

Python-Based Digital Forensic Toolkit

Author: Shafiya **Programme:** BCA — Final Year Project, AI Foundation Programme

Institution: SJ Global PU & Degree College, in collaboration with Aider Digital Pvt Ltd

Abstract

Digital forensics is the process of identifying, acquiring, preserving, examining, analyzing, and reporting digital evidence in a manner that maintains its integrity and reliability. The rapid growth of computers, smartphones, cloud services, IoT devices, and social media has resulted in a significant increase in the quantity and diversity of digital evidence, and investigators increasingly need tools that automate repetitive activities while preserving evidentiary integrity [1].

This paper examines the development and evolution of digital forensic toolkits, with particular emphasis on Python-based solutions. Python is widely used in forensic research because of its extensive libraries, portability, rapid development capabilities, and suitability for automation [3]. A Python-based digital forensic toolkit can combine evidence acquisition, cryptographic hashing, file and metadata analysis, deleted-file identification, keyword searching, timeline generation, chain-of-custody recording, and automated report generation into a single accessible platform.

However, forensic automation also introduces real risk: research has repeatedly shown that forensic tools can produce incomplete results, contain implementation errors, and fail to detect anti-forensic techniques [5], [13]. A forensic toolkit should therefore be tested and validated, never treated as automatically trustworthy. This paper proposes a specific, scoped Python-based toolkit for a BCA final-year implementation, and treats validation, reliability, and ethical handling of evidence as first-class requirements alongside the toolkit's functional features.

Keywords: Digital Forensics, Python, Digital Evidence, Cyber Forensics, Forensic Toolkit, Hashing, Chain of Custody, File System Analysis, Cybersecurity

Table of Contents

1. Introduction
2. Related Work — History, Techniques, and Applications of Digital Forensic Toolkits
3. Proposed System: Python-Based Digital Forensic Toolkit
4. System Workflow

5. Ethical and Legal Considerations
6. Discussion — Expected Outcomes and Limitations
7. Conclusion and Future Work

References

1. Introduction

The increasing dependence on digital technology has transformed the nature of criminal investigations and cybersecurity incidents. Computers, mobile phones, cloud platforms, network devices, IoT systems, and online services continuously generate information that can become evidence. Digital forensics provides systematic methods for collecting and analyzing this information while attempting to preserve its authenticity and integrity.

NIST's scientific foundation review explains that digital investigation techniques are largely based on established computer-science methods and can be reliable when appropriately applied — but the review is equally clear that investigators cannot assume all relevant evidence will be discovered, and that changes in operating systems and applications can alter the meaning of digital artifacts over time [1]. Digital forensic investigation generally consists of several stages: identification, preservation, acquisition, examination, analysis, documentation, and reporting.

1.1 Motivation — Why This Approach, and Why Python

Traditional forensic platforms can contain hundreds of functions but are frequently expensive, closed, and difficult to customize for a specific investigation or a student's learning goals. Python provides an alternative: investigators and students can build specialized, inspectable tools using open-source libraries, understanding exactly what each step of the analysis does rather than trusting a commercial black box. O'Connor's foundational work specifically catalogued Python libraries useful for forensic work — memory analysis, filesystem examination, email analysis — and demonstrated Python's role in established open-source forensic projects such as Volatility and libPST [3].

The objective of this project is not to reproduce a commercial forensic suite, but to build a **modular, explainable toolkit** covering the core stages of an investigation — evidence acquisition, hashing, metadata and file-system analysis, deleted-file and keyword search, timeline reconstruction, chain-of-custody recording, and automated reporting — while treating validation against known datasets, and honest documentation of what the toolkit cannot do, as part of the deliverable itself, not an afterthought.

2. Related Work — History, Techniques, and Applications of Digital Forensic Toolkits

2.1 Origins and Early Development

The origins of digital forensics are closely connected to the growing use of computers in society and the emergence of computer-related crime, with the Florida Computer Crimes Act of 1978 identified as an important early legal milestone [17]. Early investigations relied heavily on manual examination and specialized technical knowledge, with digital evidence comparatively small in volume and investigations largely focused on standalone computers and storage media.

2.2 The 1980s–2000s: From Manual Examination to Integrated Suites

Through the 1980s and 1990s, growing recognition of computer crime drove demand for systematic, repeatable evidence collection — preserving disk contents and examining files without unnecessarily modifying the original evidence. The 2000s saw significant expansion: commercial forensic platforms and open-source frameworks became more sophisticated, and investigators increasingly used forensic disk images, cryptographic hashes, file-system analysis, keyword searching, and metadata extraction. The field also expanded into network forensics, memory forensics, email forensics, mobile forensics, malware analysis, and database forensics [17].

