...
*** Result: .ppt
Predicting file /home/tintinmcleod/Desktop/corpus/NEW-EXP-0/948335.ppt using metadata viewer 'exiftool' command...
*** Result: .ppt
Predicting file /home/tintinmcleod/Desktop/corpus/NEW-EXP-0/948335.ppt using n-gram Analysis...
--- file /home/tintinmcleod/Desktop/corpus/NEW-EXP-0/948335.ppt is 100.0000% match with .bmp extension. (7 out of 7)
--- file /home/tintinmcleod/Desktop/corpus/NEW-EXP-0/948335.ppt is 37.0418% match with .doc extension. (1718 out of 4638)
--- file /home/tintinmcleod/Desktop/corpus/NEW-EXP-0/948335.ppt is 100.0000% match with .png extension. (208 out of 208)
--- file /home/tintinmcleod/Desktop/corpus/NEW-EXP-0/948335.ppt is 32.9190% match with .docx extension. (874 out of 2655)
--- file /home/tintinmcleod/Desktop/corpus/NEW-EXP-0/948335.ppt is 98.6228% match with .ppt extension. (7949 out of 8060)
--- file /home/tintinmcleod/Desktop/corpus/NEW-EXP-0/948335.ppt is 100.0000% match with .jpg extension. (160 out of 160)
--- file /home/tintinmcleod/Desktop/corpus/NEW-EXP-0/948335.ppt is 80.0000% match with .gif extension. (8 out of 10)
--- file /home/tintinmcleod/Desktop/corpus/NEW-EXP-0/948335.ppt is 3.8194% match with .xlsx extension. (66 out of 1728)
--- file /home/tintinmcleod/Desktop/corpus/NEW-EXP-0/948335.ppt is 30.9179% match with .pdf extension. (192 out of 621)
--- file /home/tintinmcleod/Desktop/corpus/NEW-EXP-0/948335.ppt is 5.5283% match with .xls extension. (484 out of 8755)
--- file /home/tintinmcleod/Desktop/corpus/NEW-EXP-0/948335.ppt is 27.8803% match with .pptx extension. (9958 out of 35717)
*** Result:  .bmp  .png  .ppt  .jpg
*** Final Result: Consistent filetype of .ppt
Done.
BUILD SUCCESSFUL (total time: 7 minutes 38 seconds)
```

Figure 8 Final Result of Prediction Process

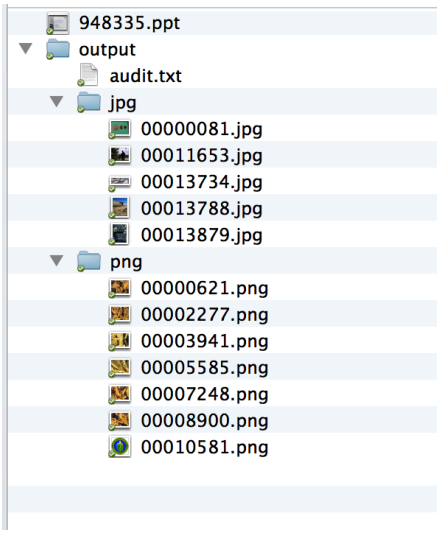


Figure 9 Extracted files from 948335.ppt

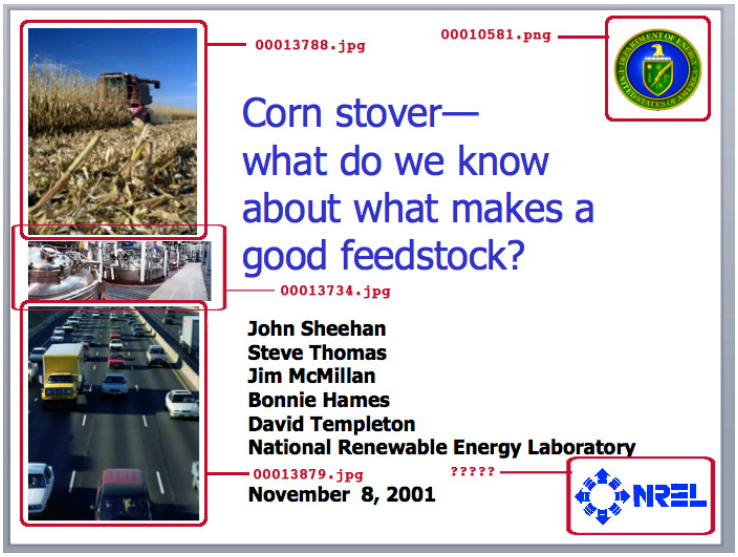


Figure 10 First slide of 948335.ppt that contains multiple pictures

```
tintinmcleod@ubuntu:~/Desktop/corpus/doctored$ xxd FileHeader_PNGtoJPG.png | head
0000000: ffd8 ffe0 0010 4a46 4946 0001 0201 0048  ......JFIF.....H
0000010: 0048 8950 4e47 0d0a 1a0a 0000 000d 4948  .H.PNG........IH
0000020: 4452 0000 05a0 0000 0384 0802 0000 00d8  DR..............
0000030: 2f01 8500 000a c069 4343 5049 4343 2050  /......iCCPICC P
0000040: 726f 6669 6c65 0000 480d ad96 7754 5349  rofile..H...wTSI
0000050: 1bc6 e7de f446 0bbd 86de 9122 1040 4ae8  .....F.....".@J.
0000060: a14b 1544 2524 8184 1262 2080 889d c515  .K.D%$...b .....
0000070: 5c0b 22d2 d405 5d69 0aae 0510 1b62 c1b6  \."...]i.....b..
0000080: 282a 6041 1764 5150 d7c5 820d 94bd 8125  (*`A.dQP.......%
0000090: bbdf 77be fdef 9b73 66e6 779f 7967 ee3b  ..w....sf.w.yg.;
tintinmcleod@ubuntu:~/Desktop/corpus/doctored$ file FileHeader_PNGtoJPG.png
FileHeader_PNGtoJPG.png: JPEG image data, JFIF standard 1.02, thumbnail 137x80
tintinmcleod@ubuntu:~/Desktop/corpus/doctored$ exiftool FileHeader_PNGtoJPG.png
ExifTool Version Number         : 9.13
File Name                       : FileHeader_PNGtoJPG.png
Directory                       : .
File Size                       : 595 kB
File Modification Date/Time     : 2013:12:10 09:43:06-05:00
File Access Date/Time           : 2013:12:10 09:46:09-05:00
File Inode Change Date/Time     : 2013:12:10 09:46:04-05:00
File Permissions                : rw-r--r--
File Type                       : JPEG
MIME Type                       : image/jpeg
JFIF Version                    : 1.02
Resolution Unit                 : inches
X Resolution                    : 72
Y Resolution                    : 72
Image Width                     : 55322
Image Height                    : 41755
Encoding Process                : Progressive DCT, differential Huffman coding
Bits Per Sample                 : 83
Color Components                : 81
Warning                         : JPEG format error
Image Size                      : 55322x41755
tintinmcleod@ubuntu:~/Desktop/corpus/doctored$
```

Figure 11 Altered PNG file and its result with tools

The author of this thesis dissects the PPT file using `foremost` and the extracted files can be seen in Figure 9 above. When this PPT file was opened in Microsoft Power Point for Mac 2011, the first slide of this file (Figure 10 above) shows multiple pictures that are identic with the extraction process. This is why the PPT file also classified as JPG and PNG files. But on the extraction process, there are no single files of BMP. The author found out that there are 106 occurrences of file header of BMP, that is 0x42 0x4D (ASCII representation: BM). Each of this file header is not followed by 4 bytes indicates its file size (Bahn). That is why foremost cannot find the proper BMP file inside this PPT file. Hence, this BMP file classification is considered as False Positive.

To test this algorithm against alteration, the author performed a file header manipulation. An original PNG file was altered into JPG file by simply add 16 bytes of JPG header into a PNG file. The binary representation of the files can be seen in Figure 11 above.

This altered file was tested using `file` and `exiftool` application. Those applications were deceived by this alteration, since both of application told that this altered PNG as a JPG file. Using the algorithm from this thesis, the result said that this altered file was PNG. Since there are inconsistencies between file extension, libmagic analysis, and metadata, the aggregate predictor said that there was an inconsistency in this file. It signaled the operator to perform further investigation toward the file.

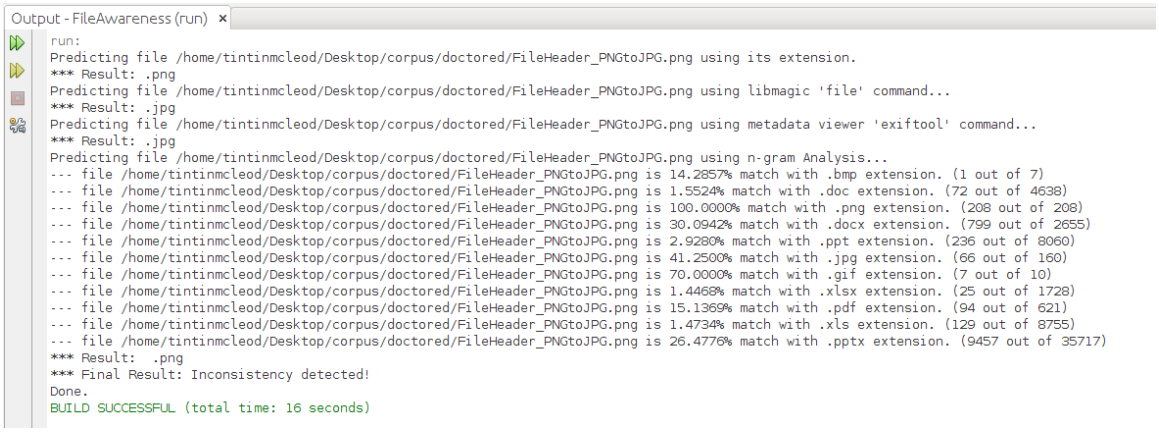


Figure 12 Aggregate Predictor result towards altered PNG file

Over time, the prediction process took less time than its first execution time. In this experiment, the first prediction test took 8 minutes 54 seconds. And then after several executions, the prediction test took only 2 seconds.

Chapter 5: Conclusion and Further Works

Performing File Type Identification to full complete file using algorithm in this thesis can be practical for enterprise to minimize the threat of data loss. Using this proposed solution, security operator in enterprise can be aware about the real file type by utilizing four different methods to classify the file type. Using this comparison, decision maker can be confident enough to take further action towards an unknown file.

5.1 Conclusion

This thesis takes different approach with (Mayer) on how to extract the feature from file and use it as an automated feature to determine the file type. All of process from learning and prediction are automated without any intervention to determine whether one feature can be used to classify one file (Mayer's use term of predictability). Mayer used a specific column that created by him manually to flag whether the gram can be used to predict file type. It requires a good knowledge and experiences to determine grams' predictability.

