A Visualization Jump Lists tool for Digital Forensics of Windows

Shiuh-Ku Weng[1*] and Jung-Yi Tu[2]
[1] Department of Computer Science and Information Engineering, Chung-Cheng Institute of Technology, National Defense University
Taoyuan City, 33551 - Taiwan (R.O.C)
[e-mail: skw@ndu.edu.tw]
[2] National Chung-Shan Institute of Science & Technology
Taoyuan City, 33551 - Taiwan (R.O.C)
[e-mail: max214013@gmail.com]
*Corresponding author: Shiuh-Ku Weng

Received July 18, 2019; revised Septebmer 20, 2019; accepted October 13, 2019; published January 31, 2020

Abstract

In this paper, a visualization digital forensics tool, called JumpList Analyzer, is implemented. The tool can analyze the complicated Jump Lists files, and then the results are demonstrated by visualization. To compare the proposed tool with the other Jump Lists tools, the proposed tool is the only one can display the analyzed results by visualization. The visualization will help the investigators more easily to find the evidence than the other tools showing the analyzed results by texts only. In the experiment, the proposed JumpList Analyzer is demonstrated its convenience at identifying artifacts for doing digital forensics in a financial fraud case. In addition, the proposed tool can also be used to reveal the computer user's behavior or background.

Keywords: Visualization, Digital Forensics, Jump Lists

http://doi.org/10.3837/tiis.2020.01.013 **ISSN : 1976-7277**

1. Introduction

Recently, Jump Lists has been one of the most important forensic artifacts. The function of Jump Lists is firstly introduced by the release of Windows 7 and keeps the function in new Windows systems [1], but the format of Jump Lists of Windows 7/8 is different from that of Window 10. The same with many forensic artifacts, the purpose of Jump Lists is to provide users with increasing usability and convenience. Since that Jump Lists is built by software applications or Windows operating system and let the users be able to "jump" to the recently accessed files and folders, it is called "Jump Lists". The way of Jump Lists maintaining the records of recently accessed files and folders is to group the records as files according to every application.

Before Jump Lists introduced, if an investigator of digital forensics would like to track a suspect's history in using applications, the only way is to access the list of the Most Recently Used (MRU) and the Most Frequently Used (MFU) in Windows registry [2]. However, Jump Lists provides more records include MRU and MFU. The records of Jump Lists are related to the MRU (Most Recently Used) and MFU (Most Frequently Used) items, file name with path, the timestamps of MAC (Modified, Accessed, and Created), disk volume name, and the history of uploading and downloading files by web browsers. By the analysis of Jump Lists, the investigators can reveal the evidences of digital forensics. For investigators, they can take advantages of this service and gather critical insight into the user's computer habits, knowledge and activities. Even the files and applications that a user ever used are deleted; the artificial records are kept in Jump Lists [3]. However, the commercial digital forensics tools, for example the famous forensics tools, FTK and Encase, have not incorporated the records of Jump Lists into their functions [3]. In the experiment of Antonovich [3], FTK, Encase and a Jump Lists parser (JumpLister [4]) cooperate to look at Jump Lists data. Owing to those characteristics which Jump Lists helps the investigation of digital forensics significantly, Jump Lists has got many discussions about how to access records of Jump Lists since Jump Lists was introduced [1] and many Jump Lists analysis tools have been developed [1][3]-[10]. Furthermore, Smith [11] identified the fraudulent documents by Windows 7 Jump Lists.

Although there are many Jump Lists analysis tools that have been developed, they analyze Jump Lists either in text based representation or only in GUI mode. All of them have no visualization demonstration. For investigators, no visualization, it is not convenient to find the clues even they are dealing with a small amount of data. Therefore, in this paper, a new Jump Lists tool with visualization is developed, and then the proposed tool is compared with all of the Jump Lists analysis tools from the Internet. We call the proposed Jump Lists tool, JumpLists Analyzer.

The advantages of JumpLists Analyzer are:

(1) It is implemented by Python language. Python is an open-source language and is a portable across operating system.
(2) It is the first visualization Jump Lists analysis tool.
(3) It demonstrates the user's activities by time axis (timeline) that is convenient to the investigators.

In the following section, the structure of Jump Lists is introduced briefly and all of the Jump Lists analysis tools from the Internet are reviewed. Then, the proposed system is introduced in section 3. The results to compare the proposed tool with the other tools are demonstrated in the section 4. The section 5 is the conclusion.

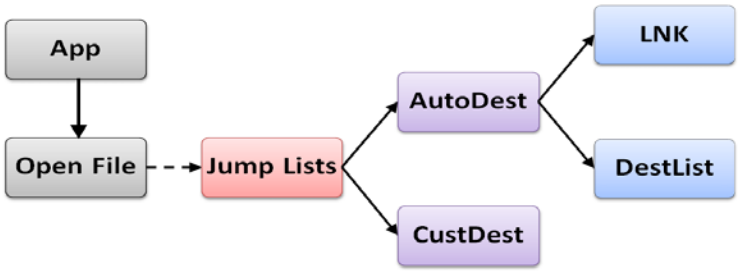


Fig. 1. The file flow chart of Jump Lists

Table 1. The file structure of Jump Lists

	Jump Lists		
File name	AutoDest		CustDest
Data streams	LNK	DestList	LNK

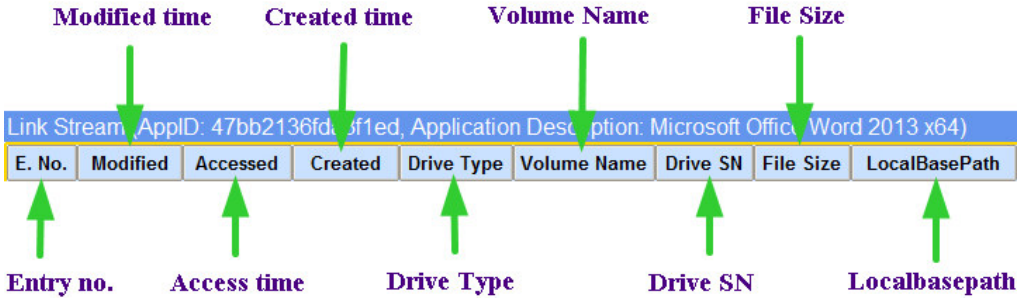


Fig. 2. The records in AutoDest (LNK) file

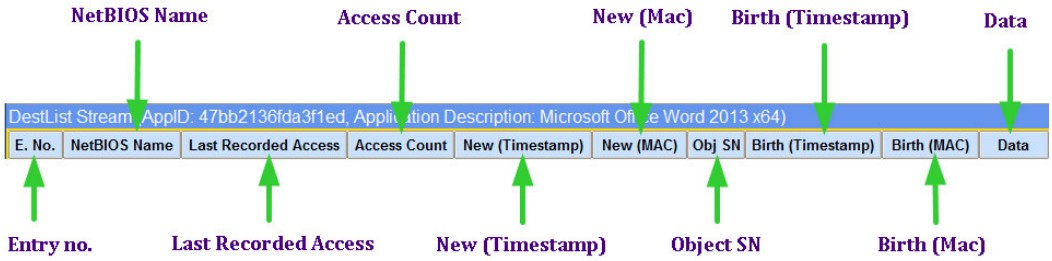


Fig. 3. The records in AutoDest (DestList) file

2. The structure of Jump Lists and Surveying Jump Lists analysis tools

2.1 The Structure of Jump Lists

The records of Jump Lists are stored in the AutoDest and CustDest files, respectively. The AutoDest consists of LNK (Shell Link) steam and DestList data stream. The records kept in CustDest are the same with LNK data streams [12]. About the DestList data stream, Microsoft has not revealed the detail information. According to the research of Singh et al. [1], by the experiment, the DestList data stream of Windows 10 is different from that of Windows 7 and 8. The records in the CustDest are the same with those of LNK data stream of AutoDest. About the expression of Jump Lists records (artifact), in the literature [1][2][12], they demonstrate Jump Lists by hexadecimal values. The hexadecimal values look complex and are not a good way for readers to understand Jump Lists. In this paper, for illustrating the file structure of Jump Lists more clearly, the file flow chart of Jump Lists is shown in **Fig. 1**. **Table 1** lists the file structure of Jump Lists. **Fig. 2** and **3** demonstrate the records stored in the two kinds of data streams, respectively. The detail explanation is in **Table 2** and **3**. The AutoDest and CustDest files are stored in

%APPDATA%\Microsoft\Windows\Recent\AutomaticDestinations

and

%APPDATA%\Microsoft\Windows\Recent\CustomDestinations, respectively.

