

Article

A Layered High-Value Evidence Area (HVEA) Model for Selective Windows Digital Forensics Imaging: NIJ-Aligned Design and Empirical Validation

Osayomore O. Aigbogun , Cihan Varol * and Narasimha Shashidhar 

Department of Computer Science, Sam Houston State University, Huntsville, TX 77341, USA;
oaa020@shsu.edu (O.O.A.); nks001@shsu.edu (N.S.)

* Correspondence: cvarol@shsu.edu

Abstract

Digital forensics increasingly operates under extreme data growth, where exhaustive bit-wise imaging of modern storage is constrained by time, storage, and processing cost. Selective imaging and artifact prioritization methods reduce acquisition volume, but they typically order artifacts by content, location, or offense type rather than by the relationships that make evidence interpretable. As a result, they risk discarding the contextual artifacts on which attribution and corroboration depend. This paper introduces the High-Value Evidence Area (HVEA) pyramid, a dependency-oriented abstraction that organizes Windows forensic artifacts into ten operational layers (A–J) grouped into four evidentiary tiers: attribution anchor, primary payload, behavioral contextualization, and structural corroboration, in which acquisition order follows interpretive prerequisites rather than artifact salience. The model is evaluated on nine Windows forensic images spanning Windows XP, Vista, and Windows 11, combining retrospective analysis of the M57-Patents corpus with a controlled fourteen-day behavioral experiment. Across systems and operating system generations, HVEA layers exhibit stable evidentiary function despite changing artifact implementations; behavioral execution telemetry persists even where user content is sparse or deliberately concealed; and a three-source timestamp corroboration pattern consistently converges within thirty seconds across independent OS mechanisms, providing an empirically grounded defensibility threshold for event reconstruction. The results support a dependency-ordered, NIJ-aligned selective acquisition strategy that preserves interpretive context while reducing acquisition footprint by approximately 95% (a 15–20× reduction) on the controlled Windows 11 image. The present evaluation is scoped to Windows environments; extending the tier mappings to other platforms is identified as future work.

Keywords: artifact prioritization; digital forensics; evidence dependency; evidentiary hierarchy; forensic acquisition; intelligence in digital forensics; M57-Patents; NTFS artifacts; selective imaging; Windows forensics



Academic Editor: Taeshik Shon

Received: 8 June 2026

Revised: 2 July 2026

Accepted: 7 July 2026

Published: 9 July 2026

Copyright: © 2026 by the authors.
Licensee MDPI, Basel, Switzerland.
This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

Digital forensics investigations increasingly operate under conditions of extreme data growth. Modern computing environments generate large volumes of digital artifacts across storage devices, network infrastructures, cloud services, and connected devices. As a result, investigators frequently encounter terabytes of potentially relevant data during routine forensics examinations, creating significant delays in evidence acquisition, processing, and analysis [1,2].

Traditional digital forensics practice has historically relied on full bit-wise imaging of storage media in order to preserve the complete state of a system for later examination. While this approach ensures evidentiary completeness, it also produces large volumes of data that may contain only a small proportion of information relevant to a particular investigation. As storage capacities and system complexity continue to increase, the practicality of exhaustive acquisition strategies has become increasingly challenged by time, storage, and computational constraints [3,4].

To address these challenges, researchers have proposed a variety of strategies designed to reduce the scale of digital forensics acquisition. Selective imaging techniques attempt to acquire only relevant data objects while preserving sufficient metadata to maintain evidentiary provenance [3]. Data reduction approaches have demonstrated that investigators can significantly reduce acquisition volumes while still preserving meaningful evidentiary artifacts [1]. Similarly, artifact-focused investigative methodologies prioritize system artifacts such as registry entries, filesystem metadata, and application traces to enable investigators to begin analysis before complete disk images are obtained [5,6].