This thesis simply populates all of occurred n-gram in certain file, and compares it with the same file type. If it is also occurred in other file but same file type, this n-gram is common enough to be considered as common gram for file type. But in order to use this common gram as feature, this common gram should be compared with other common gram from different file type. If it is not same with different file type, then this common gram can be considered as features.

Compares with this algorithm in this thesis and (Mayer), the average accuracy is 92.63% significantly lower than 96.33%. Using 'accuracy' calculation from Mayer's, this thesis yields 80% rate compared with 49%. Also, this thesis yields average

precision at 76.07% and average recall 81.09%, which are also higher than 51% and 48% respectively.

Using different approach on how choosing files data set for Learning, this algorithm also able to detect container file along with the file content inside.

5.2 Further Works

In (Mayer) the author suggested that it was necessary to apply the extracted long n-gram with statistical approach such Support Vector Machine. It is believe can significantly improve.

To speed up the process in Learning and Prediction, multithreaded programming using multiple processors believe also can decrease the processing time. Moreover, utilizing GPU such nvidia CUDA for parallel computation also useful for improvement so that learning process should not be split into multiple batches.

There are several ways to exploit or to evade from this algorithm. One of them is compressing the restricted file using a popular file format such as ZIP or TAR. It is possible to have several ZIP files inside a ZIP file. This algorithm cannot predict this technique of encapsulation.

There are a lot of file formats that could be used to evade from this algorithm, for example Microsoft Compiled HTML Help (CHM), or MIME HTML (MHTML). These file formats embed all resources as one single file using specific encoding.

From several trials, the author of this thesis notes that the trend of increasing rate of accuracy, precision, and recall when new file types were introduced to this algorithm. The detail results of these trials can bee seen on Appendix D until

Appendix G. When the learning process only recognized 4 file types, the rate of Accuracy, Precision and Recall was increased from 81.39%, 30.68%, 63.13% to 81.67%, 39.93%, 76.25% when the number of files increased from 10 files for each file type to 20 files for each file type. The increasing trends also occurred when number of files for each file type increased from 10 files to 20 files in learning process that recognized 11 file types from 88.28%, 65.39%, 59.09% to 92.63%, 76.07%, 81.09% for each Accuracy, Precision, and Recall respectively.

Table 6 Increasing trends in increasing file type and file samples

Experiment	File Types	Num of sample Each file types	Accuracy	Precision	Recall
A	4	10	81.39%	30.68%	63.13%
B	4	20	81.67%	39.93%	76.25%
C	11	10	88.28%	65.39%	59.09%
D	11	20	92.63%	76.07%	81.09%

Bibliography

Axelsson, Stefan. "The Normalised Compression Distance as a file fragment classifier." The International Journal of Digital Forensics & Incident Response 7.Supplement (2010): S24-S31.

Bahn, William L. Simplified Windows BMP Bitmap File Format Specification. 27 November 2010. 10 December 2013 <http://www.dragonwins.com/domains/getteched/bmp/bmpfileformat.htm> .

Damashek, Marc. "Gauging Similarity with n-Grams: Language-Independent Categorization of Text." Science 267.5199 (1995): 843-848.

Darwin, F. Ian. Fine Free File Command. 9 September 2013 <http://www.darwinsys.com/file/>.

Daum, Magnus and Stefan Lucks. Attacking Hash Functions by Poisoned Messages: The Story of Alice and her Boss. 15 June 2005. 10 December 2013 <http://web.archive.org/web/20071226014140/http://www.cits.rub.de/MD5Collisions/>.

Garfinkel, Simson, et al. "Bringing Science to Digital Forensics With Standardize Forensic Corpora." The International Journal of Digital Forensics & Incident Response 6.Supplement (2009): S2-S11.

Gopal, Siddharth, et al. "Statistical Learning for File-Type Identification." International Conference on Machine Learning and Applications and Workshops (ICMLA) 10th. Honolulu: Software Engineering Institute, 2011.

Hall, Gregory A. and Wilbon P. Davis. "Sliding Window Measurement for File Type Identification." (2006).

Harvey, Phil. ExifTool by Phil Harvey. 19 October 2013. 1 November 2013 <http://www.sno.phy.queensu.ca/~phil/exiftool/>.

Hiller, Scot. The System Registry. 1996. Microsoft. 5 October 2013 <http://msdn.microsoft.com/en-us/library/ms970651.aspx>.

Lewellen, Todd, et al. Spotlight On: Insider Threat from Trusted Business Partners. Pittsburgh: Software Engineering Institute, October 2012.

Li, Wei-Jen, et al. "Fileprints: Identifying File Types by n-gram Analysis." IEEE Workshop on Information Assurance and Security (2005): 64-71.

Mayer, Ryan C. Filetype Identification Using Long, Summarized n-Grams. Monterey: Master Thesis at Naval Postgraduate School, 2011.

McDaniel, Mason B. An Algorithm for Content-Based Automated File Type Recognition. Master Thesis. James Madison University. Harrisonburg, 2001.

McKenzie, Stuart. "Project MASON: Incident Response." 29 November 2012. The Times. 10 December 2013 <http://www.thetimes.co.uk/tto/multimedia/archive/00372/DOC100113-100120132_372895a.pdf>.

Olson, David L and Dursun Delen. Advanced Data Mining Techniques. 1st. Verlag Berlin Heidelberg: Springer, 2008.

Shannon, Claude E. "A Mathematical Theory of Communication." Bell System Technical Journal 27 (1948): 623-656.

Shannon, Matthew M. "Forensic Relative Strength Scoring: ASCII and Entropy Scoring ." International Journal of Digital Evidence 2.4 (2004).

Support, Microsoft. Limitations in Windows File Association Capabilities. 1 October 1996. 5 October 2013 <ftp://ftp.microsoft.com/misc1/PEROPSYS/WINDOWS/KB/Q85/3/87.TXT>.

US AFOSI and Naval Postgraduate School CISR. Foremost. 10 December 2013 <http://foremost.sourceforge.net>.

Wang, Xiaoyun and Hongbo Yu. "How to Break MD5 and Other Hash Functions." 24th Annual International Conference on the Theory and Application of Cryptographic Techniques. Aarhus, Denmark, 2005. 19-35.

Appendix A: Learning Data Set PRISTINE-DS-0

This data set is a subset of standard corpus data that was taken from (Garfinkel, Farrell and Roussev). Instead of using all files as Learning Data Set, this data set is only contains files that is not contain another file format. For example, a document file that contains image will not be considered as a learning data set.

File Name	File Type	File Size
290963.bmp	.bmp	70,614
294567.bmp	.bmp	78,654
313709.bmp	.bmp	246
354675.bmp	.bmp	60,590
371771.bmp	.bmp	246
371772.bmp	.bmp	246
409578.bmp	.bmp	60,966
433219.bmp	.bmp	33,846
810768.bmp	.bmp	14,154
812166.bmp	.bmp	18,654
812721.bmp	.bmp	22,554
822180.bmp	.bmp	21,654
822318.bmp	.bmp	19,854
822578.bmp	.bmp	21,654
822708.bmp	.bmp	21,654
822980.bmp	.bmp	21,954
823113.bmp	.bmp	18,054
823505.bmp	.bmp	20,754
823634.bmp	.bmp	21,354
824781.bmp	.bmp	26,454

File Name	File Type	File Size
003591.png	.png	576,888
020718.png	.png	623,824
042920.png	.png	335,246
045353.png	.png	1,966
046175.png	.png	185,365
067177.png	.png	660,874
102372.png	.png	20,120
111263.png	.png	4,226
114040.png	.png	43,307
115365.png	.png	896,967
128941.png	.png	14,628
271831.png	.png	284,585
304601.png	.png	176,421
331202.png	.png	214,499
365108.png	.png	272,780
380406.png	.png	199,216
423359.png	.png	292,442
423950.png	.png	229,393
423959.png	.png	273,029
459127.png	.png	221,665

File Name	File Type	File Size
000233.gif	.gif	120,908
002830.gif	.gif	16,870
105986.gif	.gif	18,971
211719.gif	.gif	24,866
539077.gif	.gif	81,666
777731.gif	.gif	200,822
811471.gif	.gif	8,742

File Name	File Type	File Size
051349.jpg	.jpg	21,135
061561.jpg	.jpg	44,096
062096.jpg	.jpg	9,428
065426.jpg	.jpg	185,559
072125.jpg	.jpg	289,573
080543.jpg	.jpg	6,681
082113.jpg	.jpg	13,895

File Name	File Type	File Size
812877.gif	.gif	10,231
812994.gif	.gif	36,929
814901.gif	.gif	86,073
818414.gif	.gif	10,231
825157.gif	.gif	98,963
825296.gif	.gif	9,921
827208.gif	.gif	10,382
837352.gif	.gif	10,222
840017.gif	.gif	10,224
841450.gif	.gif	46,560
860681.gif	.gif	24,115
862848.gif	.gif	10,523
926383.gif	.gif	11,561

File Name	File Type	File Size
082333.jpg	.jpg	123,368
083104.jpg	.jpg	68,361
096191.jpg	.jpg	20,636
242238.jpg	.jpg	6,784
620421.jpg	.jpg	59,600
644458.jpg	.jpg	5,141
663936.jpg	.jpg	46,530