Table 2. The meaning of the records of AutoDest (LNK) file

Record Name	Meaning
AppID	The identity of Application
Entry No.	Entry Number
Modified Timestamp	The time of file content got changed
Accessed Timestamp	The time of a file accessed, opened, edited and moved
Created Timestamp	The time of a file created. It won't be updated when the file opened, closed, saved and modified.
Drive Type	Disk type name
Volume Name	Disk volume name
Drive SN	Physical disk serial number
File Size	File size
Localbasepath	File name and path

Table 3. The meaning of records of AutoDest (DestList) file

Record Name	Meaning
AppID	The identity of Application
Entry No.	Entry Number
NetBIOS Name	Computer name in a local area network (LAN)
Last Recorded Access Time	The updated time by the three timestamps (Created, Accessed and Modified) updated
Access Count	The file accessed frequency
New (Timestamp)	The file created time and the time will be updated after the file moved
New (MAC)	The MAC address of computer
Object SN	The serial number of Main File Table (MFT)
Birth (Timestamp)	The file created time
Birth (MAC)	The MAC address of the computer a file created
Data	File name or path name

AutoDest file is Microsoft CFB (Compound File Binary) format and is also called OLE file. AutoDest file which contains SHLLINK stream and DestList stream is created by system program of Windows [1]. Its filename consists of AppID and the filename extension with "automaticDestinations-ms". Most of Jump Lists records are stored in this file. As for CustDest file, its filename is AppID with "customDestinations-ms" as its filename extension and is built by applications with calling ICustomDestinationList API [13] [18]. The contents of CustDest file is maintained by applications. There are not many applications creating CustDest file. According to the paper of Singh et al.[1], only several applications, for example, Web browsers and Windows Media player, relate to CustDest file.

2.2 Surveying Jump Lists Analysis Tools

Many Jump Lists analysis tools have been developed since Jump Lists were introduced. From the Internet, there are total 7 kinds of tools which can be found. They are JumpLists View [6], JumpLister [4], Jump Lister Parser [7], Jump List File Extract [8], JLECmd [9], JumpList Explorer [10] and JumpListExt [1]. In the following, these analysis tools are illustrated one by one.

JumpLists View is developed by NirSoft [6]. This tool is able to display the records of Jump Lists, for example, the file names, the timestamps of the opened files, AppID and file attributes, etc. However, it cannot parse the CustDest files and it represents the results in the text based model. For investigators, this tool is not convenient to analyze the data and to track the clues.

JumpLister is from Woanware [4]. This tool can recognize some default application names by decoding AppID and can parse both AutoDest and CustDest files of Windows 7. But, it is a text based tool and its timestamp is fixed at a time zone. Besides, it cannot read Jump Lists of Windows 10.

Jump Lister Parser comes from TZWorks [7] and is commercial software. Its drawbacks are the timestamps with a fixed time zone and without GUI interface.

Jump List File Extract is not a freeware and is designed by the developer H. Ulbrich [8]. The tool has neither functions to export the records of Jump Lists to a CSV or text file or as raw data except the fee is payed to get the complete one nor the ability to recognize application names by decoding AppID.

JLECmd & JumpList Explorer are developed by Eric Zimmerman [9][10]. JLECmd is a text based representation version. JumpList Explorer is a GUI version and it can only exports the shortcuts (.lnk) of applications to a file. Both of the two tools display timestamps with a fixed time zone.

JumpListExt is implemented by B. Singh et al. [1]. This tool is designed with GUI interface. It can load all of Jump Lists files at one time. But, it can neither read Windows 7 Jump Lists nor analyze CustDest files. Although, Windows 7 is obsoleting, there are still many systems run by the OS. Besides, it can only decode seldom AppIDs into application names and there are many bugs in this tools.

In the summary, all of the above tools have more or less several drawbacks. Besides, they all have no visualization functions. Therefore, in this paper, a new Jump Lists tool, JumpLists Analyzer, is proposed to conquer the drawbacks which the current Jump Lists tools have.

3. The Proposed JumpLists Analyzer Overview

The proposed system is implemented by Python with several modules and packages including graphic user interface model (Tkinter [14]), file handling (Olefile [15], SQLite3 [16]) and Matplotlib [17], etc. The graphic user interface of JumpLists Analyzer is shown in **Fig. 4**. **Fig. 5** is demonstrating the result of opening the Microsoft Word Jump Lists file by JumpLists Analyzer.

