 ISSN:2302-8149
e-ISSN:2540-9719

Design and Implementation of a Digital Forensics Tool to Enhance Windows Artifact Analysis

[1]Manar Talat Ahmad*
[1]Department of Computer Science, University of Mosul, Mosul, Iraq
*e-mail: manar_seyala@uomosul.edu.iq

(***received*****:** 20 April 2026, ***revised*****:** 12 June 2026, ***accepted*****:** 13 June 2026)

Abstract

Although the wide use of technology came with many advantages and facilities to the people daily life, it causes the cybercrime to be raised. Digital forensics is one of the most important scientific fields, aiming to investigate cybercrimes and analyze digital evidence. Among different technology's platforms, operating systems is one of the most important sources of evidence for digital forensic analysts providing a rich information that can used to get important insights. Examples of such evidence include identifying programs that have been executed on a computer, determining files that have been accessed, and identifying storage devices that were connected via USB ports. Practically, accessing and handling this raw information using manual methods is time-consuming, in addition to the lack of accuracy in results due to human errors. In this work, a GUI-based tool is presented to handle most of the evidence provided by Windows operating system that can be used in digital forensics. The research aims to fill the gap caused by the lack of a free tool that deals with these sources, as most available tools are either commercial tools that are complex to use and require expert-level experience. In contrast, available free tools have limited-capability since they are focusing only on one type of evidence. The introduced tool was designed and developed using the C# programming language and was tested on the Windows 10 operating system, where it successfully extracted the required information efficiently and smoothly.

Keywords: cybercrime investigation, digital forensics, jump list, LNK files, prefetch files, recent apps, userassist, windows artifacts analysis

1 Introduction

Cybercrime is a major concern for all computer users today, whether at the personal or institutional level. The wide use of technology increased the ratio of cybercrimes. Cybercrimes take different forms according to their platforms and environments, such as computer users and mobile device users [1][2]. Today, Windows operating system is one of the most widely used environments at personal-level, which makes it an important target by cybercriminals [3]. Windows, as an operating system, has many artifacts that are unknown and unnoticed by users. These hidden artifacts are important source of information that can be analyzed by digital investigators. These artifacts may let to many results helping to understand the past system behavior, such as identifying programs execution, files accessing, and many other activities [3]. From a practical perspective, the normal end-user is unable to read and analyze these artifacts because they are stored in encrypted format. Because of this complexity, many tools—mostly commercial—were developed to handle these artifacts and extract their contents. In addition, some other free tools focus only on a particular type of evidence provided by the Windows operating system.

To address this gap, this work aims to handle most of the information available within Windows operating systems by developing a GUI-based tool that presents the extracted information in an organized and readable manner for digital forensic analysts, thereby saving time and effort while ensuring accuracy in presentation. Providing a tool that deals with the most Windows artifacts using a single and centralized GUI is the main goal of this work.

Table 1 summarizes the key Windows forensic artifacts used in this study and their forensic significance.

 ISSN:2302-8149
e-ISSN:2540-9719

Table 1 Windows forensic artifacts utilized in this study

Artifact	Source Type	Location	Forensic Significance
UserAssist	Registry	HKCU...\Explorer\UserAssist	Records GUI-based program execution
RecentApps	Registry	HKCU...\Search\RecentApps	Identifies recently used applications
Prefetch Files	System Files	%SystemRoot%\Prefetch	Stores execution metadata and run counts
LNK Files	Shortcut Files	%APPDATA%\Microsoft\Windows\Recent	Tracks recently accessed files
Jump Lists	System Files	Automatic & Custom Destinations	Records application-based file access
USBSTOR	Registry	HKLM\SYSTEM...\USBSTOR	Stores USB storage device history
Enum\USB	Registry	HKLM\SYSTEM...\Enum\USB	Enumerates all USB devices

2 Literature Review

The topic of digital forensic has been addressed in several articles from various perspectives. For example, in [4], the analysis of Windows telemetry data was introduced to investigate and demonstrate the forensic value of RBS files. Other works, such as [5][6], adopted the prefetch files to detect evidences of program execution. A track execution of programs was also introduced in [7] to detect file deletion process. On another hand, study [8] handled file-related evidence analysis, while study [9] handled folder-related evidence analysis through a Windows Shellbag study. In addition, study [10] addressed the analyzing of USB-related Windows artifacts, while [11] provided a comprehensive review of Windows to forensic artifacts. On another level, Windows artifacts were utilized in [12][13][14] to trace and detect malwares. Artificial intelligence was also integrated to the filed in different ways trying to offer smarter methods for uncovering evidence and its connection to the crime – such as [15][16][17]. Finally, authors of [18][19] addressed the digital forensics topic within specific geographical and legal contexts.