Despite these advances, a critical conceptual gap remains: existing selective imaging approaches lack a formal evidentiary dependency structure, which risks losing essential contextual artifacts during acquisition. Most existing approaches focus primarily on improving acquisition efficiency or reducing data volumes. However, they often lack a formal structure that explains how different classes of digital artifacts relate to one another within the broader evidentiary context of an investigation. In practice, investigators must interpret digital artifacts by considering relationships among system configuration records, user-generated data, behavioral traces, and structural system logs. Without a clear representation of these evidentiary dependencies, selective acquisition approaches risk overlooking artifacts that provide critical contextual information for interpreting investigative findings.

This paper addresses this limitation by introducing the High-Value Evidence Area (HVEA) pyramid, a dependency-oriented abstraction model for organizing digital forensics artifacts, developed and empirically validated for Windows environments. The HVEA model makes a distinct and substantive contribution relative to prior work in three ways. First, unlike artifact cataloging approaches such as Kondapally [7] that enumerate forensically significant Windows artifacts without specifying their interpretive relationships, the HVEA model encodes those dependencies explicitly in a layered hierarchy. Second, unlike forensic triage techniques such as McKeown et al. [8] that focus narrowly on specific artifact categories, HVEA provides a cross-domain dependency structure that spans identity, payload, behavior, and structural corroboration. Third, unlike National Institute of Justice (NIJ) investigative frameworks that serve as descriptive taxonomies without prescribing acquisition order, HVEA translates investigative categories into a dependency-ordered acquisition sequence grounded in empirical validation. Rather than categorizing artifacts solely by investigative theme, the HVEA model organizes evidence sources according to their evidentiary roles and interpretive dependencies. The framework identifies ten operational artifact layers (A–J) grouped into four evidentiary tiers representing attribution anchors, primary evidentiary objects, behavioral contextualization, and structural corroboration.

To evaluate the empirical behavior of these artifact classes, this study combines retrospective analysis of an established forensics dataset (M57-Patents) [9] with a controlled behavioral experiment in a contemporary Windows environment, across nine Windows images spanning Windows XP, Vista, and Windows 11, two decades of Windows development. The empirical observations indicate that artifacts corresponding to the HVEA layers exhibit consistent presence across independent systems and user activity patterns, supporting the stability of the proposed evidentiary abstraction.

The primary contributions of this work are threefold:

- A dependency-oriented abstraction model that organizes digital forensics artifacts into structured evidentiary layers, explicitly encoding interpretive relationships among artifact classes.
- An empirical evaluation demonstrating the cross-system stability of these artifact classes using both legacy forensics datasets and controlled experimental activity across OS generations spanning twenty years of Windows development.
- A conceptual foundation and NIJ-aligned operational mapping for selective imaging strategies within the Windows operating system that prioritize High-Value Evidence Areas while preserving the contextual relationships necessary for reliable forensics interpretation.

The remainder of this paper is organized as follows. Section 2 reviews existing research on digital forensics acquisition, selective imaging, and artifact prioritization. Section 3 introduces the HVEA pyramid model and its evidentiary dependency structure. Section 4 describes the experimental design and data collection methodology used in this study. Section 5 presents empirical observations across the analyzed systems. Section 6 presents the operational mapping between NIJ investigative categories and the HVEA artifact layers. Section 7 discusses implications, limitations, and future directions. Section 8 concludes.

2. Background and Conceptual Foundations

Digital forensics investigations operate within increasingly complex technological environments characterized by rapid data growth, heterogeneous storage systems, and diverse digital ecosystems. Modern computing platforms generate large volumes of artifacts across operating systems, network infrastructures, cloud services, and connected devices, often resulting in forensics datasets that reach the terabyte scale [1,2,10].

Historically, forensics practice emphasized complete system preservation through full bit-wise imaging. However, the growth of storage capacities and system complexity has motivated research into acquisition strategies that reduce data collection overhead while maintaining evidentiary reliability.

To situate the proposed High-Value Evidence Area (HVEA) framework within existing research, this section reviews four related themes: investigative framing models, digital forensics acquisition practices, selective imaging approaches, and artifact prioritization strategies.

2.1. NIJ Digital Investigative Framing

Digital forensics investigations often rely on conceptual frameworks that guide investigators in identifying potential sources of digital evidence. One widely referenced example is the classification scheme provided by the National Institute of Justice (NIJ), which organizes digital information into investigative categories such as user-generated content, system artifacts, communication records, and web activity [11].