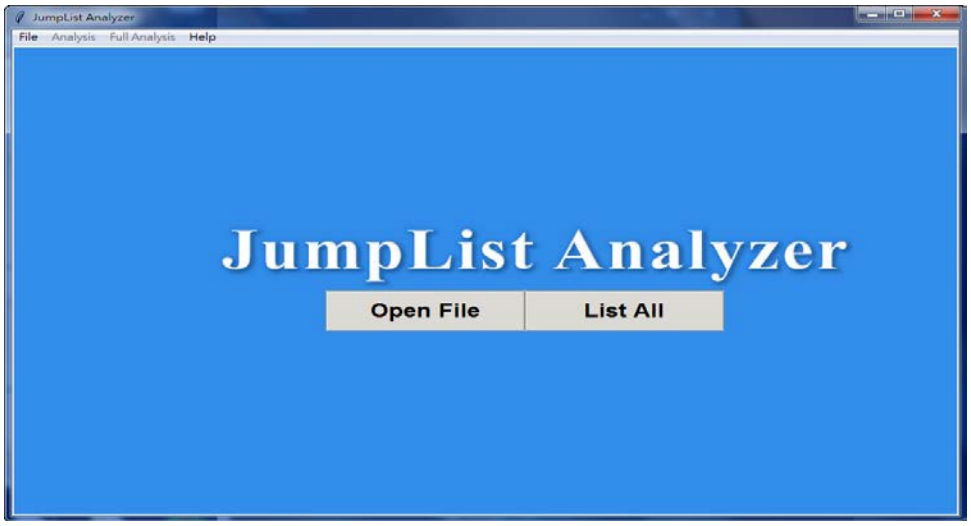


Fig. 4. The graphic user interface of JumpLists Analyzer

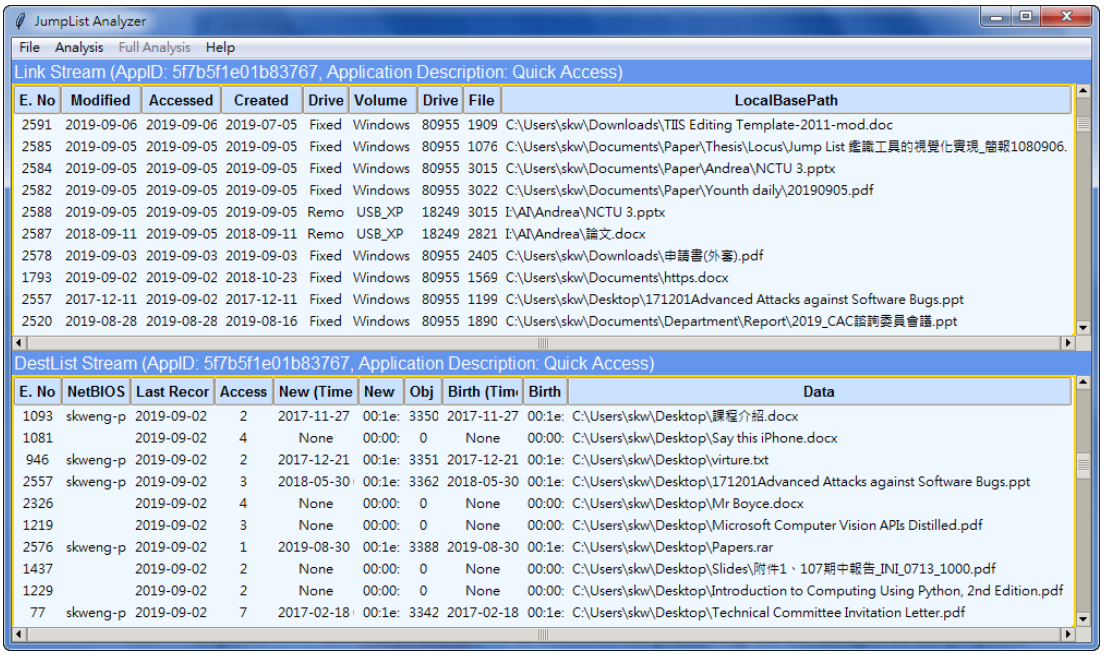


Fig. 5. The parsing result of a Jump Lists file by JumpLists Analyzer

4. Experiment

Several experiments are done in this paper. The fist experiment is to compare the proposed JumpLists Analyzer with the tools found in the Internet. Those tools are JumpListsView [6], JumpLister [4], Jump Lister Parser [7], Jump List File Extract [8], JLECmd [9], JumpList Explorer [10] and JumpListExt [1]. In addition, for presenting the advantages of the proposed JumpLists Analyzer tool, the tool is applied to identify a fraudulent document in a financial case proposed by G. S. Smith [11]. Then, we show that the proposed tool, JumpLists Analyzer, can reveal a computer user's background by analyzing the Jump Lists records obtaining from the user's computer.

Table 4. The comparisons of the Jump Lists tools

No.	Tool Name	Recognize Win 7/8	Recognize Win 10	Local Time Zone	Identify AppID	Recognize CustDest	Export Function	GUI	Visualization
1	JumpList Analyzer	✓	✓	✓	✓	✓	✓	✓	✓
2	JumpLists View v1.16 [6]	✓	✓	✓	✓		✓	✓	
3	JumpLister v1.1.0[4]	✓			✓	✓	✓	✓	
4	Jump List Parser v0.47[7]	✓	✓		✓	✓	✓		
5	Jump List File Extract v1.2[8]	✓	✓	✓		✓	✓	✓	
6	JLECmd v1.3.0.0[9]	✓	✓		✓	✓	✓		
7	JumpList Explorer v1.3.1.0[10]	✓	✓		✓	✓		✓	
8	JumpListExt v1.0[1]		✓				✓	✓	