In contrast to existing studies, the present work proposes an integrated and modular digital forensics tool that consolidates the analysis of multiple Windows forensic artifacts into a single unified graphical environment. The introduced tool aims to reduce the time and the complexity of collecting and analyzing Windows artifacts.

3 Methodology

This study focuses on developing an integrated digital forensics tool that is capable of analyzing many digital forensic evidences on Windows systems within a unified framework. More detailed are explained in the following sections.

3.1 System Architecture

The proposed tool was designed using a modular architecture that provides flexibility and scalability. Each artifact was handled by a separate module, allowing for individual processing of each piece of evidence.

The overall system architecture of the proposed tool is shown in Figure 1.

 ISSN:2302-8149
e-ISSN:2540-9719

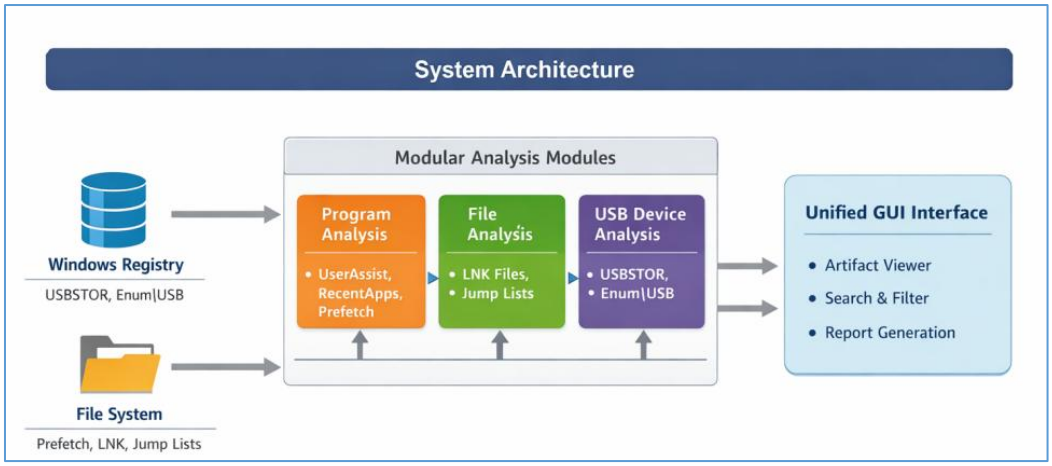

Figure 1 The overall system architecture

The system has three independent analysis modules were developed to handle program execution, file access, and USB device artifacts.

3.2 Data Extraction

As mentioned earlier, this work analyze multiple sources of information that are stored in different locations within Windows operating system, using different file formats and extensions. Below is a summary of the main information sources that were accessed and processed in this work:

1. UserAssist data were extracted from the Windows Registry Key located under HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\UserAssist, with ROT13 decoding applied to interpret encoded entries.
2. Recent Apps artifacts were done by accessing the registry keys that are located in HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Search\RecentApps. Each subkey represents a recently used application and contains metadata such as the application's name, path and the last execution time.
3. Prefetch files were decoded using custom parsers to extract program execution metadata to provide valuable evidence of user activity and helps reconstruct the timeline of events in digital forensic investigations. Each Prefetch file contains the name of the executable, the number of times it was run, the last execution timestamp and a list of file paths accessed during execution.
4. LNK files were processed to obtain details such as original file paths, creation, accessed and modified timestamps.
5. Jump List files were analyzed to get information about files that were accessed recently. These files are stored in two formats: automaticDestinations-ms and customDestinations-ms, that are located in the user's AppData directory.
6. USB device information was analyzed by extracting data from the Windows Registry, specifically from the following keys:

   HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\USB

   HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\USBSTOR

   Extracted information includes the device name, manufacturer, serial number, the first and last connection times.