While useful for investigative orientation, these frameworks function primarily as descriptive taxonomies rather than operational acquisition models. They identify potential evidence sources but do not specify how artifacts should be prioritized during acquisition under practical constraints such as time, storage capacity, and legal limitations.

Related research highlights similar challenges. Al-Dhaqm et al. note that the expansion of digital forensics subdomains has introduced inconsistencies in investigative procedures and terminology [12]. Pedapudi and Vadlamani likewise emphasize the importance of structured acquisition procedures to preserve evidentiary integrity and documentation reliability [13]. These observations illustrate the need for models that connect conceptual investigative frameworks with structured evidence acquisition strategies.

2.2. Digital Forensics Acquisition Models

The acquisition phase is a critical component of the digital forensics process. Traditionally, investigators rely on full bit-wise imaging to preserve the complete state of a storage device through sector-level duplication. Although this method ensures evidentiary completeness and allows investigators to revisit previously overlooked data, the rapid growth of modern storage capacities has significantly increased the operational burden associated with comprehensive acquisition.

Freiling et al. identify several technological factors that further complicate modern forensics acquisition, including encryption mechanisms, solid-state storage behavior such as wear leveling and TRIM operations, and the volatility of system memory [2]. These developments have encouraged researchers and practitioners to explore acquisition strategies capable of reducing data collection scope while preserving evidentiary reliability.

2.3. Selective Imaging and Forensics Triage Approaches

The first and most direct response to the volume problem is to abandon exhaustive imaging in favour of acquiring only what appears relevant. Stüttgen et al. formally define selective imaging as the acquisition of relevant digital objects together with the metadata required to identify those objects within their storage environment [3], establishing the conceptual basis for partial acquisition. Subsequent work has pursued this goal along three lines, each effective on its own terms yet sharing a common dependence on an externally supplied notion of relevance.

A first line selects by predefined profile: Grier and Richard's sifting collectors image only the disk regions matched by rules keyed to file type, name, timestamp, and metadata, achieving order-of-magnitude reductions in imaging time while preserving device-level verifiability [14]. A second line selects by data reduction objective: Quick and Choo's data reduction by selective imaging (DRbSI) targets high-value file categories across devices and reports substantial volume savings, while acknowledging the risk of discarding contextual evidence [1]. A third line reframes triage as a performance problem: Roussev et al. treat acquisition and processing as a latency-optimization task, reconstructing files during imaging so that analysis proceeds in parallel [15]. Related efforts extend selective acquisition to volatile memory [16] and to network-scale capture [17], and complementary reduction strategies address scale through deduplicated acquisition [4] and distributed processing [18].

Volatile memory acquisition, however, differs fundamentally from the dead-box disk extraction considered in this study: it requires live-acquisition tooling, is order-sensitive, and risks irreversible data loss through live execution. It is therefore treated as outside the scope of the present work. Across these approaches, the selection criterion is external to the evidence itself, that is, a content profile, a reduction target, or a throughput budget, and as Quick and Choo observe in surveying the field, the practitioner must still decide which information is necessary to answer the investigative question [19]. Selection, therefore, continues to rest on investigator judgment, and artifacts whose value lies in supporting the interpretation of other artifacts are easily omitted.

2.4. Artifact Prioritization and Layering Approaches

If selective imaging addresses how much to collect, a second body of work asks which artifact classes are worth collecting first. These approaches prioritize system-generated traces that reveal user activity early, before full-disk analysis is complete. Lee et al. propose a stepwise methodology that examines filesystem metadata and system artifacts before progressing to user-generated files [5]. Ahmadi et al. prioritize key Windows artifact registry entries, event logs, browser histories, and page files so that analysis can begin

while imaging continues [6]. Bhandari and Jusas organize forensic events into a four-level abstraction for timeline reconstruction, showing that progressive structuring improves interpretability [20], and McKeown et al. exploit the Windows thumbnail cache for rapid visual triage [8].