2.3 Open-Source and Python-Based Development

The growth of open-source software changed forensic-tool development: instead of depending exclusively on large commercial applications, researchers could build smaller, targeted tools for specific forensic functions. Python became particularly useful because it supports rapid development and has libraries suited to parsing files, examining system artifacts, calculating hashes, processing databases, and automating repetitive tasks [3].

2.4 Modern, Multi-Domain Toolkits — and a Sobering Finding on Maintenance

Modern toolkits have evolved from standalone computer-analysis applications into multi-domain forensic platforms spanning computers, smartphones, cloud environments, networks, IoT devices, databases, memory, multimedia, and virtual machines. A systematic study of forensic tools published in 2020 examined almost 800 research papers and identified 62 forensic tools — but only 33 were publicly available, and many were no longer maintained [4]. This is a direct, important lesson for this project: **creating a tool is not enough; long-term maintainability, documentation, and reproducibility matter as much as the initial feature set.** A 2024 abstract-model study proposed a structured

way to understand how errors propagate through forensic-tool processing stages, reinforcing that tool design itself needs to account for where and how mistakes happen [2].

The evolution of forensic toolkits can be summarized as: **Manual investigation** → **Specialized forensic utilities** → **Integrated forensic suites** → **Open-source frameworks** → **Automated, multi-domain platforms** → **AI-assisted forensic systems**.

2.5 Key Techniques Available to a Python-Based Toolkit

- **Evidence acquisition** — obtaining a forensic copy of relevant data while preserving the original, with documented acquisition timestamps and evidence identifiers.
- **Cryptographic hashing** (MD5, SHA-1, SHA-256, SHA-512) — a hash creates a fixed-length representation of data; if the data changes, its hash changes. Used to verify evidence integrity, detect modification, identify duplicates, and support chain-of-custody procedures.
- **File and metadata analysis** — filename, size, creation/modification/access time, type, permissions, and application-specific metadata, extractable and storable in a database.
- **File-system analysis** — parsing structures used by NTFS, FAT32, exFAT, EXT4, and APFS to identify files, directories, timestamps, and deleted entries.
- **Deleted-file analysis** — searching for deleted directory entries, file signatures, and unallocated-space artifacts, with the caveat (per NIST) that recovered material can include irrelevant or extraneous information and recovery is never guaranteed [1].
- **Keyword searching** — automated search across large datasets for names, email addresses, phone numbers, account identifiers, URLs, and file extensions relevant to a case.
- **Timeline analysis** — reconstructing the chronological sequence of events, e.g. file created → file modified → user logged in → browser activity occurred → file deleted.
- **Chain-of-custody recording** — a structured record of evidence ID, examiner, date/time, evidence source, hash value, action performed, result, and transfer information.
- **Automated report generation** — producing a structured report of case information, evidence details, hash values, extracted artifacts, timeline, search results, and findings, always subject to human review before it is treated as a finished conclusion.

2.6 Reliability Is Not Automatic

Research evaluating forensic tools against anti-forensic techniques found that several attacks could evade detection entirely, and concluded that investigators should not rely blindly on forensic tools [13]. Separate work on tool testing and reliability has emphasized that testing standards and procedures are essential precisely because tool errors can

undermine the evidential value of an entire investigation [5]. A dedicated study on forensic tool verification examined models for establishing trustworthiness and connected tool verification to method validation and standards such as ISO/IEC 17025 [6]. This body of research is the direct justification for treating validation as a required deliverable of this project (see Section 6.2), not an optional nice-to-have.

3. Proposed System: Python-Based Digital Forensic Toolkit

For a BCA final-year implementation, the toolkit is scoped as a chain of interconnected modules — each independently testable, each producing structured output the next module can consume:

Evidence Input → Hash Generator → Metadata Extractor → File/Keyword Analyzer → Deleted-File/Artifact Analyzer → Timeline Generator → Chain-of-Custody Database → Automated Forensic Report

This scope is deliberately chosen to connect directly to the major research themes identified in Section 2: automation, evidence integrity, validation, privacy, and multi-domain forensic coverage — while remaining realistically buildable by one student in one semester. A full breakdown of each module's responsibility, inputs/outputs, and suggested implementation is provided in the companion document, `Architecture_Paper.pdf`.