In addition, the workflow of the data processing of this work is shown in Figure 2.

 ISSN:2302-8149
e-ISSN:2540-9719

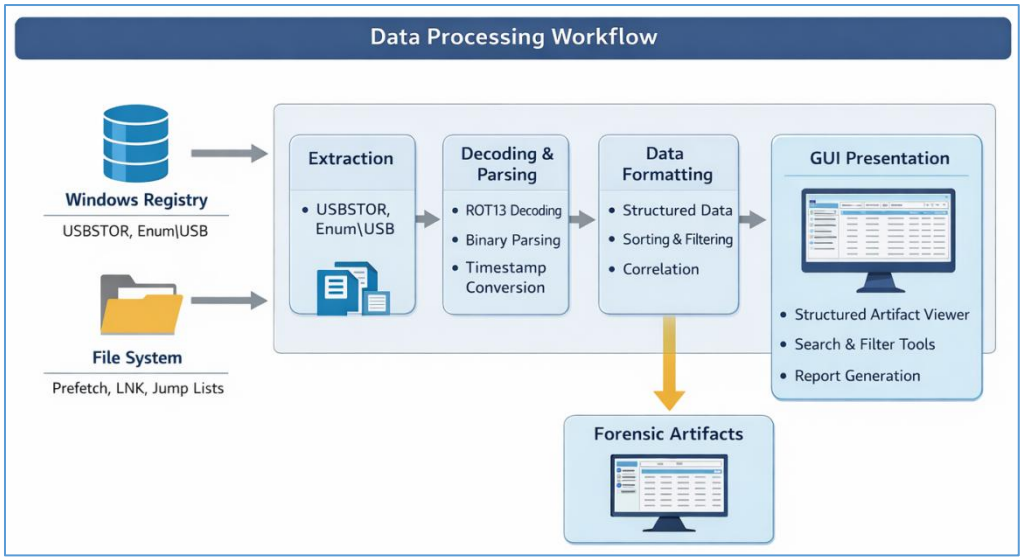

Figure 2 Digital forensic data processing and workflow

3.3. Implementation

The introduced tool was developed using C# language. The implementation focused on modularity, clarity, and the integration to support artifact extraction across multiple Windows subsystems.

A screenshot of the graphical user interface of the proposed tool is presented in Figure 3.

Figure 3 Graphical user interface of the proposed digital forensics tool

The main window enables investigators to navigate among program execution, file access, and USB device artifacts using menu options.

3.3.1 Development Environment

The development environment for the proposed system was carefully selected to ensure an efficient implementation of the forensic analysis tool. The tool was developed using the C# language under the .NET Framework 4.8, which provides a robust platform for Windows-based desktop applications. The development process was carried out using Visual Studio 2022 as an Integrated Development Environment (IDE). As well as, the Graphical User Interface (GUI) was designed using Windows Form technology. Finally, the tool was designed and tested to operate on Windows 10 (x64) operating system.

 ISSN:2302-8149
e-ISSN:2540-9719

3.3.2 Experimental Environment

All experiments were implemented on a personal computer running a 64-bit version of Windows 10. Moreover, the experiments were conducted on a computer with an Intel Core i7 processor and 8 GB of RAM.

3.3.3 Three-Layer Implementation Model

The system was implemented using three different layers, each layer is responsible for a specific aspect of the forensic process:

1. **Data Acquisition Layer:** This layer interacts directly with Windows Registry and file system. It is responsible for collecting raw digital traces generated by user activity, system events and connected USB devices. The main goal of this layer is to collect raw data.
2. **Parsing and Processing Layer:** Once the data is collected, the next step is to start decode and transform necessary raw data into structured and meaningful formats. This layer closes the gap between low-level data and the forensic investigators.
3. **Presentation Layer (User Interface):** The final layer takes the responsibility of for presenting the data to Figure 4 presents in a clear and interactive way. Forensic investigators can browse, sort, and filter the extracted artifacts easily and efficiently.