Two studies edge closer to the idea of dependency without formalizing it. Work on timestamp manipulation shows that altering one artifact tends to produce inconsistencies in others, implying that artifacts carry interdependent evidentiary relationships [21,22]. Kondapally identifies registry hives, event logs, Prefetch files, MFT entries, USN journal records, and browser artifacts as forensically significant, and observes that no single artifact is decisive; reliability emerges through cross-correlation [7]. Yet this stops at observation: it establishes that artifacts must be corroborated without specifying which classes depend on which, or in what order they should be acquired.

The pattern across this literature is that artifacts are prioritized individually but treated as independent sources, rather than organized within a structure that represents their interpretive dependencies. Table 1 summarizes representative studies and their limitations along exactly this axis.

Table 1. Comparison of prior approaches to artifact prioritization and selective evidence acquisition in digital forensics.

Study	Focus	Key Contribution	Limitation
Lee et al. (2009) [5]	Stepwise Investigation	Phased investigative methodology prioritizing system artifacts before user files.	Does not provide a formal evidentiary hierarchy.
Stüttgen et al. (2013) [3]	Selective Imaging	Formal definition of selective imaging and hierarchical data object acquisition.	Relies heavily on investigator judgment when selecting relevant objects.
Kondapally (2015) [7]	Artifact Cataloguing	Systematic identification of forensically significant Windows artifacts, including registry, event logs, Prefetch, MFT, and USN journal records.	Identifies artifact cross-correlation as necessary but does not provide a formal dependency structure or acquisition ordering guidance.
Grier & Richard (2015) [14]	Selective Imaging	Sifting collectors: rapid profile-based imaging of disk regions matched by file type, name, timestamp, and metadata.	Selection driven by predefined profiles rather than inter-artifact evidentiary dependency.
Ahmadi et al. (2018) [6]	Artifact-first Acquisition	Prioritization of Windows system artifacts to accelerate investigative analysis.	Focuses primarily on OS artifacts rather than broader evidence relationships.
Du et al. (2018) [4]	Deduplicated Acquisition	Storage-efficient forensics acquisition with verifiable image reconstruction.	Dependent on centralized infrastructure and duplication ratios.
Quick & Choo (2018) [1]	IoT Data Reduction	Data Reduction by selective imaging (DRbSI) for cross-device forensics investigations.	Risk of losing contextual evidence during reduction.
McKeown et al. (2019) [8]	Forensic Triage	Use of Windows thumbnail cache artifacts for rapid investigative triage.	Limited to specific artifact categories.
Song & Li (2020) [18]	Big Data Forensics	Distributed forensics processing framework integrating big data technologies.	Focuses on scalability rather than artifact prioritization.

2.5. Research Gap

The chain above reveals a single, consistent omission. Full imaging guarantees completeness but does not scale; selective imaging scales but delegates relevance to investigator

judgment; artifact prioritization identifies high-value classes but treats them as independent. What none of these approaches capture is the simple fact that evidence only becomes meaningful in a particular order. Some artifacts only make sense when the records that explain, attribute, or time-anchor them are present. If a payload is collected without the pieces that show who produced it, when it happened, or what it connects to, then it is technically acquired but evidentially weak.

This paper addresses that omission by introducing the High-Value Evidence Area (HVEA) pyramid, a dependency-oriented abstraction that organizes digital forensics artifacts into layered evidentiary tiers. By ordering artifact classes according to an interpretive prerequisite rather than content salience, reduction target, or crime label, HVEA makes explicit the three properties absent from prior work: which artifact classes converge, across which dependency boundaries, and to what corroboration threshold. In doing so, it provides a conceptual bridge between descriptive investigative frameworks and operational acquisition strategy.

3. High-Value Evidence Area (HVEA) Pyramid Model

Digital artifacts rarely exist as isolated sources of evidence. Instead, artifacts generated by different operating system components such as filesystems, execution mechanisms, logging services, and user account management exhibit evidentiary dependencies that enable investigators to interpret user actions within a broader system context. For example, a user-created document may be associated with execution traces generated by the application that opened it, filesystem metadata reflecting modification times, and system event logs that record related activity. These relationships suggest that digital evidence can be organized according to functional roles that contribute to investigative interpretation.