668544.jpg	.jpg	553,031
669339.jpg	.jpg	10,410
670895.jpg	.jpg	18,564
672279.jpg	.jpg	29,455
678035.jpg	.jpg	18,872
680328.jpg	.jpg	17,293

File Name	File Type	File Size
000578.pdf	.pdf	351,558
000760.pdf	.pdf	59,141
000816.pdf	.pdf	43,914
001387.pdf	.pdf	432,424
005778.pdf	.pdf	59,680
006265.pdf	.pdf	73,680
009158.pdf	.pdf	41,089
009699.pdf	.pdf	36,792
011007.pdf	.pdf	121,842
011443.pdf	.pdf	84,639
132601.pdf	.pdf	9,780
257359.pdf	.pdf	9,670
320248.pdf	.pdf	99,147
349257.pdf	.pdf	98,267
435465.pdf	.pdf	1,189
871267.pdf	.pdf	1,472
886256.pdf	.pdf	10,293
892159.pdf	.pdf	2,646
966492.pdf	.pdf	9,756
985877.pdf	.pdf	10,506

File Name	File Type	File Size
120637.ppt	.ppt	59,392
202162.ppt	.ppt	93,184
206500.ppt	.ppt	50,176
215661.ppt	.ppt	21,504
266966.ppt	.ppt	48,640
282240.ppt	.ppt	46,592
331445.ppt	.ppt	52,224
663689.ppt	.ppt	28,672
700735.ppt	.ppt	283,136
719239.ppt	.ppt	26,624
768914.ppt	.ppt	40,960

File Name	File Type	File Size
160688.pptx	.pptx	64,885
392790.pptx	.pptx	49,539
398029.pptx	.pptx	82,635

File Name	File Type	File Size
001618.doc	.doc	25,600
005649.doc	.doc	36,864
006411.doc	.doc	41,984
007135.doc	.doc	219,648
008000.doc	.doc	49,664
009367.doc	.doc	551,936
009368.doc	.doc	317,440
012147.doc	.doc	47,616
014898.doc	.doc	48,128
015139.doc	.doc	452,096
016041.doc	.doc	52,736
027716.doc	.doc	27,648
027986.doc	.doc	30,720
031129.doc	.doc	56,320
031130.doc	.doc	27,136
039585.doc	.doc	36,352
046568.doc	.doc	38,912
055742.doc	.doc	36,352
061396.doc	.doc	33,280
062414.doc	.doc	22,528

File Name	File Type	File Size
014760.docx	.docx	39,631
017091.docx	.docx	37,440
018370.docx	.docx	30,723
018371.docx	.docx	26,036
021524.docx	.docx	16,028
021526.docx	.docx	15,610
021528.docx	.docx	14,407
036269.docx	.docx	87,806
036270.docx	.docx	15,901
037025.docx	.docx	95,167
214825.docx	.docx	14,193
214826.docx	.docx	14,779
335899.docx	.docx	12,368
337861.docx	.docx	11,686
343986.docx	.docx	11,983
350647.docx	.docx	14,113
439737.docx	.docx	13,365
448890.docx	.docx	13,364
499840.docx	.docx	12,660
531177.docx	.docx	13,914

File Name	File Type	File Size
000394.xls	.xls	21,504
001084.xls	.xls	140,800
006528.xls	.xls	35,840
007249.xls	.xls	158,720
010271.xls	.xls	33,792
071973.xls	.xls	12,800
082576.xls	.xls	23,040
099521.xls	.xls	37,888
233872.xls	.xls	29,696
299200.xls	.xls	24,064
451843.xls	.xls	29,184
604440.xls	.xls	14,848
618736.xls	.xls	15,872
627113.xls	.xls	14,848
639005.xls	.xls	15,360

File Name	File Type	File Size
019916.xlsx	.xlsx	55,941
212947.xlsx	.xlsx	218,613
216693.xlsx	.xlsx	24,464
250025.xlsx	.xlsx	34,112
277247.xlsx	.xlsx	17,508
299765.xlsx	.xlsx	18,319
320165.xlsx	.xlsx	27,812
354483.xlsx	.xlsx	9,637
506449.xlsx	.xlsx	55,103
512756.xlsx	.xlsx	182,750
528206.xlsx	.xlsx	13,153
531227.xlsx	.xlsx	9,970
551532.xlsx	.xlsx	16,722
598948.xlsx	.xlsx	43,662
606946.xlsx	.xlsx	16,783

File Name	File Type	File Size
810611.xls	.xls	14,336
857899.xls	.xls	14,848
877442.xls	.xls	15,360
911444.xls	.xls	15,872
956265.xls	.xls	15,872

File Name	File Type	File Size
607004.xlsx	.xlsx	15,982
610528.xlsx	.xlsx	8,493
616494.xlsx	.xlsx	17,536
656215.xlsx	.xlsx	15,735
848990.xlsx	.xlsx	23,692

Appendix B: Number of Extracted Features per File Type and N size

File Ext	n Size	Total Features
.bmp	2	1
	4	1
	5	2
	6	2
	7	1
.doc	2	60
	3	150
	4	217
	5	258
	6	292
	7	303
	8	303
	9	295
	10	286
	11	273
	12	264
	13	258
	14	252
	15	250
	16	247
	17	242
	18	236
	19	230
	20	222

File Ext	n Size	Total Features
.docx	2	796
	3	52
	4	53
	5	67
	6	82
	7	99
	8	108
	9	115
	10	117
	11	119
	12	120
	13	121
	14	122
	15	120
	16	117
	17	114
	18	112
	19	111
	20	110
.gif	2	7
	3	2
	4	1

File Ext	n Size	Total Features
.jpg	2	9
	3	26
	4	25
	5	23
	6	20
	7	17

File Ext	n Size	Total Features
.png	2	20
	3	23
	4	23
	5	23
	6	20
	7	18

File Ext	n Size	Total Features
	8	14
	9	11
	10	8
	11	5
	12	2
.pdf	2	82
	3	132
	4	120
	5	103
	6	73
	7	51
	8	29
	9	16
	10	8
	11	4
	12	2
	13	1

File Ext	n Size	Total Features
.pptx	2	9413
	3	1006
	4	1036
	5	1088
	6	1149
	7	1210
	8	1268
	9	1319
	10	1368
	11	1429
	12	1494
	13	1553
	14	1609
	15	1663
	16	1720
	17	1774

File Ext	n Size	Total Features
	8	16
	9	14
	10	12
	11	10
	12	8
	13	6
	14	5
	15	4
	16	3
	17	2
	18	1
.ppt	2	205
	3	483
	4	548
	5	597
	6	602
	7	605
	8	586
	9	542
	10	500
	11	457
	12	416
	13	388
	14	362
	15	343
	16	323
	17	302
	18	283
	19	266
	20	252

File Ext	n Size	Total Features
	18	1822
	19	1877
	20	1919
.xls	2	100
	3	262
	4	370
	5	423
	6	460
	7	471
	8	474
	9	481
	10	494
	11	507
	12	518
	13	524
	14	529
	15	534
	16	533
	17	528
	18	523
	19	516
	20	508

File Ext	n Size	Total Features

File Ext	n Size	Total Features
.xlsx	2	23
	3	30
	4	46
	5	64
	6	81
	7	93
	8	98
	9	101
	10	104
	11	107
	12	109
	13	112
	14	115
	15	113
	16	110
	17	107
	18	106
	19	105
	20	104

Appendix C: Samples of extracted n-gram in ASCII representation

These are the ASCII representation of samples extracted n-gram from 11 file types with 20 files each. There are more than 62,000 of total n-grams extracted from 1-gram to 20-gram. These are samples of those extracted n-grams for 11 file types.

File Type	n	Gram	ASCII Gram
.bmp	2	424D	BM
	4	00280000	.(..
	5	0000280000	..(..
	5	0028000000	.(...
	6	000000280000	...(..
	6	000028000000	..(...
	7	00000028000000	...(...
.doc	20	00010043006F006D0070004F0062006A00000000	...C.o.m.p.O.b.j....
	20	00020000001E00000006000000 5469746C650003	.............Title..
	20	00030000000000000000000000000000000000	....
	20	00060000005469746C650003000000 0100000000	.....Title..........
	20	00060902000000000C0000000000000046000000	..............F...
	20	000C100000020000001E0000006000000546974	..............Tit
	20	00160044006500660061007500 6C007400200050	...D.e.f.a.u.l.t. .P
	20	001E00000060000005469746C65000300000001	.........Title......
	20	00200045006E00740072007900000000000000000	. .E.n.t.r.y........
	20	0020004E006500770020005 2006F006D0061006E	. .N.e.w. .R.o.m.a.n
	20	00200050006100720061006700720061007000 68	. .P.a.r.a.g.r.a.p.h
	20	00310054006100620 06C00650000000000000000	.1.T.a.b.l.e........
	20	0043006F006D0070004F0062006A000000000000	.C.o.m.p.O.b.j......
	20	00440065006600610075006C0074002000500061	.D.e.f.a.u.l.t. .P.a
	20	0045006E00740072007900000000000000000000	.E.n.t.r.y..........
	20	004E006500770020005 2006F006D0061006E0000	.N.e.w. .R.o.m.a.n..
	20	004F0062006A0000000000000000000000000000	.O.b.j..............
	20	00500061007200610067007200 61007000680020	.P.a.r.a.g.r.a.p.h.
	20	00540061006200 6C00650000000000000000000	.T.a.b.l.e..........
	20	00540069006D006500730020004E006500770020	.T.i.m.e.s. .N.e.w.
	20	0057006F007200640044006F00630075006D0065	.W.o.r.d.D.o.c.u.m.e
	20	00610062006C00650000000000000000000000000	.a.b.l.e............
	20	00610067007200610070006800200046006F006E	.a.g.r.a.p.h. .F.o.n

File Type	n	Gram	ASCII Gram
	20	00610070006800200046006F006E007400000000	.a.p.h. .F.o.n.t....
	20	00610072006100670072006100700068002000 46	.a.r.a.g.r.a.p.h. .F
	20	00610075006C00740020005000610072006100 67	.a.u.l.t. .P.a.r.a.g
	20	0062006A0000000000000000000000000000000 0	.b.j..............
	20	0062006C0065000000000000000000000000000 0	.b.l.e..............
	20	00640044006F00630075006D0065006E00740000	.d.D.o.c.u.m.e.n.t..
	20	0065000000000000000000000000000000000000	.e..................
	20	00650066006100750 06C00740020005000610072	.e.f.a.u.l.t. .P.a.r
	20	006500730020004E00650077002000520 06F006D	.e.s. .N.e.w. .R.o.m
	20	0065007700200052006F006D0061006E00000035	.e.w. .R.o.m.a.n...5
	20	006600610075006C00740020005000610072006 1	.f.a.u.l.t. .P.a.r.a
	20	00670072006100700068002000460 06F006E0074	.g.r.a.p.h. .F.o.n.t
	20	0069006D006500730020004E0065007700200052	.i.m.e.s. .N.e.w. .R
.docx	20	00776F72642F5F72656C732F646F63756D656E74	.word/_rels/document
	20	00776F72642F646F63756D656E742E786D6C504B	.word/document.xmlPK
	20	00776F72642F666F6E745461626C652E786D6C50	.word/fontTable.xmlP
	20	00776F72642F73657474696E67732E786D6C504B	.word/settings.xmlPK
	20	00776F72642F7468656D652F7468656D65312E78	.word/theme/theme1.x
	20	00776F72642F77656253657474696E67732E786D	.word/webSettings.xm
	20	01776F72642F5F72656C732F646F63756D656E74	.word/_rels/document
	20	2F646F63756D656E742E786D6C2E72656C7320A2	/document.xml.rels .
	20	2F646F63756D656E742E786D6C2E72656C73504B	/document.xml.relsPK
	20	2F646F63756D656E742E786D6C504B01022D0014	/document.xmlPK..-..
	20	2F73657474696E67732E786D6C504B01022D0014	/settings.xmlPK..-..
	20	2F7468656D652F7468656D65312E786D6CEC594F	/theme/theme1.xml.YO
	20	2F77656253657474696E67732E786D6C504B0102	/webSettings.xmlPK..
	20	7474696E67732E786D6C504B01022D0014000600	ttings.xmlPK..-.....