Table 2 introduces key modules that are implemented in this work.

Table 2 Key modules description

Module	Data Source	Goal
UserAssist Parser	(UserAssist Registry Key)	Track program execution – method 1
RecentApps Viewer	(RecentApps Registry Key)	Track program execution – method 2
Prefetch File Analyzer	Prefetch Files	Track program execution – method 3
LNK File Analyzer	(.lnk Files)	Detect files opened lastly – method 1
Jump List Parser	Jump Lists (Automatic & Custom Destinations)	Detect files opened lastly – method 2
USB Device Tracker	(USB & USBSTOR Registry Key)	Detect the USB devices connected to the computer.

3.3.4 Simulation Scenarios

To test the functionality of the introduced tool, different user activity scenarios were simulated:

- Scenario of Program Execution: in this scenario, many programs were run, and detection was verified via UserAssist, Prefeatch, and RecentApps
- Scenario of File Access: in this scenario, different files were opened and modified, and detection was tracked using LNK files and Jump Lists.
- Scenario USB Device: in this scenario, many USB devices were connected and disconnected, and detection was verified using USB and USBSTOR registry keys.

The implementation of the UserAssist Parser module is shown in Figure 4.

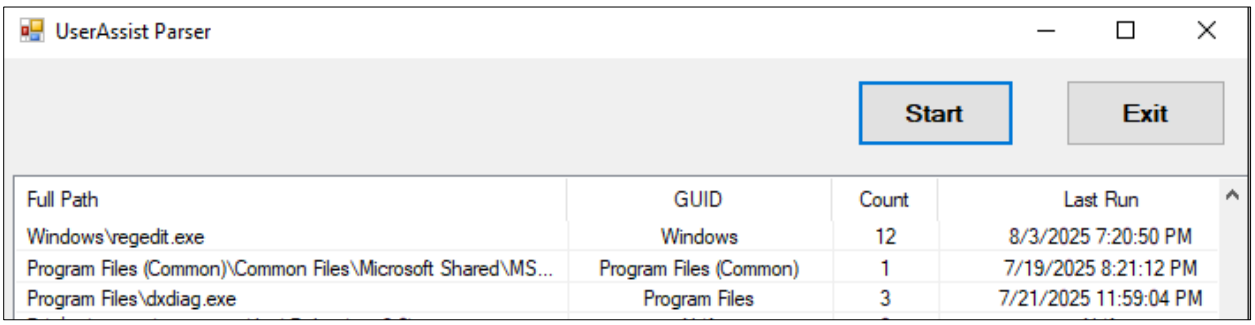

Figure 4 Window of the userassist parser module

As well as, Figure 5 shows the implementation of the RecenApps Viewer module.