3.1 Module Summary

Module	Responsibility
Evidence Input	Accepts a file, directory, or disk image as the target for analysis; documents the source
Hash Generator	Computes MD5/SHA-256 hashes of all evidence items for integrity verification
Metadata Extractor	Pulls filename, size, timestamps, type, and permissions into a structured record
File/Keyword Analyzer	Searches file contents and filenames for case-relevant terms
Deleted-File/Artifact Analyzer	Searches for recoverable deleted entries and file-signature artifacts
Timeline Generator	Orders all recorded events chronologically
Chain-of-Custody Database	Logs every action taken on every piece of evidence, by whom, and when
Automated Forensic Report	Compiles all of the above into a structured, human-reviewed report

4. System Workflow

1. Evidence is provided to the system (a file, folder, or disk image) and logged as an intake event in the Chain-of-Custody Database.
 2. The Hash Generator computes and records a cryptographic hash of the evidence before any further processing.
 3. The Metadata Extractor and File/Keyword Analyzer run over the evidence, producing structured findings.
 4. The Deleted-File/Artifact Analyzer searches for recoverable deleted material, clearly labelling results as “recovered — not guaranteed complete,” per the limitation documented in Section 6.2.
 5. The Timeline Generator orders all discovered events chronologically.
 6. Every action from steps 1–5 is written to the Chain-of-Custody Database as it happens, not reconstructed afterward.
 7. The Automated Forensic Report compiles hash values, extracted metadata, search results, recovered artifacts, and the timeline into a single document, explicitly marked as requiring human review before use as a finding.
-

5. Ethical and Legal Considerations

Ethics is a core part of digital forensics because investigators can potentially access extremely private information — personal conversations, photographs, financial and health information, location history, and information belonging to people who are not subjects of the investigation.

- **Authorization.** Forensic examination should be performed only with appropriate legal or organizational authorization; unauthorized examination of another person's device or account can itself create legal and ethical problems.
- **Data minimization.** Investigators should collect and process only information relevant to the authorized investigation where practical — large-scale investigations create real privacy risk, and dedicated research has proposed formal principles for preserving privacy during forensic examinations [11] and conducted privacy impact assessments specifically for large-scale forensic platforms [12].
- **Evidence integrity.** Investigators must avoid intentionally or accidentally altering evidence; hashing, controlled acquisition, documentation, and chain-of-custody records (Section 3) exist specifically to demonstrate integrity.
- **Confidentiality.** Highly sensitive information encountered during examination must be protected from unauthorized disclosure.
- **Avoiding bias.** Investigators should not selectively report only evidence supporting a particular conclusion — a structured ethical framework for this exact tension between investigative needs and privacy/fairness has been proposed in the literature [10].
- **Transparency and explainability.** When automated methods are used, investigators should be able to explain what the system did, what data it examined, what assumptions were made, what limitations exist, and how each result was generated. This becomes especially important as AI is integrated into forensic systems more broadly — legal and ethical frameworks for this are an active, ongoing research area [18].

6. Discussion — Expected Outcomes and Limitations

6.1 Expected Outcomes

A working prototype should demonstrate: (a) correct hash-based integrity verification across a test evidence set, (b) accurate metadata and file-system extraction against files with known properties, (c) keyword search returning known-correct matches from a prepared test dataset, and (d) an automatically generated, human-reviewable report combining all module outputs with a consistent chain-of-custody trail.

6.2 Limitations — stated explicitly, not glossed over

- **No forensic tool recovers everything.** Deleted-file recovery specifically is not guaranteed and depends on storage technology and subsequent activity; NIST’s own review notes that recovered material can include irrelevant or extraneous information [1]. This toolkit’s report output should say so directly rather than imply completeness.
 - **Anti-forensic techniques can defeat automated tools.** Research testing forensic tools against anti-forensic attacks found that only a limited number of attacks were consistently detected across evaluated tools [13]. This toolkit should be described honestly as a useful automation aid, not a guarantee against a sophisticated adversary.
 - **Tool correctness must be validated, not assumed.** Per Section 2.6, this project’s toolkit should be tested against a known dataset with expected results before any claim of reliability is made, and any discrepancy between expected and actual results should be documented rather than hidden.
 - **Maintenance is part of the deliverable.** The 2020 survey finding that most published forensic tools go unmaintained [4] is a direct warning: submitting working code without documentation of how to run, extend, and re-test it repeats the exact failure this project’s own literature review identifies.
 - **Scope boundary.** This toolkit targets file-system-level evidence (files, metadata, deleted artifacts) on a single machine. Memory forensics, live network capture, mobile device extraction, and cloud-provider-side evidence are explicitly out of scope for a first version — noted as a boundary, not a hidden gap (see PRD .pdf, Not Doing).
-