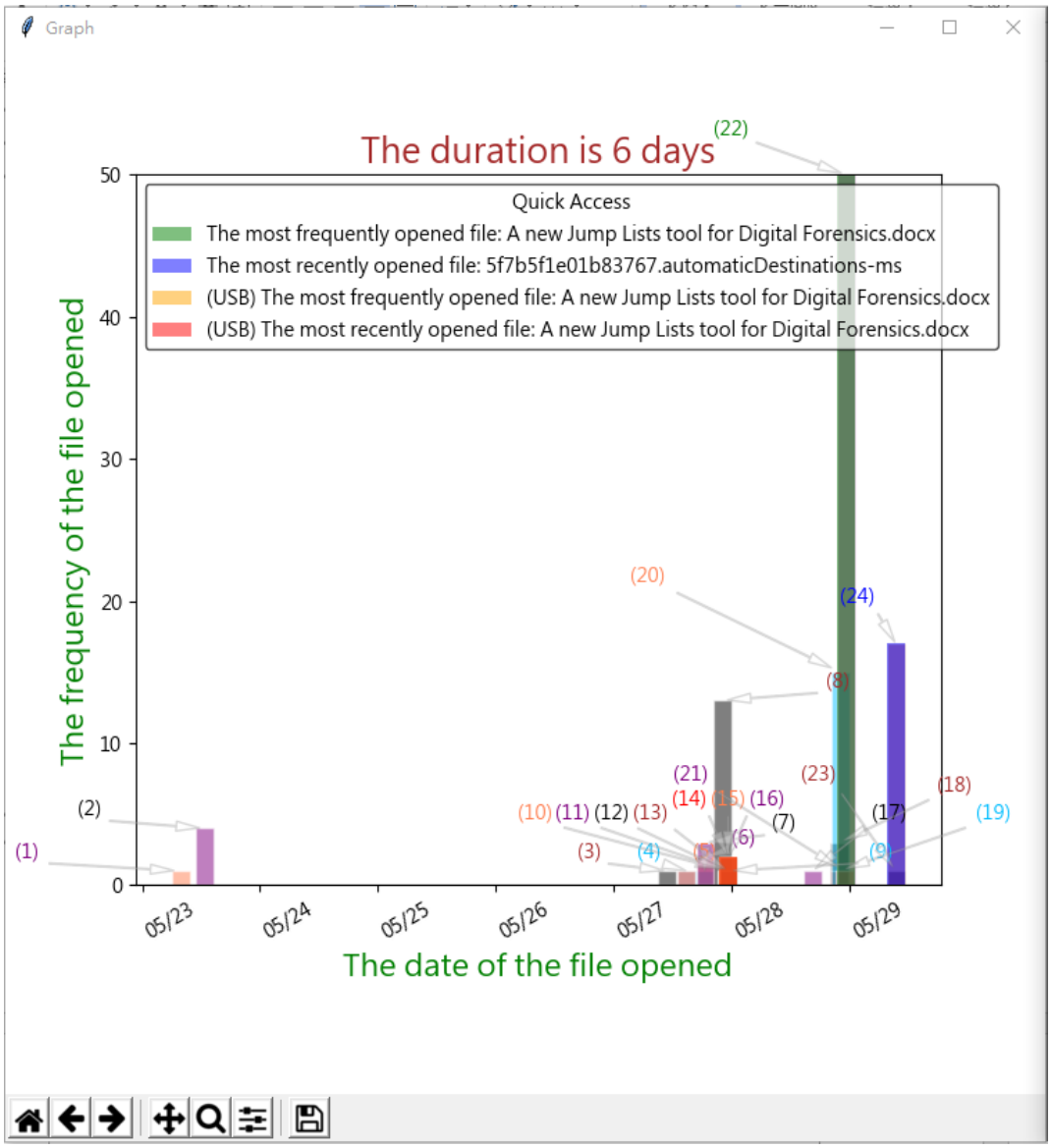
***Notice* : ✓ : Yes, Blank : No**

4.1 Comparisons of the Jump Lists Tools

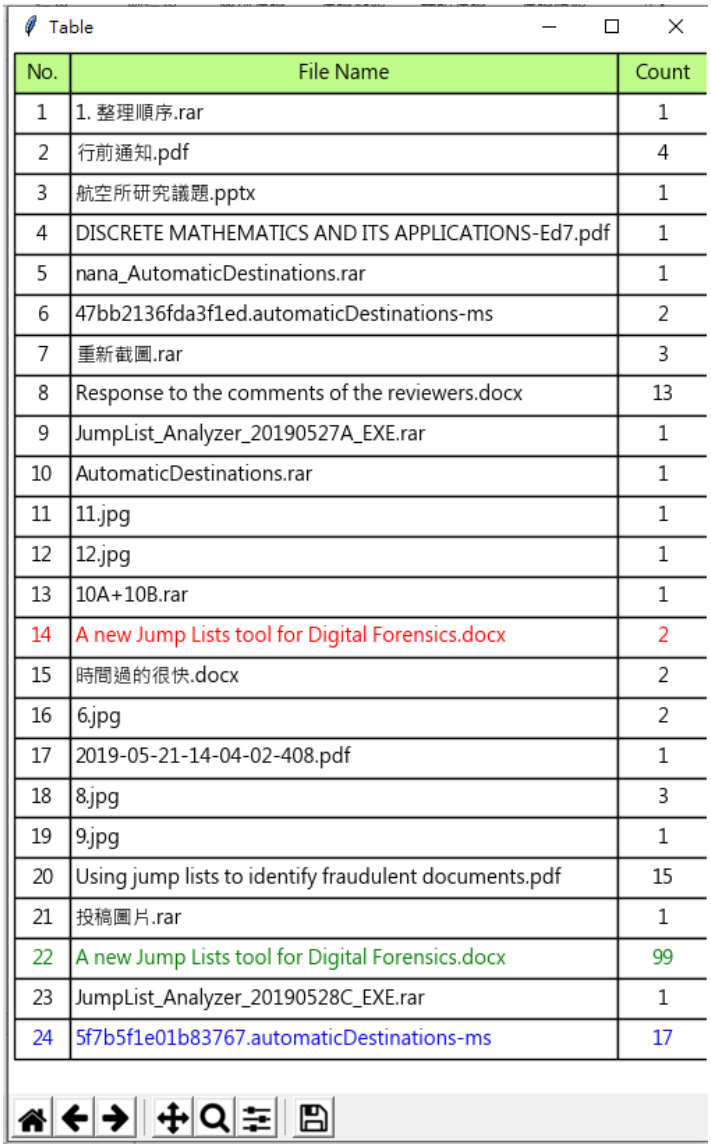
To evaluate a digital forensic tool, there are several aspects to be considered. They are compatibility, friendliness, functionality, etc. For example, in the compatibility, Jump Lists tools should be able to recognize the Jump Lists file produced by all of the versions. Jump Lists firstly are introduced with the release of Windows 7. As for the friendliness, it will consider convenience, GUI interface, visualization, etc. About the functionality, for Jump Lists, if a tool can parse CustDest file or not and the function to export Jump Lists records to a file are considered. Therefore, eight functions are chosen to be the criteria for doing comparisons. They will be "recognize the Jump Lists of Windows 7/8 version", "recognize the

Jump Lists of Windows 10 version", "GUI interface", "recognize CustDest file", "export recorded data to a file", "displaying local time zone", "identify AppID" and "visualization".

To make comparisons, the different tools one by one are used to analyze the same Jump Lists file. The tools for the experiment will be JumpList Analyzer, JumpLists View, JumpLister, Jump Lister Parser, Jump List File Extract, JLECmd, JumpList Explorer and JumpListExt in sequence.



(a)



No.	File Name	Count
1	1. 整理順序.rar	1
2	行前通知.pdf	4
3	航空所研究議題.pptx	1
4	DISCRETE MATHEMATICS AND ITS APPLICATIONS-Ed7.pdf	1
5	nana_AutomaticDestinations.rar	1
6	47bb2136fda3f1ed.automaticDestinations-ms	2
7	重新截圖.rar	3
8	Response to the comments of the reviewers.docx	13
9	JumpList_Analyzer_20190527A_EXE.rar	1
10	AutomaticDestinations.rar	1
11	11.jpg	1
12	12.jpg	1
13	10A+10B.rar	1
14	A new Jump Lists tool for Digital Forensics.docx	2
15	時間過的很快.docx	2
16	6.jpg	2
17	2019-05-21-14-04-02-408.pdf	1
18	8.jpg	3
19	9.jpg	1
20	Using jump lists to identify fraudulent documents.pdf	15
21	投稿圖片.rar	1
22	A new Jump Lists tool for Digital Forensics.docx	99
23	JumpList_Analyzer_20190528C_EXE.rar	1
24	5f7b5f1e01b83767.automaticDestinations-ms	17

(b)

Fig. 6. An analyzed result by JumpList Analyzer with visualization. (a) Histogram chart. (b) Look-up table of file names

Firstly, JumpLists Analyzer can parse Jump Lists files from Windows 7/8 and 10, respectively. In addition, JumpLists Analyzer can display the correct time zone, map AppID to corresponding application name, recognize Custdest file and export records to a file. **Fig. 6** shows an analyzed result by the proposed JumpList Analyzer with visualization. The frequency of every opened file is illustrated in the histogram for the duration 6 days. In the top of **Fig. 6**, the four kinds of file names are listed. They are the two most frequently opened files and the two most recently opened files from the hard disk and USB, respectively. That is to say, it can also reveal the most frequently and recently opened files which accessed from USB disk.

A date in the **X**-axis is the most recent date to open a file. A number pointing to a bar in **Fig. 6 (a)** maps to a file. The investigators can look up the corresponding file name and file opened count (frequency) by the number in the table of **Fig. 6 (b)**. Please notice that for balance the displaying graph, the frequency bar over 50 is truncated.

To compare the proposed tool with the other tools, the functions for individual tool are summarized in **Table 4**. Most of tools own the functions of recognizing different Windows versions, identifying AppID, supporting CustDest file, exporting data and designing by GUI. The less tools provide the function of time zone transformation. That is to say, only several tools can display the records of timestamps according to investigator's local time zone which is not fixed at a tool developer's time zone. By **Table 4**, the tool with the least functions is JumpListExt[1]. It only owns three kinds of functions. In the summary, the proposed tool has all of the functions which are from the perspectives of compatibility, friendliness and functionality.