 ISSN:2302-8149
e-ISSN:2540-9719

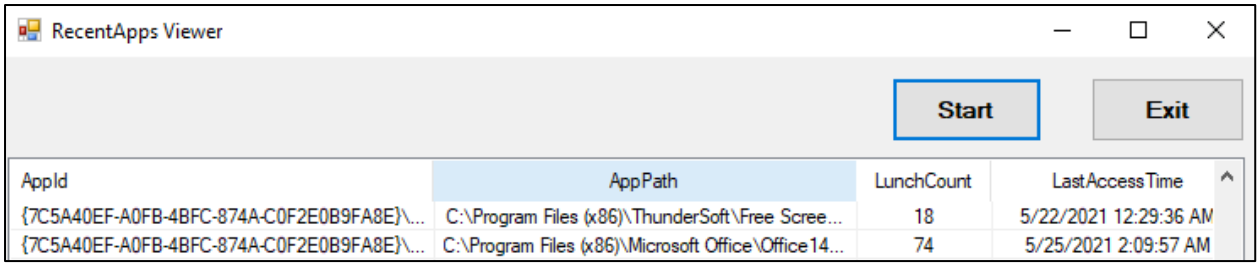

Figure 5 Window of recenapps viewer module

Whereas Figure 6 shows the implementation of Prefetch Analyzer module

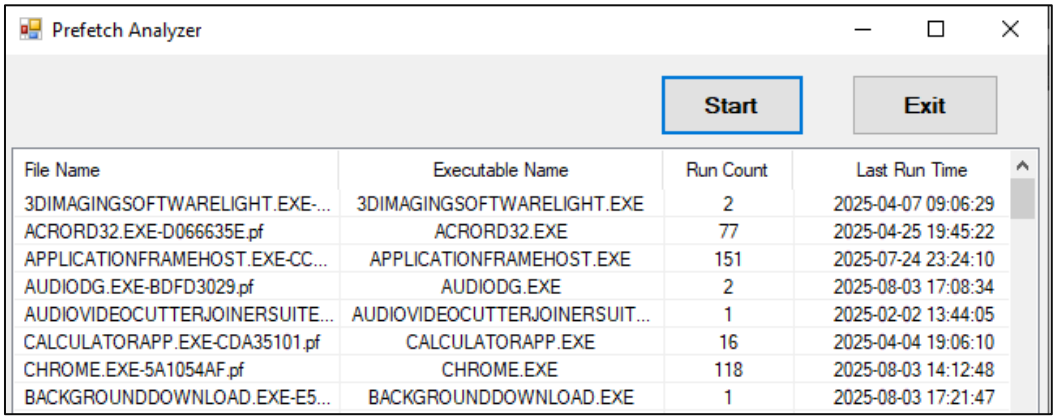

Figure 6 Window of the prefetch analyzer module

Moreover Figure 7 and Figure 8 show the result of the LNK File Analyzer module and JumpList Parser module, respectively.