The High-Value Evidence Area (HVEA) model proposes a dependency-oriented abstraction that organizes digital forensics artifacts into layered evidentiary tiers. Rather than treating artifacts as independent evidence sources, the HVEA framework groups artifacts according to their investigative role and interpretive relationship to other artifacts.

Figure 1 illustrates the resulting evidentiary hierarchy, which consists of four dependency-ordered levels.

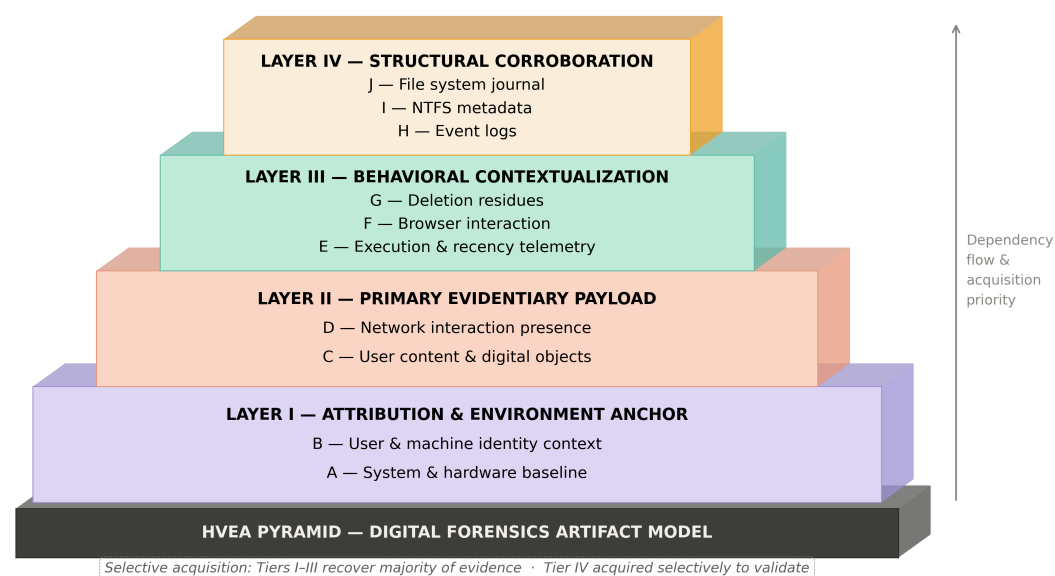


Figure 1. The High-Value Evidence Area (HVEA) pyramid illustrating dependency-oriented evidentiary layers in digital forensics environments.

The order of these levels is not a matter of convenience or prevalence, but of interpretive necessity: each tier exists because the tier below it, however complete, leaves a specific interpretive question unanswered. We therefore derive the four tiers as a chain, in which every level is forced into being by a gap in the one preceding it.

Layer I: Attribution and Environment Anchor: Interpretation cannot begin until the system itself is characterized and its activity is attributable to an actor. This foundational layer contains the artifacts that establish the identity and configuration context of a system like system identifiers, user account structures, and configuration artifacts that define the attribution scope of an investigation. Without this anchor, every subsequent artifact is evidence without an owner: a file, a log entry, or a timestamp that cannot be tied to a user, a configuration, or a clock. Attribution is therefore the prerequisite for all higher interpretation, and the chain begins here.

Layer II: Primary Evidentiary Payload: Once the environment is anchored, the investigation requires the objects it is actually about. Artifacts in this layer represent the objects of investigative interest, which are user-generated files, communications, and other digital objects associated with investigative hypotheses. Yet an attributed payload still answers only what exists, not how it came to exist: a document tied to a user and a system carries no inherent record of how it was created, accessed, or handled. That unanswered question forces the next tier.