4.2 Case study

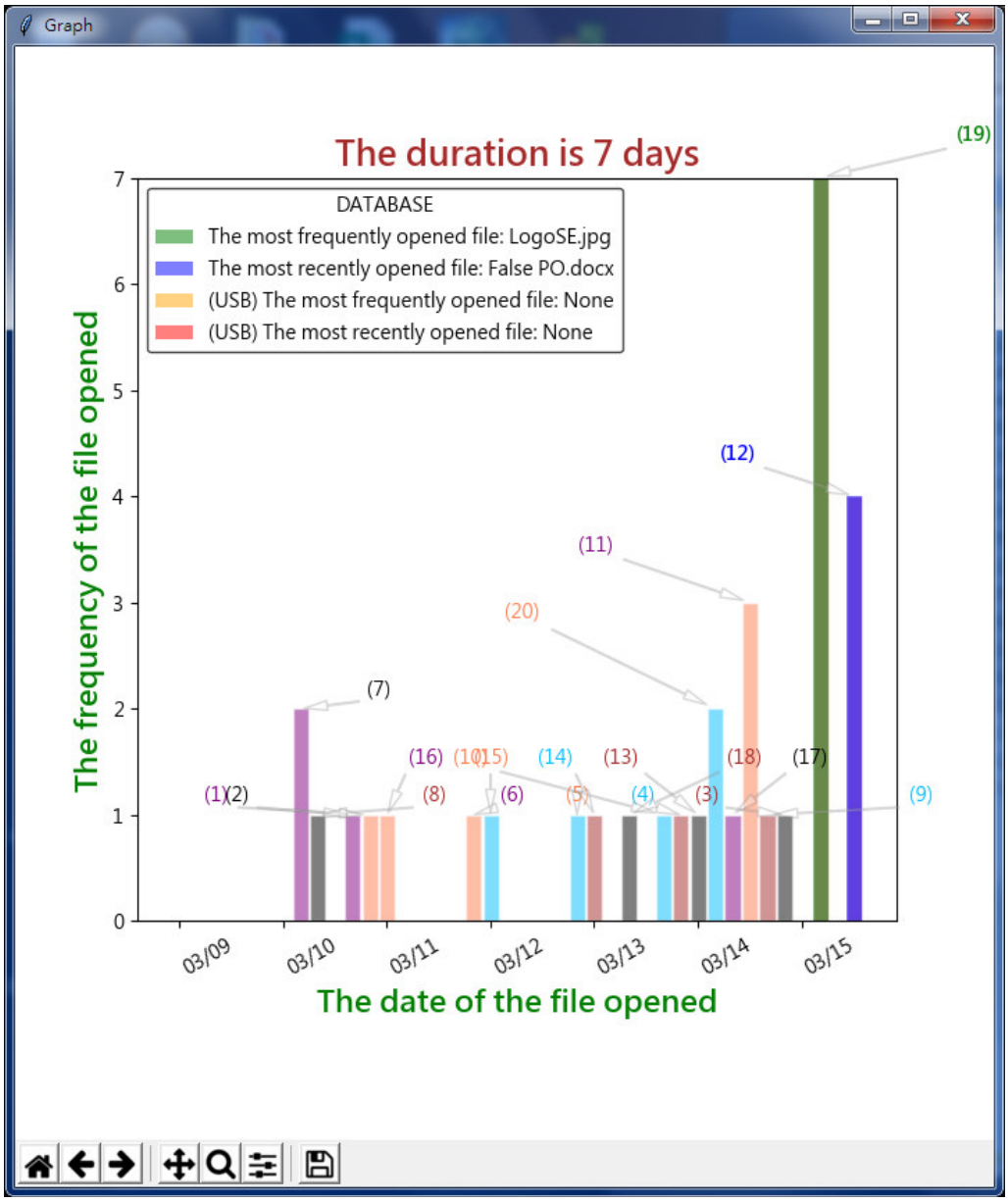
In this subsection, the proposed JumpList Analyzer is used to show its convenience in identifying a financial fraud case. The financial fraud case study is proposed in the paper [11]. In this case, a fraudster creates a purchase order by MS Word to replace the real one and a Jump Lists tool called Jumplister [4] is applied to reveal the complete trail of the fraudster in creating fraudulent documents while using computer. Firstly, the fraudster downloads a logo (file name : LogoSE.jpg) from Southeastern Oklahoma University website and saves it to the Picture folder on his PC. Next, the fraudster creates a purchase order template of MS Word (file name : PurchaseOrder.docx). Then, the fraudster completes the fabricated purchase order (file name : False PO.docx) and moves the fraudulent document to a folder where it would be hard to be found. After sending out his fraudulent order, the fraudster deletes the fraudulent order file from his computer. In the paper [11], the author applies a Jump Lists tool, called Jumplister [4], to find the evidence that the fraudster fabricates the purchase order.

According to the description of the investigation steps of digital forensics in the paper [11], the steps how to investigate the fraudulent case by Jumplister [4] are listed as follows.

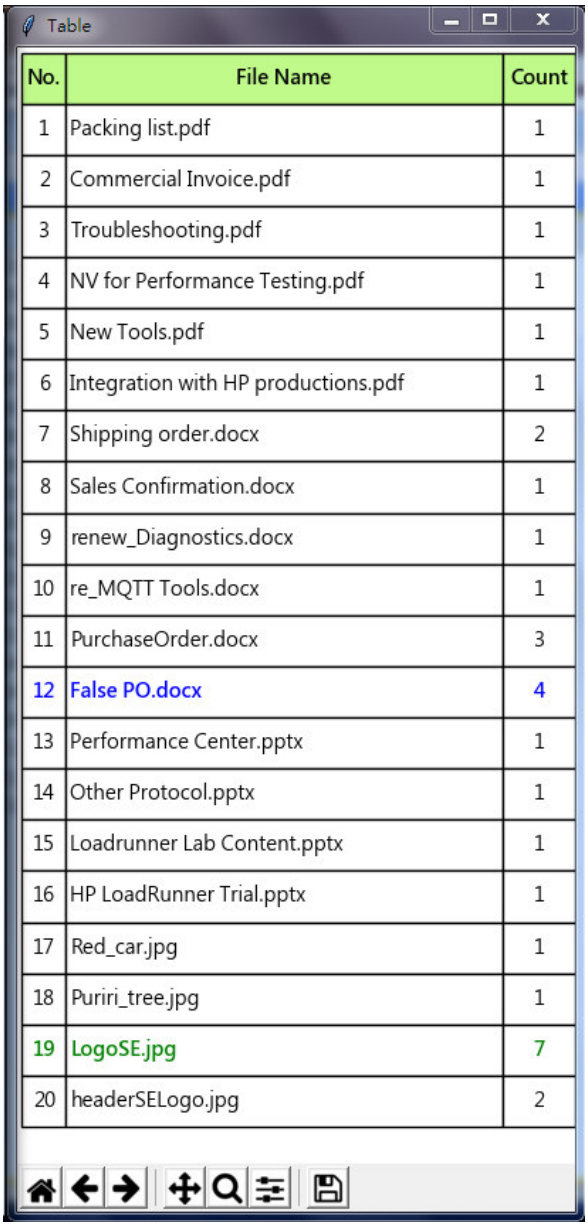
1. The first step is to find the date that the fraudulent order is created and the software application is used to create the file.
2. The second step is to search the artifacts (records) for activities relating to the creation of the fraudulent purchase order by the date.

In the demonstration of the paper [11], Jumplister [4] is not easy to locate the correct artifacts for finding the activities of building the fraudulent document. Besides, according to our experiment, Jumplister cannot recognize Windows 10 jump list file and cannot display the correct time zone where an investigator is located and is using the tool. No.3 of **Table 4** lists the functions of JumpLister. The proposed JumpList Analyzer is more convenient to do the investigation into the case in comparison with doing it by JumpLister since the investigators can find the suspect files in a duration by JumpList Analyzer using the visualization. However, by JumpLister, the investigators have to check the records line by line to find the suspect files in paper [11]. For example, the case study of the paper [11] is simulated and the analyzed result is shown in **Fig. 7**. The X-axis of **Fig. 7 (a)** represents the activity timeline of a duration. From the histogram of the figure, the investigators can find the suspect files from the specific date or from the most frequently and the most recently opened files in the duration, then look into the details of Jump Lists records in **Fig. 8**. In **Fig.7 (a)**, the file name of every histogram bar can be looked it up in the table of **Fig.7 (b)** according to the number pointing to every bar.

Then, by **Fig. 8**, the files can be located where they are saved easily. In the case study, the files of the top three highest frequencies, LogoSE.jpg, False PO.docx and PurchaseOrder.docx, are the suspect files. Please notice that **Fig. 8 (a)** and **(b)** only displays the Jump Lists records of MS Word files and those of JPEG files, respectively.



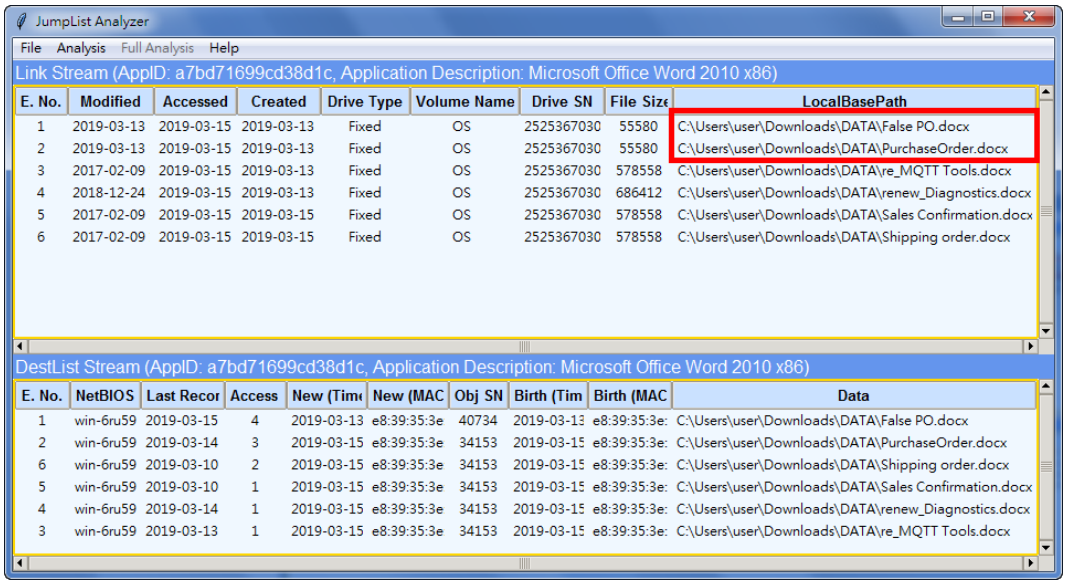
(a)