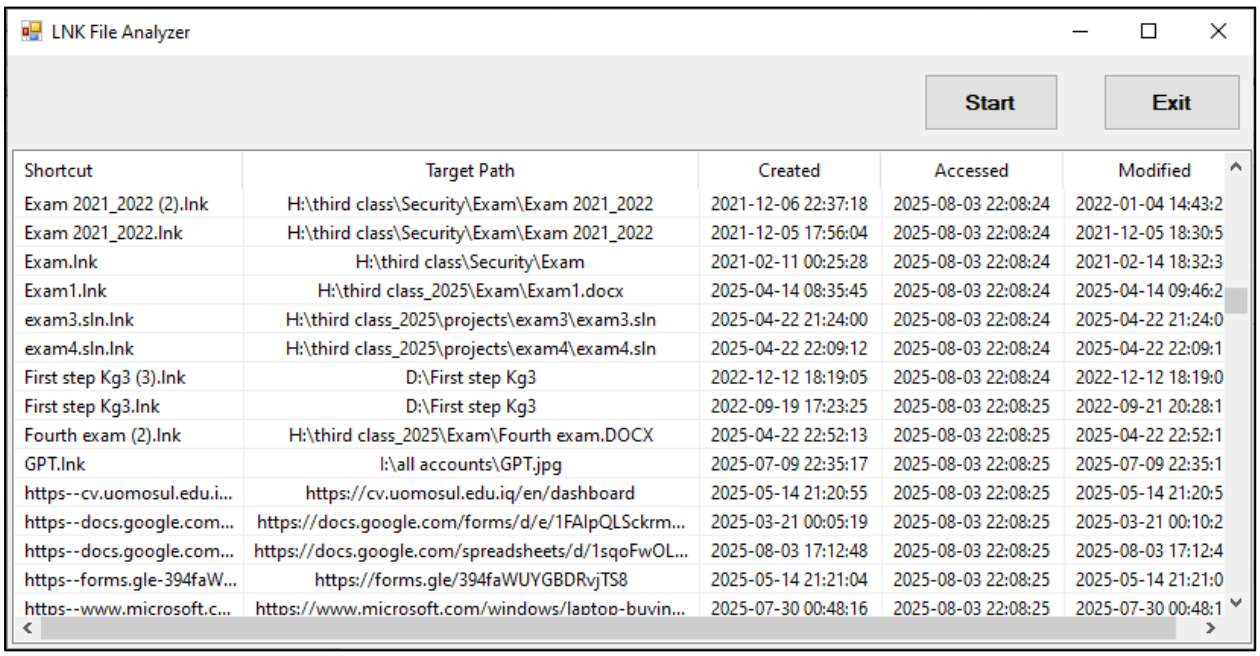

Figure 7 Window of the LNK file analyer

 ISSN:2302-8149
e-ISSN:2540-9719

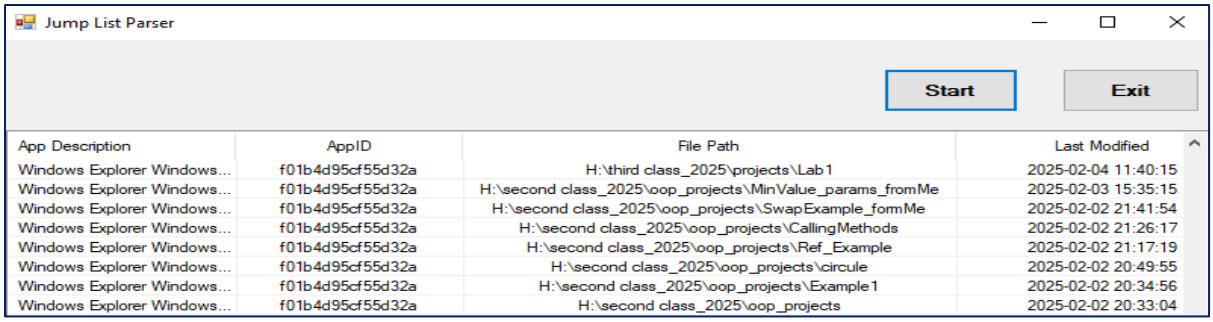

Figure 8 Window of jumplist parser module

Finally, Figure 9 presents the implementation of the USB Devices Tracker module.

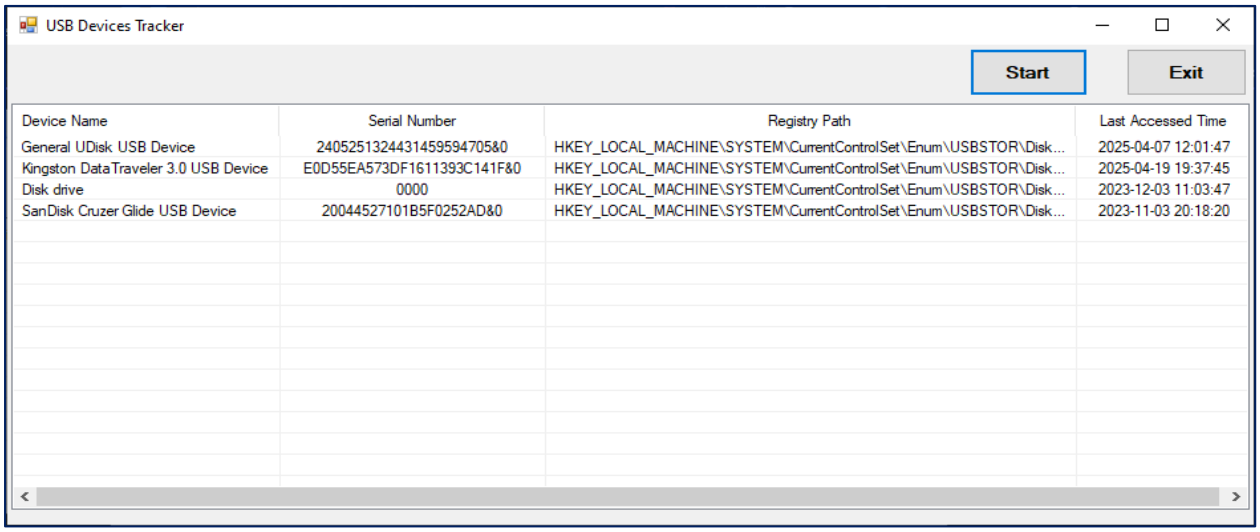

Figure 9 Window of USB devices tracker module

4. Results and Analysis

The introduced tool was evaluated on a Windows 10 (64-bit) to show its effectiveness in extracting and analyzing artifacts. The evaluation focused on the accuracy of extracted data and the usability of the integrated graphical interface.