Layer III: Behavioral Contextualization: To explain how payload objects were produced and manipulated, the model draws on artifacts that capture user interaction with the system, which range from program execution records, browsing activity, and transient interaction traces. Behavioral artifacts supply the causal narrative that the payload alone lacks, linking the attributed identity of Layer I to the handling of the payload in Layer II. But these traces are themselves generated by the operating system and may be incomplete, ambiguous, or deliberately manipulated; on their own, they cannot certify their own accuracy. This limitation forces the final tier.

Layer IV: Structural Corroboration: The highest tier contains artifacts generated by system subsystems such as event logs and filesystem metadata that record the same activity through mechanisms independent of the behavioral traces they validate. Because these records are produced by separate OS subsystems, their agreement with lower-layer inferences constitutes genuine corroboration rather than restatement, and their disagreement exposes gaps or tampering. Layer IV closes the chain by confirming the temporal and structural consistency of everything inferred below it.

When read in sequence, the four tiers form a dependency chain rather than a ranked list: attribution makes the payload interpretable, behavioral context explains the payload's history, and structural corroboration validates that history through independent record. Dependency is thus defined here as interpretive reliance, where each higher tier requires contextual grounding from the tiers below it for attribution, semantic interpretation, or chronological validation, and it is this reliance, not artifact prevalence, investigative theme, or offense classification, that fixes the acquisition order.

3.1. Dependency-Ordered Layered Abstraction

The ten operational layers (A–J) represent substrate-level evidentiary classes abstracted into four dependency tiers reflecting interpretive reliance and corroborative hierarchy. While investigative procedures may target individual layers, acquisition prioritization decisions are guided at the tier level within the HVEA abstraction.

Layer IV artifacts provide structural validation independent of semantic payload interpretation, enabling corroboration of lower-layer inferences regarding identity, activity chronology, and artifact lifecycle.

Table 2 presents the dependency-ordered abstraction of HVEA tiers, and Table 3 defines the ten operational layers at the substrate level.

Table 2. Dependency-ordered abstraction of HVEA tiers. The hierarchy reflects evidentiary dependency and validation flow for selective acquisition rather than artifact frequency or crime-type grouping.

Dependency Tier	Substrate Classes and Evidentiary Function
Layer I—Attribution & Environment Anchor	A—System & Hardware Baseline Establishes the operating environment required for interpretation (OS version, filesystem type, disk layout, installed program baseline, system time configuration). B—User & Machine Identity Context Defines attribution scope and identity anchoring (user accounts/SIDs, profile directories, registry system hives, group membership, machine identifiers). All higher layers rely on Layer I for interpretive stability.
Layer II—Primary Evidentiary Object Layer	C—User Content (Evidentiary Payload) Contains primary evidentiary objects (documents, media, financial records, downloads). These artifacts represent the semantic focus of investigation but require Layer I context for attribution. D—Network Interaction Presence Captures outward interaction indicators (host indicators, communication traces, network profile artifacts) that contextualize boundary-crossing activity under Layer I attribution.
Layer III—Behavioral Contextualization Layer	E—Execution & Recency Telemetry Provides execution chronology and recency traces (Prefetch, Shimcache, Amcache, recent files) linking identity (Layer I) to payload access and handling (Layer II). F—Browser & Application Interaction Reconstructs human–application and web interaction patterns (history, cookies, web storage, IndexedDB, typed URLs, bookmarks) contextualizing acquisition, research, and dissemination behavior. G—Deletion & Transient Residues Encodes cleanup intent and artifact lifecycle behavior (Recycle Bin, temporary artifacts, cache traces, thumbnail cache) reflecting post-creation handling of payload.
Layer IV—System-Level Independent Corroboration	H—Event Logs (System Temporal Chronology) Provides OS-level authentication and activity chronology (security, system, application logs) capable of validating or contradicting lower-layer inferences. I—NTFS Metadata (Structural Ground Truth) Supplies file lifecycle validation via metadata structures (e.g., \$MFT entries, MAC timestamps, structural attributes) independent of semantic interpretation. J—Filesystem Journal & Structural Chronology Provides journaling-based change records via \$UsnJrn1 (filesystem change reasons: create, modify, rename, delete), enabling fine-grained reconstruction of file lifecycle and concealment behavior. Distinct from \$LogFile, which supports transactional crash recovery and is not used for evidentiary chronology reconstruction.