	20	756D656E742E786D6C2E72656C7320A2040128A0	ument.xml.rels ...(.
	20	756D656E742E786D6C2E72656C73504B01022D00	ument.xml.relsPK..-.
	20	756D656E742E786D6C504B01022D001400060008	ument.xmlPK..-.....
	20	77656253657474696E67732E786D6C504B01022D	webSettings.xmlPK..-
	20	776F72642F5F72656C732F646F63756D656E742E	word/_rels/document.
	20	776F72642F646F63756D656E742E786D6C504B01	word/document.xmlPK.
	20	776F72642F666F6E745461626C652E786D6C504B	word/fontTable.xmlPK
	20	776F72642F73657474696E67732E786D6C504B01	word/settings.xmlPK.
	20	776F72642F7468656D652F7468656D65312E786D	word/theme/theme1.xm

File Type	n	Gram	ASCII Gram
	20	776F72642F77656253657474696E67732E786D6C	word/webSettings.xml
	20	BF000000FFFF0300504B0304140006000800000	.....PK.....
.gif	2	010A	..
	2	3880	8.
	2	4638	F8
	2	4880	H.
	2	7001	p.
	2	8001	..
	2	8002	..
	3	474946	GIF
	3	494638	IF8
	4	47494638	GIF8
.jpg	6	00104A464946	..JFIF
	12	FFD8FFE000104A4649460001	......JFIF..
	12	FFDA000C0301000211031100	.....
.pdf	7	255044462D312E	%PDF-1.
	7	312030206F626A	1 0 obj
	7	322030206F626A	2 0 obj
	7	332030206F626A	3 0 obj
	7	342030206F626A	4 0 obj
	7	352030206F626A	5 0 obj
	7	362030206F626A	6 0 obj
	8	2F436174616C6F67	/Catalog
	8	2F4C656E67746820	/Length
	8	2F53756274797065	/Subtype
	9	2F42617365466F6E74	/BaseFont
	9	2F436F6E74656E7473	/Contents
	9	2F4D65646961426F78	/MediaBox
	10	2F5265736F7572636573	/Resources
	10	2F436F6E74656E747320	/Contents
	11	30303030303030303020	0000000000
	13	2F4372656174696F6E44617465	/CreationDate
.png	5	0D49484452	.IHDR
	5	89504E470D	.PNG.
	8	000049454E44AE42	..IEND.B
	18	89504E470D0A1A0A0000000D494844520000	.PNG........IHDR..

File Type	n	Gram	ASCII Gram
.ppt	20	0000000B000000466F6E74732055736564000300	.......Fonts Used...
	20	00000044657369676E2054656D706C6174650003	...Design Template..
	20	0050006F00770065007200050006F0069006E0074	.P.o.w.e.r.P.o.i.n.t
	20	006F0069006E007400200044006F00630075006D	.o.i.n.t. .D.o.c.u.m
	20	007400200044006F00630075006D0065006E0074	.t. .D.o.c.u.m.e.n.t
.pptx	20	00007070742F7072657350726F70732E786D6C8C	..ppt/presProps.xml.
	20	00007070742F70726573656E746174696F6E2E78	..ppt/presentation.x
	20	00007070742F736C6964654C61796F7574732F5F	..ppt/slideLayouts/_
	20	00007070742F736C6964654C61796F7574732F73	..ppt/slideLayouts/s
	20	00007070742F736C6964654D6173746572732F5F	..ppt/slideMasters/_
	20	00007070742F736C6964654D6173746572732F73	..ppt/slideMasters/s
	20	00007070742F736C696465732F5F72656C732F73	..ppt/slides/_rels/s
	20	00007070742F736C696465732F736C696465312E	..ppt/slides/slide1.
	20	00007070742F736C696465732F736C696465322E	..ppt/slides/slide2.
	20	00007070742F736C696465732F736C696465332E	..ppt/slides/slide3.
	20	00007070742F7461626C655374796C65732E786D	..ppt/tableStyles.xm
	20	00007070742F7468656D652F7468656D65312E78	..ppt/theme/theme1.x
	20	2F7072657350726F70732E786D6C504B01022D00	/presProps.xmlPK..-.
	20	2F70726573656E746174696F6E2E786D6C2E7265	/presentation.xml.re
	20	2F70726573656E746174696F6E2E786D6C504B01	/presentation.xmlPK.
	20	2F736C696465312E786D6C2E72656C73504B0102	/slide1.xml.relsPK..
	20	2F736C696465312E786D6C504B01022D00140006	/slide1.xmlPK..-....
	20	2F736C696465322E786D6C2E72656C73504B0102	/slide2.xml.relsPK..
	20	2F736C696465322E786D6C504B01022D00140006	/slide2.xmlPK..-....
	20	2F736C696465332E786D6C2E72656C73504B0102	/slide3.xml.relsPK..
	20	2F736C696465332E786D6C504B01022D00140006	/slide3.xmlPK..-....
	20	2F736C6964654C61796F7574312E786D6C2E7265	/slideLayout1.xml.re
	20	2F736C6964654C61796F7574312E786D6C504B01	/slideLayout1.xmlPK.
.xls	16	0057006F0072006B0062006F006F006B	.W.o.r.k.b.o.o.k
	16	00576F726B73686565747300030000000	.Worksheets.....
	17	0057006F0072006B0062006F006F006B00	.W.o.r.k.b.o.o.k.
	19	28222422232C2323302E30305C291E04270008	("$"#,##0.00\)..'..
	19	28222422232C2323302E30305C291E0437002A	("$"#,##0.00\)..7.*
	19	28222422232C2323305C291E04210006001C00	("$"#,##0\)..!.....
	19	28222422232C2323305C291E04220007001D00	("$"#,##0\)..".....
	19	282224222A20222D223F3F5F293B5F28405F29	("$"* "-"??_);_(@_)

File Type	n	Gram	ASCII Gram
	19	282224222A20222D225F293B5F28405F291E04	("$"* "-"_);_(@_)..
	19	282224222A20232C2323302E30305F293B5F28	("$"* #,##0.00_);_(
	19	282224222A20232C2323305F293B5F28222422	("$"* #,##0_);_("$"
	19	282224222A205C28232C2323302E30305C293B	("$"* \(#,##0.00\);
	19	282224222A205C28232C2323305C293B5F2822	("$"* \(#,##0\);_("
	19	5B5265645D5C28222422232C2323302E30305C	[Red]\("$"#,##0.00\
	19	5B5265645D5C28222422232C2323305C291E04	[Red]\("$"#,##0\)..
	20	00000057006F0072006B0062006F006F006B0000	...W.o.r.k.b.o.o.k..
	20	00000B000000576F726B73686565747300030000	......Worksheets....
	20	5B5265645D5C28222422232C2323302E30305C29	[Red]\("$"#,##0.00\)
	20	5B5265645D5C28222422232C2323305C291E0422	[Red]\("$"#,##0\).."
.xlsx	20	2F736861726564537472696E67732E786D6C504B	/sharedStrings.xmlPK
	20	2F7468656D65312E786D6C504B01022D00140006	/theme1.xmlPK..-....
	20	2F776F726B626F6F6B2E786D6C2E72656C7320A2	/workbook.xml.rels .
	20	2F776F726B626F6F6B2E786D6C2E72656C73504B	/workbook.xml.relsPK
	20	2F776F726B626F6F6B2E786D6C504B01022D0014	/workbook.xmlPK..-..
	20	2F776F726B7368656574732F7368656574312E78	/worksheets/sheet1.x
	20	736861726564537472696E67732E786D6C504B01	sharedStrings.xmlPK.
	20	6C2F776F726B626F6F6B2E786D6C504B01022D00	l/workbook.xmlPK..-.
	20	6C2F776F726B7368656574732F7368656574312E	l/worksheets/sheet1.

Appendix D: Prediction Result using 4 File Types with 10 Files each

These are the results of Prediction Process towards NEW-EXP-0 data set. The n-grams that used in Learning Process were extracted from 40 files that represent only 4 file types (BMP, PNG, JPG, and GIF) with 10 files each. The percentage in cell indicates the matching rates. This matching rate was calculated with (the number of extracted n-grams for each file type that exist on each file in column Filename) divided by (the total of extracted n-grams from Learning Process).