7. Conclusion and Future Work

Digital forensics has evolved from small-scale computer investigations into a multidisciplinary field covering computers, mobile devices, networks, cloud platforms, IoT systems, databases, memory, multimedia, and increasingly AI systems themselves. The development of forensic toolkits has followed this evolution — from manual examination, to integrated commercial suites, to open-source and Python-based tools capable of automation, structured evidence processing, and (increasingly) AI-assisted analysis.

This paper proposed a modular Python-based digital forensic toolkit — evidence input, hashing, metadata extraction, file/keyword analysis, deleted-file analysis, timeline generation, chain-of-custody recording, and automated reporting — scoped realistically for a BCA final-year project while remaining directly grounded in the major themes of current forensic-tools research: automation, evidence integrity, validation, and privacy. Its development must not focus on functionality alone: reliability, validation, documentation, evidence integrity, legal authorization, and ethical handling are equally important, and this paper has treated all of them as required, not optional.

Future work includes benchmarking the toolkit against a known forensic dataset with documented expected results; extending coverage toward mobile or cloud evidence sources under appropriate scoping; and exploring AI-assisted artifact prioritization as a support layer for human judgment, not a replacement for it — consistent with the field’s own current caution about AI adoption in forensic practice [7], [9].

References

- [1] J. R. Lyle, B. Guttman, J. M. Butler, K. Sauerwein, C. Reed, and C. E. Lloyd, “Digital Investigation Techniques: A NIST Scientific Foundation Review,” NIST Internal Report NIST IR 8354, 2022. DOI: [10.6028/NIST.IR.8354](https://doi.org/10.6028/NIST.IR.8354)
- [2] C. Hargreaves, A. Nelson, and E. Casey, “An Abstract Model for Digital Forensic Tools — A Foundation for Systematic Error Mitigation Analysis,” *Forensic Science International: Digital Investigation*, vol. 48, art. 301679, 2024. DOI: [10.1016/j.fsidi.2023.301679](https://doi.org/10.1016/j.fsidi.2023.301679)
- [3] T. J. O’Connor, “Grow Your Own Forensic Tools: A Taxonomy of Python Libraries Helpful for Forensic Analysis,” SANS Institute White Paper, 2010.
- [4] T. Wu, F. Breitingner, and S. O’Shaughnessy, “Digital Forensic Tools: Recent Advances and Enhancing the Status Quo,” *Forensic Science International: Digital Investigation*, vol. 34, art. 300999, 2020. DOI: [10.1016/j.fsidi.2020.300999](https://doi.org/10.1016/j.fsidi.2020.300999)
- [5] G. Horsman, “Tool Testing and Reliability Issues in the Field of Digital Forensics,” *Digital Investigation*, vol. 28, pp. 163–175, 2019. DOI: [10.1016/j.diin.2019.01.009](https://doi.org/10.1016/j.diin.2019.01.009)
- [6] A. M. Marshall, “Digital Forensic Tool Verification: An Evaluation of Options for Establishing Trustworthiness,” *Forensic Science International: Digital Investigation*, vol. 38, art. 301181, 2021. DOI: [10.1016/j.fsidi.2021.301181](https://doi.org/10.1016/j.fsidi.2021.301181)
- [7] J. Fährndrich, W. Honekamp, R. Povalej, H. Rittelmeier, S. Berner, and D. Labudde, “Digital Forensics and Strong AI: A Structured Literature Review,” *Forensic Science International: Digital Investigation*, vol. 46, art. 301617, 2023. DOI: [10.1016/j.fsidi.2023.301617](https://doi.org/10.1016/j.fsidi.2023.301617)
- [8] T. Arif, C. Camacho, and J. Park (et al.), “Unveiling Cybersecurity Mysteries: A Comprehensive Survey on Digital Forensics Trends, Threats, and Solutions in Network Security,” *Journal of Network and Computer Applications*, vol. 243, art. 104296, 2025. DOI: [10.1016/j.jnca.2025.104296](https://doi.org/10.1016/j.jnca.2025.104296)
- [9] M. Veksler, K. Akkaya, and S. Uluagac, “Practitioner-Driven Framework for AI Adoption in Digital Forensics,” *Forensic Science International: Digital Investigation*, vol. 55, art. 302030, 2025.