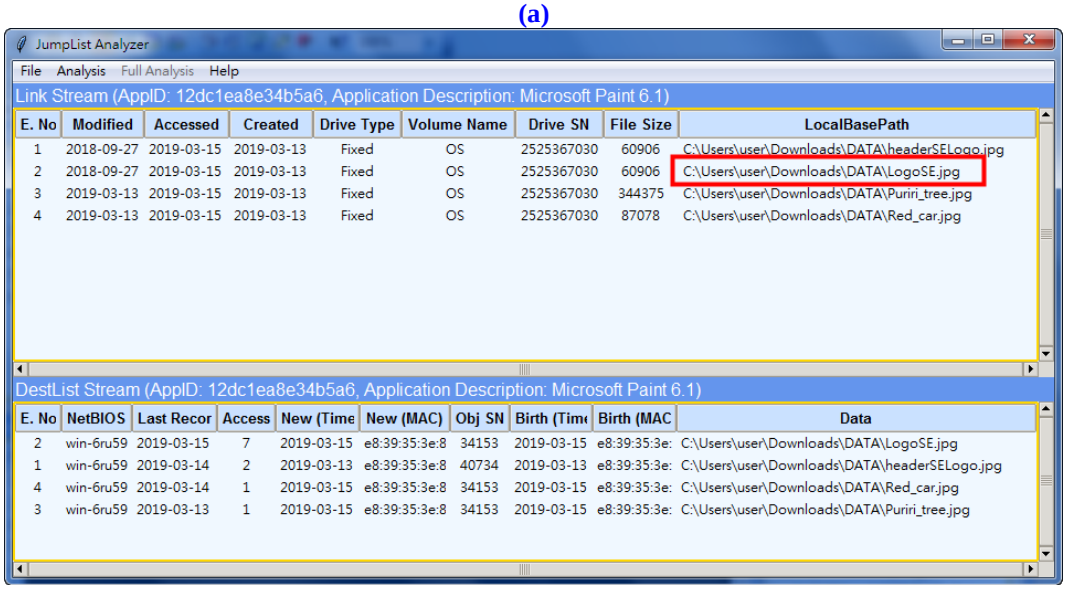
No.	File Name	Count
1	Packing list.pdf	1
2	Commercial Invoice.pdf	1
3	Troubleshooting.pdf	1
4	NV for Performance Testing.pdf	1
5	New Tools.pdf	1
6	Integration with HP productions.pdf	1
7	Shipping order.docx	2
8	Sales Confirmation.docx	1
9	renew_Diagnostics.docx	1
10	re_MQTT Tools.docx	1
11	PurchaseOrder.docx	3
12	False PO.docx	4
13	Performance Center.pptx	1
14	Other Protocol.pptx	1
15	Loadrunner Lab Content.pptx	1
16	HP LoadRunner Trial.pptx	1
17	Red_car.jpg	1
18	Puriri_tree.jpg	1
19	LogoSE.jpg	7
20	headerSELogo.jpg	2

(b)

Fig. 7. (a) Investigation of fraudulent purchase order by JumpList Analyzer. (a) Histogram chart. (b) Look-up table of file names.



(a)



(b)

Fig. 8. (a) The location of False PO.docx and PurchaseOrder.docx (b) The location of LogoSE.jpg

4.3 User background and behavior analysis

The proposed JumpList Analyzer has the other one advantage in doing user's background and behavior analysis. For example, in **Fig. 9** and **10**, they are the visualized results by JumpList Analyzer for analyzing a user's Jump Lists in a duration. **Fig. 9** displays the histogram of the opened files with numbers pointing to and the numbers corresponding to the opened file names with the opened frequency (count) respectively are listed in **Fig. 10**. The block in the top of **Fig. 9** lists the two most recently opened files and the two most frequently opened files from hard disk and USB disk, respectively. In **Fig. 9**, the two highest frequencies are the number 78 (blue one) and 72 (green one). By looking up file names in **Fig. 10**, the number 78 and 72 are a MS Office Word thesis file with opened count 53 and a WPR file with opened count 55,

respectively. Please notice that the two bars almost overlap together. The WPR file type is primarily associated with Wing IDE. Wing IDE is a software development environment for the Python programming language. An investigator can realize that the user was running Python programs and doing his/her thesis for research during the period of time. Therefore, an investigator can guess that he/she was a graduate student.