Filename	.bmp	.png	.jpg	.gif	type
048569.bmp	95.45%	2.74%	5.41%	9.64%	.bmp
186118.bmp	86.36%	0.00%	2.32%	2.41%	.bmp
192402.bmp	95.45%	10.50%	12.36%	33.73%	.bmp
295989.bmp	95.45%	7.76%	3.86%	19.28%	.bmp
354675.bmp	95.45%	7.31%	6.95%	22.89%	.bmp
424240.bmp	90.91%	8.22%	13.51%	38.55%	.bmp
811459.bmp	100.00%	3.65%	5.41%	10.84%	.bmp
874196.bmp	86.36%	17.81%	30.89%	91.57%	.bmp
486901.doc	59.09%	4.11%	10.81%	28.92%	.doc
580687.doc	95.45%	10.05%	98.46%	51.81%	.doc
643587.doc	95.45%	12.79%	27.80%	69.88%	.doc
687675.doc	90.91%	3.65%	14.67%	33.73%	.doc
880064.doc	81.82%	5.02%	15.06%	33.73%	.doc
120257.gif	13.64%	7.31%	10.04%	53.01%	.gif
519722.gif	9.09%	4.11%	8.49%	42.17%	.gif
589990.gif	0.00%	1.37%	2.32%	20.48%	.gif
823904.gif	18.18%	6.39%	12.36%	98.80%	.gif
827478.gif	59.09%	7.76%	20.85%	98.80%	.gif
875216.gif	9.09%	6.39%	11.20%	98.80%	.gif
912800.gif	18.18%	3.65%	8.11%	45.78%	.gif
919144.gif	22.73%	5.94%	11.20%	72.29%	.gif
924835.gif	22.73%	12.79%	22.39%	100.00%	.gif
956880.gif	0.00%	0.00%	2.32%	16.87%	.gif
670890.jpg	22.73%	10.96%	60.62%	85.54%	.jpg
672116.jpg	45.45%	12.79%	98.46%	60.24%	.jpg
710340.jpg	36.36%	5.94%	91.51%	33.73%	.jpg
739365.jpg	27.27%	10.50%	94.59%	84.34%	.jpg

Filename	.bmp	.png	.jpg	.gif	type
827462.jpg	50.00%	13.24%	96.53%	92.77%	.jpg
907395.jpg	27.27%	4.57%	92.66%	36.14%	.jpg
945965.jpg	40.91%	11.87%	97.68%	92.77%	.jpg
949268.jpg	27.27%	13.24%	100.00%	93.98%	.jpg
956747.jpg	18.18%	4.57%	89.58%	28.92%	.jpg
987041.jpg	27.27%	11.87%	100.00%	93.98%	.jpg
184216.pdf	22.73%	8.68%	100.00%	71.08%	.pdf
191846.pdf	18.18%	90.36%	11.87%	20.46%	.pdf
469082.pdf	4.57%	6.95%	18.18%	45.78%	.pdf
614817.pdf	31.82%	92.77%	12.33%	74.90%	.pdf
624005.pdf	4.11%	74.52%	18.18%	54.22%	.pdf
714314.pdf	3.86%	5.48%	0.00%	37.35%	.pdf
756794.pdf	16.99%	83.13%	11.87%	18.18%	.pdf
764163.pdf	2.70%	4.55%	13.25%	2.74%	.pdf
858257.pdf	54.22%	9.09%	12.74%	7.76%	.pdf
905600.pdf	1.16%	6.02%	0.00%	1.37%	.pdf
270700.png	18.18%	100.00%	19.69%	90.36%	.png
365624.png	18.18%	100.00%	13.13%	65.06%	.png
705948.png	13.64%	100.00%	13.13%	72.29%	.png
720718.png	13.64%	100.00%	15.83%	71.08%	.png
821535.png	13.64%	100.00%	10.42%	36.14%	.png
862652.png	13.64%	100.00%	15.44%	72.29%	.png
920125.png	18.18%	100.00%	18.92%	92.77%	.png
920697.png	18.18%	100.00%	20.08%	92.77%	.png
920706.png	18.18%	100.00%	20.85%	91.57%	.png
942196.png	0.00%	100.00%	6.95%	18.07%	.png
130790.ppt	95.45%	100.00%	94.98%	72.29%	.ppt
250560.ppt	90.91%	9.59%	25.87%	63.86%	.ppt
307260.ppt	100.00%	100.00%	100.00%	97.59%	.ppt
388207.ppt	90.91%	100.00%	20.46%	56.63%	.ppt
547335.ppt	100.00%	20.55%	100.00%	96.39%	.ppt
593251.ppt	100.00%	100.00%	29.34%	85.54%	.ppt
623300.ppt	100.00%	100.00%	100.00%	96.39%	.ppt
717834.ppt	90.91%	6.85%	17.37%	48.19%	.ppt
948335.ppt	100.00%	100.00%	100.00%	97.59%	.ppt
388283.xls	90.91%	6.85%	19.31%	48.19%	.xls
458821.xls	95.45%	8.22%	19.31%	53.01%	.xls

Filename	.bmp	.png	.jpg	.gif	type
505158.xls	90.91%	8.22%	14.29%	43.37%	.xls
817954.xls	90.91%	8.22%	15.06%	53.01%	.xls
901341.xls	90.91%	7.31%	12.36%	39.76%	.xls
216689.docx	22.73%	9.13%	99.23%	55.42%	docx
223624.docx	81.82%	100.00%	100.00%	95.18%	docx
299759.docx	31.82%	9.59%	17.37%	44.58%	docx
313486.docx	77.27%	100.00%	100.00%	100.00%	docx
372437.docx	40.91%	100.00%	100.00%	92.77%	docx
598299.docx	31.82%	100.00%	20.46%	78.31%	docx
605377.docx	13.64%	0.91%	6.95%	21.69%	docx
615326.docx	18.18%	5.48%	10.04%	16.87%	docx
182614.pptx	45.45%	100.00%	100.00%	96.39%	pptx
263566.pptx	59.09%	100.00%	100.00%	96.39%	pptx
318133.pptx	86.36%	100.00%	100.00%	100.00%	pptx
596250.pptx	95.45%	100.00%	35.14%	100.00%	pptx
606936.pptx	36.36%	100.00%	100.00%	96.39%	pptx
609019.pptx	59.09%	100.00%	100.00%	96.39%	pptx
729601.pptx	100.00%	100.00%	100.00%	96.39%	pptx
978134.pptx	40.91%	100.00%	98.84%	78.31%	pptx
996017.pptx	100.00%	100.00%	100.00%	100.00%	pptx
212946.xlsx	31.82%	13.24%	23.17%	92.77%	xlsx
299765.xlsx	22.73%	3.65%	9.27%	27.71%	xlsx
411102.xlsx	18.18%	10.96%	94.59%	72.29%	xlsx
500968.xlsx	95.45%	7.76%	19.69%	49.40%	xlsx
528206.xlsx	18.18%	4.11%	8.11%	22.89%	xlsx
848990.xlsx	13.64%	4.11%	9.65%	27.71%	xlsx

Appendix E: Prediction Result using 4 File Types with 20 Files each

These are the results of Prediction Process towards NEW-EXP-0 data set. The n-grams that used in Learning Process were extracted from 80 files that represent only 4 file types (BMP, PNG, JPG, and GIF) with 20 files each. The percentage in cell indicates the matching rates. This matching rate was calculated with (the number of extracted n-grams for each file type that exist on each file in column Filename) divided by (the total of extracted n-grams from Learning Process).

Filename	.bmp	.png	.jpg	.gif	type
048569.bmp	100.00%	2.74%	4.95%	6.67%	.bmp
186118.bmp	86.36%	0.00%	3.47%	3.33%	.bmp
192402.bmp	95.45%	10.50%	11.88%	20.00%	.bmp
295989.bmp	95.45%	7.76%	2.48%	13.33%	.bmp
354675.bmp	100.00%	7.31%	5.45%	13.33%	.bmp
424240.bmp	95.45%	8.22%	13.86%	26.67%	.bmp
811459.bmp	100.00%	3.65%	2.48%	10.00%	.bmp
874196.bmp	86.36%	17.81%	28.71%	86.67%	.bmp
486901.doc	63.64%	4.11%	12.38%	50.00%	.doc
580687.doc	100.00%	10.05%	100.00%	63.33%	.doc
643587.doc	100.00%	12.79%	28.71%	80.00%	.doc
687675.doc	95.45%	3.65%	14.36%	56.67%	.doc
880064.doc	86.36%	5.02%	16.83%	50.00%	.doc
120257.gif	13.64%	7.31%	10.40%	83.33%	.gif
519722.gif	4.55%	4.11%	6.93%	53.33%	.gif
589990.gif	0.00%	1.37%	2.97%	46.67%	.gif
823904.gif	13.64%	6.39%	13.37%	100.00%	.gif
827478.gif	59.09%	7.76%	21.78%	100.00%	.gif
875216.gif	9.09%	6.39%	13.37%	100.00%	.gif
912800.gif	18.18%	3.65%	6.93%	60.00%	.gif
919144.gif	22.73%	5.94%	10.40%	76.67%	.gif
924835.gif	18.18%	12.79%	21.78%	100.00%	.gif
956880.gif	0.00%	0.00%	2.97%	36.67%	.gif
670890.jpg	22.73%	10.96%	62.38%	76.67%	.jpg
672116.jpg	45.45%	12.79%	100.00%	63.33%	.jpg

Filename	.bmp	.png	.jpg	.gif	type
710340.jpg	40.91%	5.94%	99.01%	53.33%	.jpg
739365.jpg	31.82%	10.50%	99.51%	73.33%	.jpg
827462.jpg	50.00%	13.24%	100.00%	80.00%	.jpg
907395.jpg	31.82%	4.57%	98.51%	30.00%	.jpg
945965.jpg	40.91%	11.87%	99.01%	80.00%	.jpg
949268.jpg	27.27%	13.24%	100.00%	83.33%	.jpg
956747.jpg	18.18%	4.57%	96.04%	40.00%	.jpg
987041.jpg	27.27%	11.87%	100.00%	83.33%	.jpg
184216.pdf	22.73%	8.68%	100.00%	66.67%	.pdf
191846.pdf	13.64%	11.87%	19.31%	76.67%	.pdf
469082.pdf	13.64%	4.57%	7.43%	43.33%	.pdf
614817.pdf	31.82%	12.33%	72.77%	80.00%	.pdf
624005.pdf	18.18%	4.11%	70.79%	50.00%	.pdf
714314.pdf	0.00%	5.48%	4.46%	33.33%	.pdf
756794.pdf	13.64%	11.87%	14.85%	63.33%	.pdf
764163.pdf	4.55%	2.74%	2.48%	6.67%	.pdf
858257.pdf	9.09%	7.76%	12.87%	50.00%	.pdf
905600.pdf	0.00%	1.37%	1.49%	10.00%	.pdf
270700.png	13.64%	100.00%	18.32%	76.67%	.png
365624.png	18.18%	100.00%	14.36%	63.33%	.png
705948.png	9.09%	100.00%	11.88%	53.33%	.png
720718.png	13.64%	100.00%	13.86%	63.33%	.png
821535.png	9.09%	100.00%	11.88%	46.67%	.png
862652.png	9.09%	100.00%	13.86%	66.67%	.png
920125.png	13.64%	100.00%	18.32%	80.00%	.png
920697.png	13.64%	100.00%	18.81%	80.00%	.png
920706.png	13.64%	100.00%	20.30%	80.00%	.png
942196.png	4.55%	100.00%	5.94%	13.33%	.png
130790.ppt	95.45%	100.00%	99.51%	86.67%	.ppt
250560.ppt	95.45%	9.59%	26.73%	76.67%	.ppt
307260.ppt	100.00%	100.00%	100.00%	93.33%	.ppt
388207.ppt	95.45%	100.00%	21.29%	73.33%	.ppt
547335.ppt	100.00%	20.55%	100.00%	90.00%	.ppt
593251.ppt	100.00%	100.00%	28.22%	83.33%	.ppt
623300.ppt	100.00%	100.00%	100.00%	90.00%	.ppt
717834.ppt	95.45%	6.85%	16.83%	60.00%	.ppt
948335.ppt	100.00%	100.00%	100.00%	93.33%	.ppt

Filename	.bmp	.png	.jpg	.gif	type
388283.xls	95.45%	6.85%	19.80%	70.00%	.xls
458821.xls	100.00%	8.22%	19.80%	73.33%	.xls
505158.xls	95.45%	8.22%	16.34%	66.67%	.xls
817954.xls	95.45%	8.22%	17.33%	70.00%	.xls
901341.xls	95.45%	7.31%	14.85%	53.33%	.xls
216689.docx	27.27%	9.13%	100.00%	56.67%	docx
223624.docx	81.82%	100.00%	100.00%	86.67%	docx
299759.docx	31.82%	9.59%	17.33%	40.00%	docx
313486.docx	77.27%	100.00%	100.00%	100.00%	docx
372437.docx	40.91%	100.00%	100.00%	80.00%	docx
598299.docx	31.82%	100.00%	20.30%	73.33%	docx
605377.docx	18.18%	0.91%	9.41%	16.67%	docx
615326.docx	22.73%	5.48%	11.88%	13.33%	docx
182614.pptx	45.45%	100.00%	100.00%	90.00%	pptx
263566.pptx	59.09%	100.00%	100.00%	90.00%	pptx
318133.pptx	86.36%	100.00%	100.00%	100.00%	pptx
596250.pptx	95.45%	100.00%	33.66%	100.00%	pptx
606936.pptx	36.36%	100.00%	100.00%	90.00%	pptx
609019.pptx	59.09%	100.00%	100.00%	90.00%	pptx
729601.pptx	100.00%	100.00%	100.00%	90.00%	pptx
978134.pptx	40.91%	100.00%	100.00%	76.67%	pptx
996017.pptx	100.00%	100.00%	100.00%	100.00%	pptx
212946.xlsx	31.82%	13.24%	22.28%	80.00%	xlsx
299765.xlsx	22.73%	3.65%	10.89%	30.00%	xlsx
411102.xlsx	22.73%	10.96%	100.00%	66.67%	xlsx
500968.xlsx	100.00%	7.76%	20.79%	70.00%	xlsx
528206.xlsx	22.73%	4.11%	9.41%	26.67%	xlsx
848990.xlsx	18.18%	4.11%	12.38%	26.67%	xlsx

Appendix F: Prediction Result using 11 File Types with 10 Files each

These are the results of Prediction Process towards NEW-EXP-0 data set. The n-grams that used in Learning Process were extracted from 110 files that represent 11 file types as stated in scope of work from this thesis with 10 files each. The percentage in cell indicates the matching rates. This matching rate was calculated with (the number of extracted n-grams for each file type that exist on each file in column Filename) divided by (the total of extracted n-grams from Learning Process).