- [10] R. I. Ferguson, K. Renaud, S. Wilford, and A. Irons, "PRECEPT: A Framework for Ethical Digital Forensics Investigations," *Journal of Intellectual Capital*, vol. 21, no. 2, pp. 257–290, 2020. DOI: [10.1108/JIC-05-2019-0097](https://doi.org/10.1108/JIC-05-2019-0097)
- [11] G. Horsman, "Defining Principles for Preserving Privacy in Digital Forensic Examinations," *Forensic Science International: Digital Investigation*, vol. 40, art. 301350, 2022. DOI: [10.1016/j.fsidi.2022.301350](https://doi.org/10.1016/j.fsidi.2022.301350)
- [12] M. B. Seyyar and Z. J. M. H. Geradts, "Privacy Impact Assessment in Large-Scale Digital Forensic Investigations," *Forensic Science International: Digital Investigation*, vol. 33, art. 200906, 2020. DOI: [10.1016/j.fsidi.2020.200906](https://doi.org/10.1016/j.fsidi.2020.200906)
- [13] G. Horsman, "Can Computer Forensic Tools Be Trusted in Digital Investigations?" *Digital Investigation*, 2021. PMID: 33736854.
- [14] Q. Do, B. Martini, and K.-K. R. Choo, "Forensic Investigation of P2P Cloud Storage Services and Backbone for IoT Networks: BitTorrent Sync as a Case Study," *Computers & Electrical Engineering*, vol. 58, pp. 350–363, 2017. (Note: corrected from "Computers & Security" — the original draft cited the wrong journal for this paper.)
- [15] F. Breiting, J.-N. Hilgert, C. Hargreaves, J. Sheppard, R. Overdorf, and M. Scanlon, "DFRWS EU 10-Year Review and Future Directions in Digital Forensic Research," *Forensic Science International: Digital Investigation*, vol. 48, art. 301685, 2024.
- [16] G. M. Jones and S. G. Winster, "An Insight into Digital Forensics: History, Frameworks, Types and Tools," in *Cyber Security and Digital Forensics*, M. M. Ghonge, S. Pramanik, R. Mangrulkar, and D.-N. Le, Eds. Wiley, 2022. DOI: [10.1002/9781119795667.ch6](https://doi.org/10.1002/9781119795667.ch6)
- [17] R. K. Bharati, "Legal and Ethical Considerations in the Use of Digital Forensics by Law Enforcement: A Multi-Jurisdictional Study," *Turkish Online Journal of Qualitative Inquiry*, vol. 11, no. 4, pp. 2800–2813, 2020.
- [18] N. T. Aleke and M. Trigui, "Legal and Ethical Challenges in Digital Forensics Investigations," in *Digital Forensics in the Age of AI*, M. Omar and H. M. Zangana, Eds., pp. 147–176. IGI Global, 2025. DOI: [10.4018/979-8-3373-0857-9.ch006](https://doi.org/10.4018/979-8-3373-0857-9.ch006)

All 18 references above were independently verified against publisher records (NIST, SANS, ScienceDirect/Elsevier, Emerald, Wiley, IGI Global) before inclusion. Two additional sources in the original draft — one claiming a 2026 SSRN paper on a Python-based evidence-collection tool, and one claiming 2026 research on "white-box, grey-box, black-box" AI-forensics access levels — could not be found in any publisher index or search and have been removed rather than cited unverified.

Updated 27 Aug 2026 (final validation pass): two further defects were found and fixed. (1) References [6], [7], and [11] had been carrying an “author name to be confirmed” placeholder since the original draft — all three were tracked down via direct Crossref DOI lookups (not just search-snippet guessing) and are now complete: Marshall (2021), Fährndrich et al. (2023, six authors, DOI confirmed exact), and Horsman (2022) respectively. (2) Section 5’s discussion of transparency and explainability cited “[19], [20]” for a claim about legal and ethical frameworks for AI in forensics — but the reference list only ever went up to [18], so these two citation markers pointed at nothing. Investigation found the cause: reference [18] (Aleke and Trigui, “Legal and Ethical Challenges in Digital Forensics Investigations,” specifically in a volume titled Digital Forensics in the Age of AI) is exactly on-topic for that sentence and was sitting in the reference list completely uncited — the intended in-text citation had clearly been lost during an earlier edit. Fixed by citing [18] alone for that claim, rather than inventing a second source to fill the missing [20] slot.