The experimental results showed that the tool successfully extracted meaningful and human-readable forensic data based on different Windows artifacts; including program execution, file access, and USB device connection history.

4.1 Program Execution Analysis

Program execution was analyzed using three complementary modules: UserAssist Parser, RecentApps Viewer, and Prefetch File Analyzer. Each module uses a different Windows artifact and that provides unique insights. Table 3 summarizes these sources.

Table 3 Comparison of program execution artifacts

Feature	UserAssist	RecentApps	Prefetch
Data Source	Windows Registry	Windows Registry	Prefetch Files
Execution Timestamp	Last execution time	Last access time	Multiple execution times
Execution Count	Available	Limited	Available
Applications Covered	GUI-launched apps	Apps and some files	Apps and system services
Requires Decoding	Yes (ROT13)	No	No

 ISSN:2302-8149
e-ISSN:2540-9719

The evaluation showed that Prefetch files source provides the most detailed execution history, while UserAssist artifacts are particularly useful for identifying applications explicitly run by the user. On the other hand, RecentApps data offers rapid insight into recent user activity but lacks detailed execution counts. Considering the above comparison, where each method has its limitations, the proposed approach in this study highlights its strength and effectiveness by using multiple data sources to track user activities.

4.2 File Access Analysis

File access activity was tracked using two sources: the LNK File Analyzer and Jump List Parser modules. A comparison of these two sources is shown in Tabel 4.

Table 4 Comparison of file access artifacts

Feature	LNK Files	Jump Lists
Artifact Type	Shortcut files	Application-based lists
Storage Location	Recent folder	Automatic & Custom Destinations
Number of Entries	Single per shortcut	Multiple per application
Timestamps	Creation, access, modification	Last access and metadata

The results show that LNK files are effective to get individual file access, while Jump Lists provide a broader context of file usage in specific applications.

4.3 USB Device Analysis

USB device was analyzed using registry-based artifacts extracted from USBSTOR and Enum\USB keys. These sources have details about the connected devices, as explained in Table 5.

Table 5 Comparison of USB registry artifacts

Feature	USBSTOR	Enum\USB
Device Types	Storage devices only	All USB devices
Information Extracted	Serial number, last connection time	Manufacturer, product ID

5. Conclusion

In this work, a design and an implementation of a Windows digital forensics tool was introduced aiming to enhance evidence collection and analysis on Microsoft Windows systems. The proposed tool consider analyzing multiple Windows forensic artifacts, including UserAssist, RecentApps, Prefetch files, LNK shortcuts, Jump Lists, and USB device traces, within a single graphical application. Experimental results showed that the tool can extract forensic data in a clear and human-readable form, and reduce the need to use multiple separate tools. By combining different data sources, the introduced tool helps investigators to get a good understanding of user activity and system behavior. Overall, the tool got its importance by providing a practical, extensible, and user-friendly way for Windows artifact analysis. It supports investigations and addresses the gap identified in existing forensic methodologies. As a future work, new data sources, such as Windows Event Logs and browser artifacts, can be included. In addition, machine learning techniques may be used to analyze user behavior and detect unusual activities.

References

[1] D. Huici and R. Rodríguez, "*A Dataset of Windows System Binaries and Similarity Digests for Enhanced Forensic Analysis*," Data in Brief, Vol. 62, p. 111993, 2025, DOI: 10.1016/j.dib.2025.111993.

[2] Y. Huang, "*Application of Digital Forensics in Cybercrime Investigations*," Applied and Computational Engineering, Vol. 151, pp. 69–74, 2025, DOI: 10.54254/2755-2721/2025.22847.