Filename	.bmp	.doc	.docx	.gif	.jpg	.pdf	.png	.ppt	.pptx	.xls	.xlsx	FileExt
048569.bmp	50.00%	0.38%	0.53%	7.50%	0.99%	0.44%	1.45%	0.52%	0.36%	0.18%	0.07%	.bmp
186118.bmp	50.00%	0.11%	0.00%	2.50%	0.00%	0.00%	0.00%	0.07%	0.02%	0.03%	0.00%	.bmp
192402.bmp	100.00%	0.91%	13.56%	35.00%	7.43%	18.41%	8.70%	0.43%	13.52%	0.55%	2.08%	.bmp
295989.bmp	100.00%	0.30%	11.43%	32.50%	2.48%	16.56%	7.25%	0.22%	10.76%	0.40%	1.63%	.bmp
354675.bmp	50.00%	0.65%	6.33%	32.50%	2.97%	12.51%	6.28%	0.24%	5.88%	0.42%	1.04%	.bmp
424240.bmp	50.00%	1.35%	10.45%	32.50%	5.94%	11.10%	6.28%	0.67%	9.97%	0.69%	1.44%	.bmp
811459.bmp	100.00%	0.23%	2.57%	7.50%	2.48%	10.22%	3.86%	0.19%	2.67%	0.23%	0.48%	.bmp
874196.bmp	100.00%	4.13%	19.89%	90.00%	13.86%	24.67%	13.04%	2.93%	18.13%	2.49%	3.19%	.bmp
486901.doc	50.00%	25.94%	0.82%	7.50%	0.00%	9.07%	0.48%	2.39%	0.39%	0.84%	0.08%	.doc
580687.doc	50.00%	99.75%	6.61%	30.00%	98.51%	22.56%	5.80%	9.22%	6.72%	2.52%	1.07%	.doc
643587.doc	50.00%	99.03%	11.60%	47.50%	8.91%	35.33%	8.21%	13.31%	9.62%	4.99%	1.62%	.doc
687675.doc	0.00%	98.57%	1.55%	5.00%	0.99%	7.67%	0.00%	3.50%	0.61%	1.43%	0.06%	.doc
880064.doc	0.00%	92.24%	0.65%	12.50%	1.49%	7.05%	1.45%	4.06%	0.34%	1.70%	0.08%	.doc
120257.gif	50.00%	0.67%	12.17%	45.00%	2.48%	7.58%	5.80%	0.47%	10.92%	0.46%	1.75%	.gif
519722.gif	50.00%	0.55%	8.29%	47.50%	3.96%	6.08%	2.90%	0.35%	7.92%	0.30%	1.26%	.gif
589990.gif	0.00%	0.08%	0.82%	15.00%	0.50%	0.70%	0.00%	0.10%	0.73%	0.05%	0.06%	.gif

Filename	.bmp	.doc	.docx	.gif	.jpg	.pdf	.png	.ppt	.pptx	.xls	.xlsx	FileExt
823904.gif	50.00%	1.01%	6.82%	97.50%	1.98%	5.29%	3.86%	0.82%	5.70%	0.97%	1.04%	.gif
827478.gif	50.00%	1.64%	7.43%	97.50%	6.44%	6.08%	4.35%	1.62%	7.05%	1.83%	1.22%	.gif
875216.gif	50.00%	0.70%	4.04%	100.00%	2.97%	4.58%	3.38%	0.57%	4.36%	0.68%	0.77%	.gif
912800.gif	50.00%	0.70%	9.84%	42.50%	2.97%	7.05%	2.42%	0.36%	9.78%	0.33%	1.56%	.gif
919144.gif	100.00%	0.78%	14.50%	67.50%	4.95%	10.22%	4.83%	0.50%	14.33%	0.54%	2.23%	.gif
924835.gif	100.00%	1.94%	26.58%	100.00%	10.40%	19.03%	9.66%	1.60%	25.48%	1.47%	4.05%	.gif
956880.gif	0.00%	0.02%	1.67%	17.50%	0.50%	1.06%	0.00%	0.10%	1.49%	0.07%	0.21%	.gif
670890.jpg	100.00%	1.41%	25.97%	85.00%	51.49%	17.44%	8.70%	1.06%	24.76%	0.96%	3.75%	.jpg
672116.jpg	100.00%	1.24%	17.56%	37.50%	99.01%	12.33%	9.18%	1.56%	16.85%	1.10%	2.50%	.jpg
710340.jpg	0.00%	0.65%	2.49%	10.00%	93.56%	3.70%	3.86%	0.69%	2.90%	0.49%	0.41%	.jpg
739365.jpg	0.00%	1.60%	22.25%	80.00%	95.54%	15.15%	7.73%	1.05%	20.99%	1.02%	3.21%	.jpg
827462.jpg	100.00%	1.92%	26.79%	92.50%	96.04%	19.82%	10.14%	1.67%	25.76%	1.58%	4.26%	.jpg
907395.jpg	0.00%	0.57%	8.86%	27.50%	93.07%	5.82%	3.86%	0.35%	8.80%	0.32%	1.36%	.jpg
945965.jpg	100.00%	1.75%	26.66%	92.50%	100.00%	19.65%	9.18%	1.33%	25.79%	1.40%	4.19%	.jpg
949268.jpg	100.00%	1.45%	26.58%	92.50%	100.00%	19.12%	9.66%	1.17%	25.59%	1.12%	4.06%	.jpg
956747.jpg	100.00%	1.35%	9.31%	12.50%	89.11%	5.46%	1.93%	0.67%	9.31%	0.67%	1.24%	.jpg
987041.jpg	100.00%	1.52%	26.50%	92.50%	100.00%	18.50%	9.18%	1.21%	25.65%	1.22%	4.03%	.jpg
184216.pdf	50.00%	1.16%	21.48%	62.50%	100.00%	96.65%	6.76%	0.90%	20.21%	0.78%	3.10%	.pdf
191846.pdf	100.00%	1.58%	25.97%	92.50%	8.91%	99.91%	9.18%	1.15%	25.14%	1.09%	3.91%	.pdf
469082.pdf	100.00%	1.26%	10.66%	35.00%	2.97%	99.12%	3.86%	0.51%	10.46%	0.68%	1.57%	.pdf
614817.pdf	100.00%	3.22%	27.81%	92.50%	69.31%	99.82%	9.66%	1.63%	25.86%	1.30%	4.12%	.pdf
624005.pdf	50.00%	1.37%	12.09%	50.00%	70.79%	94.98%	3.86%	0.68%	11.20%	0.64%	1.65%	.pdf
714314.pdf	0.00%	0.89%	9.43%	35.00%	0.50%	98.06%	4.83%	0.49%	8.61%	0.32%	1.29%	.pdf
756794.pdf	100.00%	1.54%	24.91%	87.50%	7.92%	94.10%	9.18%	1.08%	24.21%	0.88%	3.76%	.pdf

Filename	.bmp	.doc	.docx	.gif	.jpg	.pdf	.png	.ppt	.pptx	.xls	.xlsx	FileExt
764163.pdf	0.00%	0.97%	5.47%	17.50%	0.99%	93.39%	2.42%	0.40%	5.32%	0.16%	0.81%	.pdf
858257.pdf	0.00%	0.86%	18.05%	55.00%	5.45%	99.47%	6.28%	0.75%	17.47%	0.92%	2.70%	.pdf
905600.pdf	0.00%	0.11%	1.88%	2.50%	0.50%	91.10%	1.45%	0.21%	1.58%	0.03%	0.20%	.pdf
270700.png	100.00%	1.54%	24.34%	92.50%	7.92%	18.24%	100.00%	0.98%	23.18%	0.96%	3.63%	.png
365624.png	50.00%	1.01%	16.29%	65.00%	5.45%	12.16%	100.00%	0.65%	14.93%	0.56%	2.24%	.png
705948.png	100.00%	1.01%	20.25%	77.50%	5.45%	13.66%	100.00%	0.73%	19.96%	0.72%	3.04%	.png
720718.png	50.00%	0.95%	18.66%	65.00%	7.43%	13.74%	100.00%	0.63%	17.68%	0.61%	2.73%	.png
821535.png	50.00%	0.70%	5.43%	20.00%	3.47%	5.37%	100.00%	0.27%	4.95%	0.34%	0.80%	.png
862652.png	50.00%	1.24%	18.66%	70.00%	5.94%	13.04%	100.00%	0.79%	18.58%	0.75%	2.87%	.png
920125.png	100.00%	1.50%	26.05%	92.50%	7.43%	17.62%	100.00%	1.02%	25.12%	0.99%	3.86%	.png
920697.png	100.00%	1.60%	26.01%	92.50%	7.92%	17.80%	100.00%	1.09%	25.19%	0.99%	3.90%	.png
920706.png	100.00%	1.64%	26.38%	90.00%	7.92%	17.89%	100.00%	1.09%	25.31%	1.01%	3.92%	.png
942196.png	0.00%	0.11%	1.39%	20.00%	1.49%	1.67%	100.00%	0.19%	1.11%	0.08%	0.19%	.png
130790.ppt	50.00%	37.09%	17.31%	52.50%	94.06%	22.82%	100.00%	75.57%	15.65%	3.62%	2.44%	.ppt
250560.ppt	0.00%	35.22%	3.72%	40.00%	8.42%	17.53%	5.80%	78.08%	3.45%	5.76%	0.68%	.ppt
307260.ppt	100.00%	39.24%	29.97%	95.00%	100.00%	36.65%	100.00%	92.99%	27.90%	6.53%	7.03%	.ppt
388207.ppt	0.00%	35.51%	5.51%	27.50%	4.46%	8.46%	100.00%	80.80%	4.96%	3.41%	0.75%	.ppt
547335.ppt	100.00%	27.63%	27.81%	95.00%	100.00%	29.43%	15.94%	65.35%	26.53%	4.79%	4.98%	.ppt
593251.ppt	100.00%	37.62%	27.52%	77.50%	10.40%	31.19%	100.00%	70.34%	24.83%	4.54%	3.90%	.ppt
623300.ppt	100.00%	40.25%	28.58%	92.50%	100.00%	31.10%	100.00%	89.97%	26.51%	5.98%	4.69%	.ppt
717834.ppt	0.00%	31.74%	6.33%	22.50%	0.99%	6.61%	2.90%	61.58%	6.09%	2.07%	0.91%	.ppt
948335.ppt	100.00%	39.28%	29.69%	95.00%	100.00%	31.63%	100.00%	91.17%	27.07%	5.43%	5.65%	.ppt
388283.xls	0.00%	18.99%	1.96%	15.00%	3.96%	21.32%	1.93%	5.29%	1.21%	98.03%	0.25%	.xls