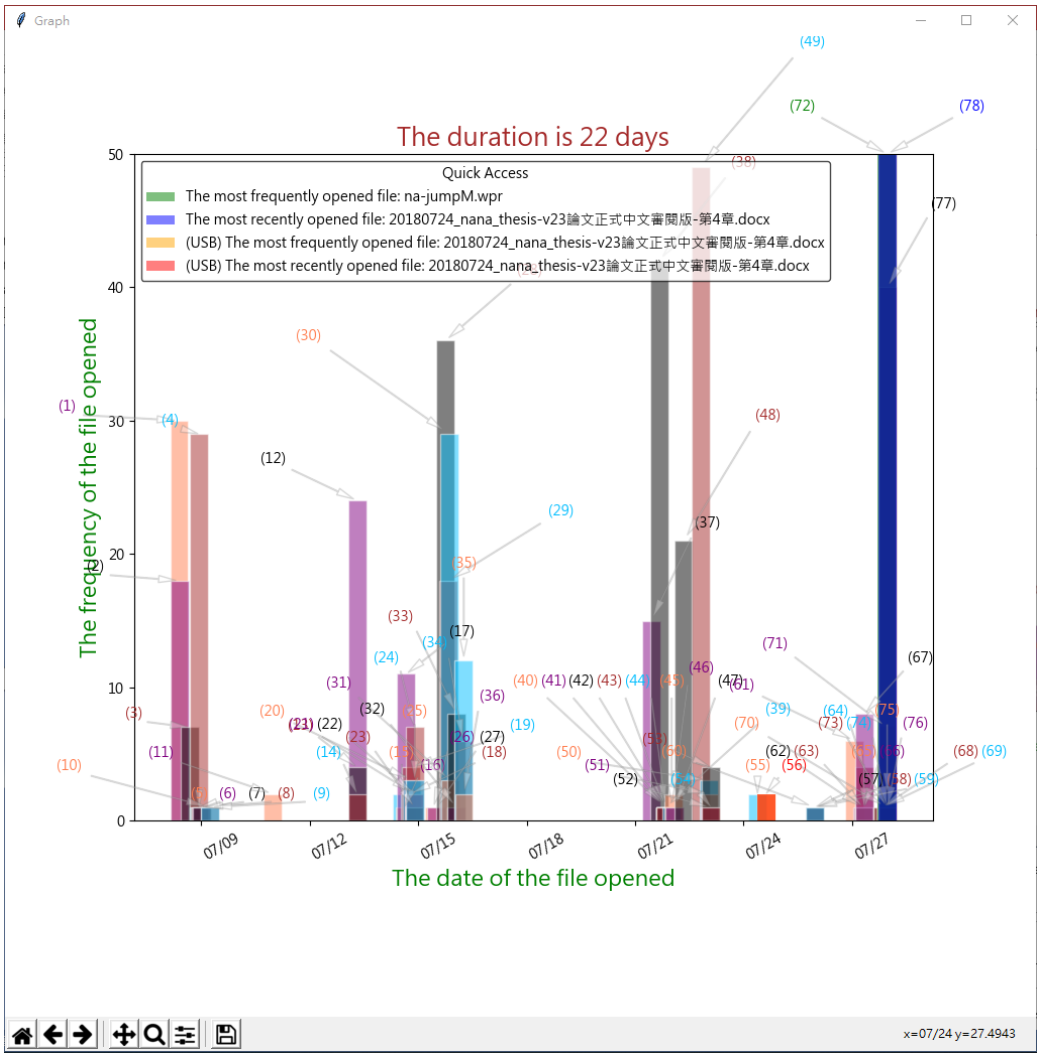


Fig. 9. Histogram of user behavior analysis by JumList Analyzer in a duration

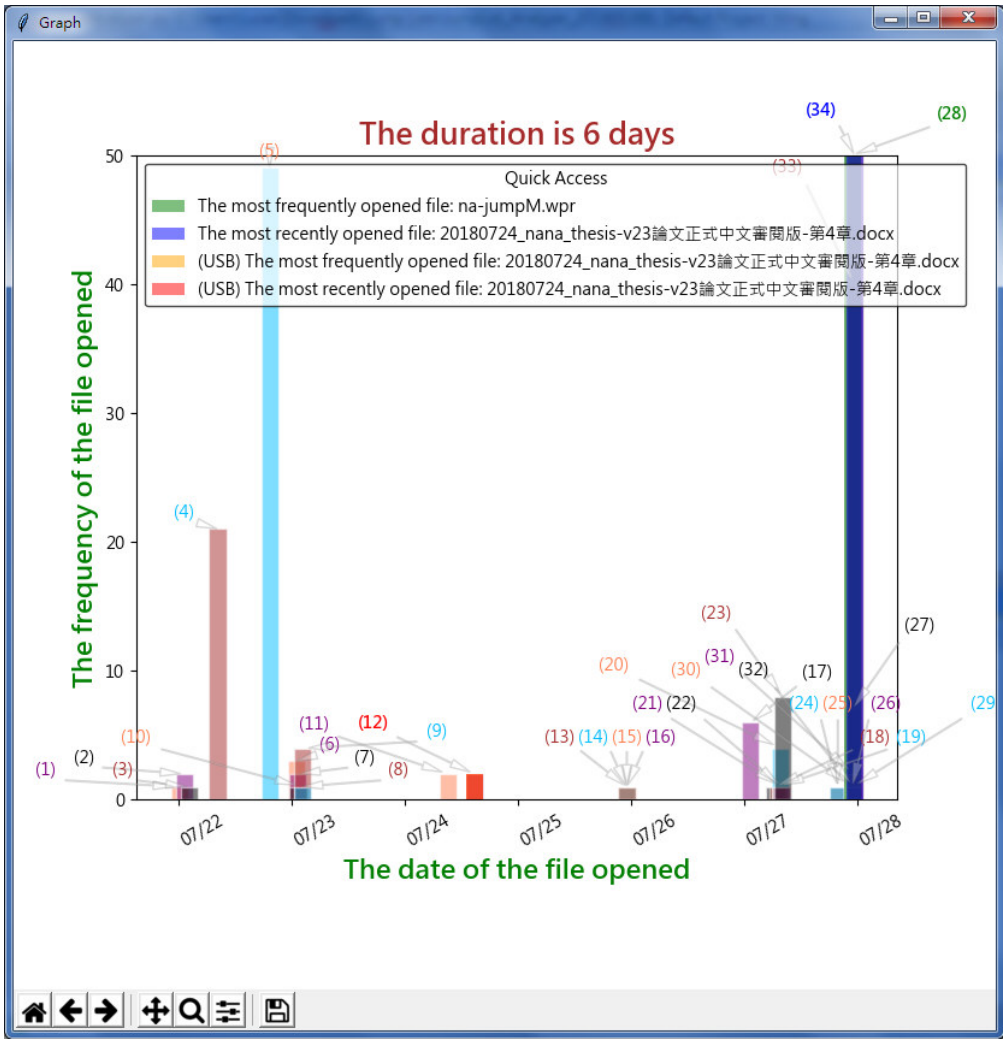
In addition, in this case, about the most frequently and recently opened file which is accessed from USB, the file is the number 56 (the red color one) in **Fig. 10** and its accessed date is July 24 by looking up the number 56 in **Fig. 9**. In this case, the file of the number 56 in **Fig. 10** is also the most recently opened file - the number 78 (the blue color one). That is to say, the most recent dates that the user accessed and opened the file from USB disk (opened twice) and from the hard disk (opened 54 times) are on July 24th and around on July 28th, respectively. Therefore, the record which a user accesses a file from USB disk can also be revealed by the proposed JumpList Analyzer. Although **Fig. 9** is very crowded and the number 56 seems to be hard to find in **Fig. 9**, the duration can be set up in a shorter one to zoom in the

visualized result. The zoomed in result with setting up the duration around the date of the number 56 is demonstrated in **Fig. 11**. Please notice that the file number 56, 72, and 78 of **Fig. 9** are the number 12, 28 and 34 of **Fig. 11**, respectively.

Table

No.	File Name	Count	No.	File Name	Count
1	20180624_nana_thesis-v11論文正式中文版.docx	30	40	S_8292465822071.jpg	1
2	20180620_nana_thesis-v9論文正式版-中文初稿-weng	18	41	S_8292465871124.jpg	1
3	楞伽-南懷瑾-02.docx	7	42	S_8292465916506.jpg	1
4	nana_thesis-v1論文full-schema版.txt	29	43	S_8292465963484.jpg	1
5	1070708-ann-01.jpg	1	44	S_8292466018298.jpg	1
6	snp14.pdf	1	45	kc&柯p.jpg	1
7	58_Paper.pdf	1	46	Doc1.docx	2
8	6-Table1-1.png	1	47	91615571371.pdf	1
9	81-1.gif	1	48	20180721_nana_Top5_Open Source Employee Monit	21
10	download13.pdf	1	49	20180721_nana_thesis-v21論文正式中文審閱版.docx	49
11	900828-1060609nana_交總局版.doc	2	50	1070727JumpListData-M.sqlite	3
12	20180630_nana_thesis-v13論文正式中文審閱版.docx	24	51	na-JumpListM.py	2
13	A forensic insight into Windows 10 Jump Lists_20171	4	52	na-JumpListM-v01-20180722.py	1
14	20180630_nana_thesis-v13論文正式中文審閱版.docx	2	53	na-JumpListM-v01.py	4
15	word尋找取代特殊字表.pdf	2	54	na-JumpListM-white.py	1
16	Capture0.png	1	55	20180722_nana_thesis-v22論文正式中文審閱版.docx	2
17	[MS-CFB]-151016.docx	11	56	20180724_nana_thesis-v23論文正式中文審閱版-第4章	2
18	IMG_9571.JPG	2	57	1070725a.jpg	1
19	[MS-SHLLINK]-151016.docx	4	58	1070725b.jpg	1
20	[MS-DTYP].pdf	3	59	1070725c.jpg	1
21	Image_a6050fe.jpg	2	60	1070725d.jpg	1
22	filesystem-timestamps-tick-36842.pdf	2	61	1070724nana-Lab-補充文字說明.txt	6
23	論文研究、蒐集、整理與歸納.pdf	1	62	na-JumpListM-v01-20180722.py	1
24	碩士論文封面-資工系.doc	7	63	na-JumpListM-v01-20180727.py	1
25	1011127論文格式.doc	3	64	na-JumpListM-white.py	4
26	filesystem-timestamps-tick-36842.pdf	1	65	na-JumpListM-v01-0727a.py	1
27	20180714_nana_thesis-v15論文正式中文審閱版_na原.	1	66	na-JumpListM-usb.py	1
28	20180714_nana_thesis-v15論文正式中文審閱版.docx	36	67	1070722-appname-show.docx	8
29	20180715_nana_thesis-v16論文正式中文審閱版.docx	18	68	na-JumpListM-v01-usb.py	1
30	A forensic insight into Windows 10 Jump Lists_20171	29	69	na-JumpListM-v01-usb-20180727.py	1
31	20180715_nana_thesis-v17論文正式中文審閱版.docx	3	70	na-JumpListM02.py	1
32	獻給導師.mp3	1	71	na-JumpListM-v01.py	7
33	師父的手.mp4	8	72	na-jumpM.wpr	55
34	nana_thesis-v1論文自製圖表.pptx	6	73	bitstring.py	1
35	20180715_nana_thesis-v18論文正式中文審閱版.docx	12	74	na-JumpListM02.py	1
36	1070716email-weng-screen.docx	2	75	na-JumpListM02.py	2
37	1070714-objectID-timestamp-MAC-File Times.docx	15	76	bitstring.py	1
38	20180630nana_top7_monitor_compare_table.docx	42	77	na-JumpListM.py	40
39	S_8292465780400.jpg	1	78	20180724_nana_thesis-v23論文正式中文審閱版-第4章	53