[3] M. Murugamani, B. Unhelkar, and S. S., "*Digital Forensics in Cybercrime Investigations: Legal and Technical Challenges*," International Insurance Law Review, Vol. 33, 2026, DOI: 10.65677/iilr.33.S5.68.
[4] J. Han, J. Park, H. Chung, and S. Lee, "*Forensic Analysis of the Windows Telemetry for Diagnostics*," arXiv, 2020, DOI: 10.48550/arXiv.2002.12506.
[5] A. Budhrani, U. Singh, and B. Singh, "*Forensic Analysis of Windows 11 Prefetch Artifact*," in Proceedings of the 2022 IEEE Bombay Section Signature Conference (IBSSC), 2022, DOI: 10.1109/IBSSC56953.2022.10037260.
[6] A. Neyaz and N. Shashidhar, "*Windows Prefetch Forensics*," in Digital Forensics, Springer, 2022, DOI: 10.1007/978-3-031-10706-1_9.
[7] D. Joo, J. Lee, and D. Jeong, "*A Reference Database of Windows Artifacts for File-Wiping Tool Execution Analysis*," Journal of Forensic Sciences, Vol. 68, 2023, DOI: 10.1111/1556-4029.15240.
[8] J. Choi, J. Park, and S. Lee, "*Forensic Exploration on Windows File History*," Forensic Science International: Digital Investigation, Vol. 36, p. 301134, 2021, DOI: 10.1016/j.fsidi.2021.301134.
[9] A. Neyaz, N. Shashidhar, C. Varol, and A. Rasheed, "*Digital Forensics Analysis of Windows 11 Shellbag with Comparative Tools*," in Proceedings of the International Symposium on Digital Forensics and Security (ISDFS), 2022, pp. 1–10, DOI: 10.1109/ISDFS55398.2022.9800788.
[10] A. Neyaz and N. Shashidhar, "*USB Artifact Analysis using Windows Event Viewer, Registry and File System Logs*," Electronics, Vol. 8, p. 1322, 2019, DOI: 10.3390/electronics8111322.
[11] S. Pandey and M. Pal, "*A Review on Forensic Significance of Windows 10 Operating System*," International Journal of Research and Analytical Reviews (IJRAR), Vol. 7, No. 2, pp. 130–136, Jun. 2020.
[12] D. Sulekha, A. J. J, I. Venugopal, and M. Sabarinath, "*Cyber Forensics: Discovering Traces of Malware on Windows Systems*," 2020, DOI: 10.1109/RAICS51191.2020.9332496.
[13] D. Rathod and P. Sharma, "*Digital Forensic Analysis of Ransomware Infected Windows System*," JETIR, Vol. 6, No. 5, pp. 652–664, 2019.
[14] F. Fiadufe, K. Modi, K. Shukla, and F. O. Etyang, "*Forensic Investigation and Analysis of Malware in Windows OS*," Int. J. Electron. Secur. Digit. Forensics, Vol. 17, No. 1–2, pp. 169–182, 2025, DOI: 10.1504/IJESDF.2025.143477.
[15] J. Kim, B. Son, J. Yu, and J. Yun, "*AI-Driven Prioritization and Filtering of Windows Artifacts for Enhanced Digital Forensics*," Computers, Materials & Continua, Vol. 81, pp. 3371–3393, 2024, DOI: 10.32604/cmc.2024.057234.
[16] R. Jain, J. Gothania, P. D. Zaveri, M. Shah, P. G. Paija, and Y. P. Chawda, "*Leveraging AI for Behavioural Analysis of Digital Forensic Artifacts in Cybercrime Investigations*," in Cyber Forensic Frameworks for User-Centric Human Threat Intelligence Analysis, S. Kadry, M. Rai, and P. Tripathi, Eds., IGI Global, 2026, pp. 375–402, DOI: 10.4018/979-8-3373-4898-8.ch012.
[17] S. S. Iyengar, S. Nabavirazavi, Y. Hariprasad, H. B. Prasad, and C. K. Mohan, "*Digital Forensics: Tools, Techniques, and Methodologies*," in Artificial Intelligence in Practice. Cham, Switzerland: Springer, 2025, pp. 89–137, DOI: 10.1007/978-3-031-89327-8_3.
[18] N. Aleisa, "*The Study of Digital Forensics in KSA: Education, and Prosecution Capabilities: A Needs-based Analysis*," Electronics, Vol. 15, p. 316, 2026, DOI: 10.3390/electronics15020316.
[19] A. Awwad and A. Abdelsattar, "*Digital Evidence in Forensic Accounting: A Study in Saudi Legislation*," Cogent Social Sciences, Vol. 11, 2025, DOI: 10.1080/23311886.2025.2522958.