458821.xls	50.00%	29.27%	1.51%	30.00%	4.95%	13.30%	3.38%	5.41%	1.59%	96.76%	0.31%	.xls

Filename	.bmp	.doc	.docx	.gif	.jpg	.pdf	.png	.ppt	.pptx	.xls	.xlsx	FileExt
505158.xls	0.00%	23.73%	2.04%	12.50%	0.50%	15.59%	3.86%	4.84%	0.78%	93.01%	0.17%	.xls
817954.xls	0.00%	19.87%	1.84%	22.50%	1.49%	20.18%	3.38%	4.50%	1.64%	100.00%	0.29%	.xls
901341.xls	0.00%	22.00%	0.57%	10.00%	0.00%	3.70%	2.90%	2.26%	0.34%	99.76%	0.07%	.xls
216689.docx	0.00%	1.16%	98.16%	47.50%	99.01%	11.19%	4.83%	0.74%	14.03%	0.61%	5.49%	docx
223624.docx	100.00%	3.41%	100.00%	92.50%	100.00%	23.61%	100.00%	2.80%	26.82%	2.29%	7.55%	docx
299759.docx	100.00%	1.12%	97.51%	45.00%	5.45%	9.34%	5.31%	0.63%	13.63%	0.47%	5.29%	docx
313486.docx	100.00%	4.76%	100.00%	100.00%	100.00%	24.93%	100.00%	3.85%	28.36%	3.74%	8.33%	docx
372437.docx	100.00%	1.85%	100.00%	92.50%	100.00%	19.03%	100.00%	1.40%	26.19%	1.36%	7.27%	docx
598299.docx	50.00%	1.37%	100.00%	75.00%	7.43%	14.19%	100.00%	0.86%	19.90%	0.77%	6.27%	docx
605377.docx	0.00%	0.21%	100.00%	17.50%	0.00%	1.67%	0.00%	0.22%	2.61%	0.10%	3.74%	docx
615326.docx	50.00%	0.34%	100.00%	10.00%	0.99%	3.88%	4.35%	0.35%	5.61%	0.24%	4.32%	docx
182614.pptx	100.00%	2.61%	27.36%	92.50%	100.00%	21.59%	100.00%	2.39%	97.13%	2.13%	4.71%	pptx
263566.pptx	100.00%	4.36%	27.81%	92.50%	100.00%	24.23%	100.00%	3.50%	75.57%	2.62%	4.62%	pptx
318133.pptx	100.00%	5.31%	29.56%	100.00%	100.00%	25.29%	100.00%	3.88%	89.42%	3.93%	7.80%	pptx
596250.pptx	100.00%	5.56%	29.40%	100.00%	16.83%	25.37%	100.00%	4.11%	99.49%	4.09%	7.89%	pptx
606936.pptx	100.00%	2.78%	27.52%	92.50%	100.00%	20.62%	100.00%	1.96%	99.72%	2.04%	4.66%	pptx
609019.pptx	100.00%	4.07%	27.64%	92.50%	100.00%	20.97%	100.00%	2.97%	73.71%	2.41%	7.23%	pptx
729601.pptx	100.00%	6.24%	28.26%	92.50%	100.00%	29.60%	100.00%	3.65%	69.74%	3.19%	5.11%	pptx
978134.pptx	100.00%	1.77%	22.66%	75.00%	99.01%	15.77%	100.00%	1.27%	94.45%	1.05%	5.75%	pptx
996017.pptx	100.00%	4.87%	29.40%	100.00%	100.00%	28.46%	100.00%	3.40%	78.69%	3.05%	5.32%	pptx
212946.xlsx	100.00%	1.79%	27.40%	92.50%	7.92%	18.41%	9.18%	1.15%	26.86%	1.03%	100.00%	xlsx
299765.xlsx	50.00%	0.44%	6.08%	20.00%	1.98%	4.14%	1.45%	0.22%	5.86%	0.38%	5.91%	xlsx
411102.xlsx	0.00%	1.18%	18.58%	70.00%	93.56%	11.72%	7.25%	0.89%	17.55%	1.15%	99.82%	xlsx
500968.xlsx	50.00%	29.08%	2.12%	17.50%	4.95%	26.26%	3.38%	6.70%	1.45%	92.17%	0.27%	xlsx

Filename	.bmp	.doc	.docx	.gif	.jpg	.pdf	.png	.ppt	.pptx	.xls	.xlsx	FileExt
528206.xlsx	0.00%	0.32%	4.25%	15.00%	0.50%	2.29%	1.93%	0.16%	3.54%	0.21%	99.91%	xlsx
848990.xlsx	0.00%	0.46%	6.45%	22.50%	0.99%	4.93%	0.97%	0.30%	6.33%	0.30%	99.87%	xlsx

Appendix G: Prediction Result using 11 File Types with 20 Files each

These are the results of Prediction Process towards NEW-EXP-0 data set. The n-grams that used in Learning Process were extracted from 194 files that represent 11 file types as stated in scope of work from this thesis with 20 files each. The percentage in cell indicates the matching rates. This matching rate was calculated with (the number of extracted n-grams for each file type that exist on each file in column Filename) divided by (the total of extracted n-grams from Learning Process).

Filename	.bmp	.doc	.docx	.gif	.jpg	.pdf	.png	.ppt	.pptx	.xls	.xlsx	filetype
048569.bmp	100.00%	0.26%	0.72%	0.00%	1.25%	0.32%	1.44%	1.67%	0.40%	0.18%	0.23%	.bmp
186118.bmp	100.00%	0.24%	0.00%	0.00%	0.00%	0.00%	0.00%	0.21%	0.02%	0.03%	0.00%	.bmp
192402.bmp	100.00%	0.75%	15.74%	10.00%	5.63%	13.53%	9.13%	1.23%	13.90%	0.70%	1.22%	.bmp
295989.bmp	100.00%	0.32%	12.92%	30.00%	1.25%	11.27%	7.69%	0.61%	11.05%	0.49%	0.98%	.bmp
354675.bmp	100.00%	0.73%	7.16%	30.00%	1.88%	8.53%	6.73%	0.66%	6.06%	0.47%	0.81%	.bmp
424240.bmp	100.00%	1.21%	12.09%	0.00%	5.00%	7.25%	6.73%	1.71%	10.27%	0.83%	0.81%	.bmp
811459.bmp	100.00%	0.28%	2.94%	0.00%	0.63%	5.48%	3.85%	0.57%	2.80%	0.29%	0.41%	.bmp
874196.bmp	71.43%	4.08%	22.60%	80.00%	12.50%	23.35%	13.46%	6.44%	18.71%	2.27%	2.66%	.bmp
486901.doc	14.29%	28.81%	1.02%	10.00%	0.63%	12.24%	0.48%	9.34%	0.47%	0.93%	1.10%	.doc
580687.doc	100.00%	99.78%	7.68%	30.00%	100.00%	21.58%	5.77%	22.47%	7.00%	2.73%	4.22%	.doc
643587.doc	100.00%	98.10%	12.99%	40.00%	11.25%	35.43%	8.65%	28.56%	10.06%	4.71%	4.57%	.doc
687675.doc	85.71%	99.96%	1.66%	20.00%	1.25%	10.31%	0.00%	11.59%	0.71%	1.52%	0.52%	.doc
880064.doc	57.14%	96.79%	0.83%	20.00%	2.50%	6.92%	1.44%	9.54%	0.42%	1.77%	0.87%	.doc
120257.gif	14.29%	0.56%	13.75%	80.00%	2.50%	5.48%	6.25%	1.15%	11.26%	0.55%	0.64%	.gif
519722.gif	0.00%	0.41%	9.42%	60.00%	3.13%	4.35%	3.37%	0.84%	8.16%	0.38%	0.41%	.gif
589990.gif	0.00%	0.19%	0.98%	40.00%	0.63%	0.32%	0.00%	0.27%	0.75%	0.05%	0.12%	.gif

Filename	.bmp	.doc	.docx	.gif	.jpg	.pdf	.png	.ppt	.pptx	.xls	.xlsx	filetype
823904.gif	0.00%	0.91%	7.57%	100.00%	1.88%	4.35%	3.85%	1.94%	5.92%	1.04%	0.52%	.gif
827478.gif	14.29%	1.75%	8.55%	100.00%	7.50%	4.67%	4.33%	3.85%	7.33%	1.83%	0.64%	.gif
875216.gif	0.00%	0.60%	4.90%	100.00%	3.13%	4.51%	3.37%	1.36%	4.49%	0.79%	0.41%	.gif
912800.gif	0.00%	0.56%	11.19%	60.00%	1.25%	5.15%	2.88%	0.94%	10.03%	0.40%	0.64%	.gif
919144.gif	14.29%	0.67%	16.38%	80.00%	3.13%	7.73%	5.29%	1.29%	14.74%	0.66%	0.69%	.gif
924835.gif	14.29%	1.77%	29.94%	100.00%	8.75%	14.33%	10.10%	4.14%	26.20%	1.66%	1.45%	.gif
956880.gif	0.00%	0.02%	1.69%	40.00%	0.63%	0.64%	0.00%	0.31%	1.54%	0.08%	0.06%	.gif
670890.jpg	14.29%	1.19%	29.23%	60.00%	54.38%	13.53%	9.13%	2.68%	25.47%	1.13%	1.27%	.jpg
672116.jpg	14.29%	1.21%	19.47%	30.00%	100.00%	8.53%	9.62%	3.64%	17.37%	1.18%	1.10%	.jpg
710340.jpg	0.00%	0.82%	2.90%	10.00%	100.00%	2.42%	3.85%	1.48%	3.02%	0.59%	0.00%	.jpg
739365.jpg	0.00%	1.34%	24.75%	50.00%	100.00%	11.11%	8.17%	2.78%	21.58%	1.13%	1.27%	.jpg
827462.jpg	14.29%	1.64%	30.13%	70.00%	100.00%	15.46%	10.58%	4.17%	26.51%	1.59%	1.50%	.jpg