Fig. 10. Lookup table of user behavior analysis by JumList Analyzer


Graph
The duration is 6 days
Quick Access
The most frequently opened file: na-jumpM.wpr
The most recently opened file: 20180724_nana_thesis-v23論文正式中文審閱版-第4章.docx
(USB) The most frequently opened file: 20180724_nana_thesis-v23論文正式中文審閱版-第4章.docx
(USB) The most recently opened file: 20180724_nana_thesis-v23論文正式中文審閱版-第4章.docx
The frequency of the file opened
The date of the file opened
07/22
07/23
07/24
07/25
07/26
07/27
07/28

(a)

No.	File Name	Count
1	kc&柯p.jpg	1
2	Doc1.docx	2
3	91615571371.pdf	1
4	20180721_nana_Top5_Open Source Employee Monitoring Software Fo	21
5	20180721_nana_thesis-v21論文正式中文審閱版.docx	49
6	1070727JumpListData-M.sqlite	3
7	na-JumpListM.py	2
8	na-JumpListM-v01-20180722.py	1
9	na-JumpListM-v01.py	4
10	na-JumpListM-white.py	1
11	20180722_nana_thesis-v22論文正式中文審閱版.docx	2
12	20180724_nana_thesis-v23論文正式中文審閱版-第4章.docx	2
13	1070725a.jpg	1
14	1070725b.jpg	1
15	1070725c.jpg	1
16	1070725d.jpg	1
17	1070724nana-Lab-補充文字說明.txt	6
18	na-JumpListM-v01-20180722.py	1
19	na-JumpListM-v01-20180727.py	1
20	na-JumpListM-white.py	4
21	na-JumpListM-v01-0727a.py	1
22	na-JumpListM-usb.py	1
23	1070722-appname-show.docx	8
24	na-JumpListM-v01-usb.py	1
25	na-JumpListM-v01-usb-20180727.py	1
26	na-JumpListM02.py	1
27	na-JumpListM-v01.py	7
28	na-jumpM.wpr	55
29	bitstring.py	1
30	na-JumpListM02.py	1
31	na-JumpListM02.py	2
32	bitstring.py	1
33	na-JumpListM.py	40
34	20180724_nana_thesis-v23論文正式中文審閱版-第4章.docx	53

(b)

Fig. 11. The roomed in result of **Fig. 9** by a shorter duration. (a) Histogram chart. (b) Look-up table of file names.

5. Conclusion

The digital forensics are increasing significantly. A good digital forensics tool will help the investigators to narrow down the search space and make the investigation more accurate and faster. The visualization is very efficient for dealing with big data. The proposed Jump Lists tool uses the statistical charts to visualize the analyzed results. By the analyzed results, the proposed tool can reduce the investigating complexity. In the experiment, the financial fraudulent case study has shown it. In addition, according to the visualized results, the proposed tool can also reveal the user's behavior or background easily. To know the suspect's behavior or background will help the investigators in doing the investigation significantly. Furthermore, all of Jump Lists tools from the Internet are compared with the proposed tool according to the compatibility, friendliness and functionality. The comparison results show that the proposed tool has the most advantages.

References

[1] B. Singh, U. Singh, "A forensic insight into Windows 10 Jump Lists," *Digital Forensics*, vol. 17 pp. 1-13, June 2016. Article (CrossRef Link).

[2] R. Lyness, "Forensic analysis of windows 7 jump lists," *forensic focus*, 2012. Article (CrossRef Link).

[3] C. Antonovich, "Jump List Forensics", Patrick Leahy Center for Digital Investigation (LCDI), Champlain College Miller Center, Burlington, USA, April, 2014.

[4] M. Woan, JumpLister v1.1.0., May 16, 2013. Retrieved from https://github.com/woanware/JumpLister

[5] A. Ghafarian, "Investigating Forensics Values of Windows Jump Lists Data," in *Proc. of Annual ADFSL Conference on Digital Forensics, Security and Law, University of North Georgia, Dahlonega*, May, 2015.

[6] Nir Softer, JumpListsView v1.16., 2018. Retrieved from https://www.nirsoft.net/utils/jump_lists_view.html

[7] TZWorks, Jump List Parser (jmp), 2019. Retrieved from https://tzworks.net/prototype_page.php?proto_id=20

[8] H. Ulbrich, Jump List File Extract, March 15, 2011. Retrieved from https://download.cnet.com/Jumplist-File-Extract/3000-2086_4-75326742.html

[9] Eric Zimmerman, JLECmd 1.3.0.0 – Jump List parser. Retrieved from https://ericzimmerman.github.io

[10] Eric Zimmerman, JumpList Explorer 1.3.1.0 – GUI based Jump List viewer. Retrieved from https://ericzimmerman.github.io

[11] G. S. Smith, "Using jump lists to identify fraudulent documents," *Digital Investigation*, vol. 9, no. 3-4, pp. 193-199, February 2013. Article (CrossRef Link).

[12] Eric Zimmerman, Jump lists in depth: Understand the format, February 2016. Retrieved from https://binaryforay.blogspot.com/2016/02/jump-lists-in-depth-understand-format.html

[13] M. Sjögren, UWP Jump lists done right, May 31, 2018. Retrieved from https://blog.jayway.com/2018/05/31/uwp-jump-lists-done-right/

[14] tkinter — Python interface to Tcl/Tk, September 4, 2019. Retrieved from https://docs.python.org/3/library/tkinter.html

[15] P. Lagadec, Olefile - a Python module to read/write MS OLE2 files, September 9, 2018. Retrieved from https://www.decalage.info/olefile

[16] SQLite3 — DB-API 2.0 interface for SQLite databases, September 4, 2019. Retrieved from https://docs.python.org/3/library/sqlite3.html

[17] J. Hunter, D. Dale, E. Firing and M. Droettboom, Matplotlib, Augest 26, 2019. Retrieved from https://matplotlib.org/

[18] T. Larson, "Forensic examination of windows 7 jump lists," June 6, 2011. Retrieved from https://www.slideshare.net/ctin/windows-7-forensics-jump-listsrv3public

Shiuh-Ku Weng received Ph.D degree from Chung Cheng Institute of Technology (CCIT) of National Defense University (NDU) in EE in 1997. Currently, he is an associate professor in Department of Computer Science and Information Engineering, CCIT, NDU. His research interests include Digital Forensics, Cyber Security and Video Object Tracking.

JungYi Tu received MS degree from Department of Computer Science and Information Engineering of Chung Cheng Institute of Technology (CCIT) of National Defense University (NDU) in 2018. He is working in National Chung Shan Institute of Science & Technology as an Engineer now. His research interests are in Digital Forensics and Cyber Security. Her research interests are in Digital Forensics and Cyber Security.