907395.jpg	0.00%	0.43%	9.79%	30.00%	100.00%	4.19%	3.85%	0.96%	9.06%	0.41%	0.35%	.jpg
945965.jpg	14.29%	1.51%	30.02%	70.00%	100.00%	14.17%	9.62%	3.46%	26.53%	1.54%	1.50%	.jpg
949268.jpg	14.29%	1.21%	29.91%	70.00%	100.00%	14.17%	10.10%	3.03%	26.32%	1.38%	1.45%	.jpg
956747.jpg	14.29%	1.42%	10.85%	10.00%	95.63%	3.06%	1.92%	1.75%	9.57%	0.75%	0.41%	.jpg
987041.jpg	14.29%	1.40%	29.87%	70.00%	100.00%	13.53%	9.62%	3.06%	26.38%	1.46%	1.50%	.jpg
184216.pdf	0.00%	1.03%	23.95%	50.00%	100.00%	98.87%	7.21%	2.41%	20.79%	0.98%	1.10%	.pdf
191846.pdf	14.29%	1.47%	29.27%	70.00%	6.88%	100.00%	9.62%	2.90%	25.86%	1.35%	1.50%	.pdf
469082.pdf	14.29%	1.88%	12.09%	50.00%	2.50%	100.00%	3.85%	1.36%	10.79%	0.90%	0.87%	.pdf
614817.pdf	14.29%	2.98%	31.15%	70.00%	66.88%	100.00%	10.10%	4.04%	26.60%	1.56%	1.85%	.pdf
624005.pdf	14.29%	1.10%	14.05%	40.00%	65.63%	100.00%	3.85%	1.84%	11.59%	0.77%	0.75%	.pdf
714314.pdf	0.00%	0.80%	10.55%	20.00%	1.25%	100.00%	4.81%	1.34%	8.90%	0.39%	0.81%	.pdf
756794.pdf	14.29%	1.25%	27.95%	60.00%	5.63%	99.68%	9.62%	2.62%	24.89%	1.12%	1.50%	.pdf

Filename	.bmp	.doc	.docx	.gif	.jpg	.pdf	.png	.ppt	.pptx	.xls	.xlsx	filetype
764163.pdf	0.00%	0.88%	5.95%	20.00%	1.25%	100.00%	2.40%	1.12%	5.53%	0.21%	0.41%	.pdf
858257.pdf	0.00%	0.86%	20.26%	40.00%	5.00%	100.00%	6.25%	1.95%	17.99%	1.13%	1.27%	.pdf
905600.pdf	0.00%	0.19%	2.03%	10.00%	1.25%	99.03%	1.44%	0.56%	1.65%	0.05%	0.35%	.pdf
270700.png	14.29%	1.38%	27.53%	70.00%	5.63%	14.33%	100.00%	2.64%	23.85%	1.17%	1.33%	.png
365624.png	14.29%	0.93%	18.42%	50.00%	5.00%	8.70%	100.00%	1.72%	15.36%	0.74%	0.75%	.png
705948.png	14.29%	0.95%	22.79%	60.00%	3.13%	10.79%	100.00%	2.02%	20.53%	0.90%	1.16%	.png
720718.png	14.29%	0.97%	20.98%	40.00%	4.38%	10.47%	100.00%	1.63%	18.23%	0.71%	0.75%	.png
821535.png	0.00%	0.45%	6.10%	20.00%	4.38%	3.70%	100.00%	0.81%	5.15%	0.41%	0.23%	.png
862652.png	0.00%	1.01%	21.43%	60.00%	3.13%	9.34%	100.00%	2.21%	19.14%	0.97%	1.10%	.png
920125.png	14.29%	1.34%	29.30%	70.00%	5.63%	13.04%	100.00%	2.73%	25.83%	1.20%	1.39%	.png
920697.png	14.29%	1.44%	29.38%	70.00%	5.63%	12.88%	100.00%	2.93%	25.91%	1.18%	1.33%	.png
920706.png	14.29%	1.42%	29.76%	70.00%	6.88%	13.20%	100.00%	2.89%	26.04%	1.22%	1.39%	.png
942196.png	0.00%	0.13%	1.58%	0.00%	0.63%	2.09%	100.00%	0.50%	1.16%	0.15%	0.00%	.png
130790.ppt	85.71%	36.22%	20.04%	60.00%	100.00%	23.67%	100.00%	97.78%	16.21%	3.61%	3.65%	.ppt
250560.ppt	85.71%	34.11%	4.37%	50.00%	9.38%	20.61%	5.77%	99.26%	3.74%	4.80%	2.03%	.ppt
307260.ppt	100.00%	37.06%	33.26%	80.00%	100.00%	34.46%	100.00%	97.63%	28.70%	6.51%	5.15%	.ppt
388207.ppt	85.71%	35.17%	6.21%	30.00%	6.25%	10.95%	100.00%	98.18%	5.23%	3.54%	2.78%	.ppt
547335.ppt	100.00%	24.90%	31.22%	80.00%	100.00%	28.02%	16.35%	91.43%	27.35%	4.76%	3.70%	.ppt
593251.ppt	100.00%	36.44%	30.58%	60.00%	10.00%	28.82%	100.00%	98.72%	25.62%	4.53%	4.40%	.ppt
623300.ppt	100.00%	37.90%	31.90%	70.00%	100.00%	25.60%	100.00%	98.56%	27.32%	5.51%	4.75%	.ppt
717834.ppt	85.71%	30.88%	7.19%	20.00%	1.88%	6.76%	2.88%	92.66%	6.41%	2.19%	0.69%	.ppt
948335.ppt	100.00%	37.04%	32.92%	80.00%	100.00%	30.92%	100.00%	98.62%	27.88%	5.53%	3.82%	.ppt
388283.xls	85.71%	15.93%	2.15%	20.00%	4.38%	22.06%	2.40%	14.16%	1.37%	98.50%	2.14%	.xls
458821.xls	100.00%	26.89%	2.07%	50.00%	5.00%	12.88%	3.85%	16.18%	1.76%	98.47%	2.31%	.xls

Filename	.bmp	.doc	.docx	.gif	.jpg	.pdf	.png	.ppt	.pptx	.xls	.xlsx	filetype
505158.xls	85.71%	20.91%	2.18%	20.00%	1.88%	18.84%	4.33%	13.46%	0.92%	97.56%	2.26%	.xls
817954.xls	85.71%	16.41%	2.07%	20.00%	3.13%	18.04%	3.85%	12.12%	1.83%	100.00%	1.68%	.xls
901341.xls	85.71%	23.50%	0.64%	0.00%	1.25%	4.19%	3.37%	8.34%	0.44%	99.95%	0.93%	.xls
216689.docx	0.00%	0.97%	98.31%	50.00%	100.00%	8.05%	5.29%	1.87%	14.47%	0.78%	3.01%	docx
223624.docx	71.43%	3.77%	100.00%	70.00%	100.00%	19.00%	100.00%	6.03%	27.65%	2.43%	4.05%	docx
299759.docx	14.29%	0.88%	97.66%	40.00%	4.38%	5.96%	5.77%	1.70%	14.13%	0.61%	2.95%	docx
313486.docx	42.86%	4.14%	100.00%	100.00%	100.00%	20.77%	100.00%	9.12%	29.18%	3.68%	4.05%	docx
372437.docx	14.29%	1.57%	99.70%	70.00%	100.00%	14.81%	100.00%	3.66%	26.99%	1.63%	1.97%	docx
598299.docx	0.00%	1.14%	99.70%	60.00%	6.25%	10.79%	100.00%	2.12%	20.58%	0.93%	1.56%	docx
605377.docx	0.00%	0.13%	100.00%	10.00%	0.00%	1.29%	0.00%	0.48%	2.74%	0.11%	1.79%	docx
615326.docx	14.29%	0.26%	100.00%	10.00%	0.63%	3.86%	4.33%	0.97%	5.87%	0.26%	2.78%	docx
182614.pptx	14.29%	2.16%	31.49%	70.00%	100.00%	17.07%	100.00%	5.63%	97.15%	2.17%	1.74%	pptx
263566.pptx	14.29%	4.03%	32.17%	70.00%	100.00%	19.16%	100.00%	7.87%	75.88%	2.70%	1.91%	pptx
318133.pptx	71.43%	5.71%	33.79%	100.00%	100.00%	22.06%	100.00%	9.13%	89.55%	3.97%	3.70%	pptx
596250.pptx	100.00%	5.11%	34.54%	100.00%	16.25%	21.42%	100.00%	9.62%	99.50%	4.10%	3.65%	pptx
606936.pptx	14.29%	2.26%	32.05%	70.00%	100.00%	15.78%	100.00%	5.00%	99.72%	2.26%	1.45%	pptx
609019.pptx	14.29%	4.33%	31.98%	70.00%	100.00%	16.59%	100.00%	7.49%	74.04%	2.71%	3.70%	pptx
729601.pptx	100.00%	6.88%	32.62%	70.00%	100.00%	26.89%	100.00%	8.35%	70.12%	3.38%	3.24%	pptx
978134.pptx	14.29%	1.60%	27.31%	60.00%	100.00%	11.76%	100.00%	3.34%	94.41%	1.37%	2.84%	pptx
996017.pptx	100.00%	4.03%	33.67%	100.00%	100.00%	23.51%	100.00%	8.04%	78.96%	3.24%	3.07%	pptx
212946.xlsx	14.29%	1.57%	32.96%	70.00%	5.63%	13.85%	9.62%	2.98%	27.70%	1.29%	100.00%	xlsx
299765.xlsx	0.00%	0.43%	8.47%	10.00%	1.88%	3.86%	1.92%	0.66%	6.12%	0.46%	100.00%	xlsx
411102.xlsx	0.00%	1.14%	24.78%	70.00%	100.00%	8.37%	7.21%	2.34%	18.38%	1.33%	100.00%	xlsx
500968.xlsx	100.00%	27.02%	2.52%	30.00%	5.63%	24.48%	3.85%	17.17%	1.67%	96.11%	0.81%	xlsx

Filename	.bmp	.doc	.docx	.gif	.jpg	.pdf	.png	.ppt	.pptx	.xls	.xlsx	filetype
528206.xlsx	0.00%	0.39%	11.53%	20.00%	0.63%	1.77%	1.92%	0.57%	4.48%	0.31%	100.00%	xlsx
848990.xlsx	0.00%	0.47%	13.56%	30.00%	1.25%	3.22%	1.44%	0.96%	7.28%	0.38%	100.00%	xlsx