The operational layers presented in Table 3 collectively define the evidentiary substrate from which investigative reconstruction emerges. While each layer represents a distinct class of artifacts, the investigative value of the model lies not in isolated artifact recovery but in the cross-layer dependency relationships that enable corroborative interpretation.

Table 3. Operational HVEA layers (A–J) defining substrate-level evidentiary classes used in the dependency-based abstraction. Each layer groups artifacts according to investigative function, typical artifact sources, and interpretive role within the HVEA pyramid model.

Layer	Layer Name	Core Artifacts	Investigative Purpose	Typical Sources
A	System & Hardware Baseline	OS version, filesystem type, disk layout, installed program baseline, system time configuration	Establish environmental operating context and platform constraints	System configuration repositories, disk metadata, registry baseline keys
B	User & Machine Identity Context	User accounts/SIDs, profile directories, registry hives, group membership, machine identifiers	Establish attribution boundary and identity anchoring	SAM/SYSTEM hives, user profile paths, registry keys
C	User Content (Evidentiary Payload)	Documents, media files, financial records, archives, downloads	Recover primary evidentiary objects under identity attribution	User directories (Documents, Desktop, Downloads, Pictures)
D	Network Interaction Presence	Host indicators, communication traces, network profile artifacts	Identify boundary-crossing activity and external interaction context	Network configuration stores, connection history artifacts
E	Execution & Recency Telemetry	Prefetch files, Shimcache, Amcache, recent files, shortcut traces	Reconstruct execution chronology linking identity to payload access	Prefetch directory, AppCompat artifacts, Recent destinations
F	Browser & Application Interaction	History, cookies, web storage, IndexedDB, typed URLs, bookmarks, download metadata	Contextualize acquisition, research, and dissemination behavior	Browser SQLite databases, profile directories
G	Deletion & Transient Residues	Recycle Bin entries, temporary artifacts, thumbnail cache, cleanup traces	Detect deletion intent and artifact lifecycle behavior	\$Recycle.Bin, %TEMP%, cache directories
H	Event Logs (System Temporal Chronology)	Security, system, and application event records	Independently corroborate authentication, boot, and service chronology	Windows Event Log repositories
I	NTFS Metadata (Structural Ground Truth)	\$MFT entries, MAC timestamps, structural attributes	Validate file lifecycle and structural consistency independent of semantic interpretation	NTFS metadata structures
J	USN Journal & File Change Chronology	\$UsnJrnl entries, structural change indicators [†]	Reconstruct fine-grained filesystem change chronology and concealment behaviour	NTFS journaling artifacts

[†] \$LogFile is retained by the OS for transactional crash recovery and is structurally distinct from \$UsnJrnl. \$LogFile records are not used for evidentiary chronology reconstruction in this study; only \$UsnJrnl change-reason records are used for file lifecycle analysis. See Cross-generation implementation differences are examined later in this paper.

The HVEA abstraction formalizes this principle by organizing artifact classes according to their interpretive dependency structure. At the foundational level, Layers A and B establish the attribution boundary. System configuration, machine identifiers, and user account structures define the operating environment within which subsequent artifacts

must be interpreted. Without this attribution anchor, higher-layer artifacts such as browser activity, execution traces, or user-generated files cannot be reliably associated with a specific user or system context.

Layers C and D represent the primary evidentiary payload layer, containing artifacts that typically form the focal point of investigative interest. Documents, media files, archives, and network interaction traces often provide the direct evidentiary material relevant to investigative hypotheses. However, these artifacts alone do not establish how or when the artifacts were accessed, modified, transmitted, or deleted.

The interpretive bridge between attribution and payload is provided by the behavioral contextualization layer (Layers E–G). Execution telemetry, application interaction traces, and transient artifact residues encode patterns of system use that link identity context to evidentiary payload. These behavioral traces have been widely used in forensics investigations to reconstruct user activity sequences and interaction timelines [23,24].

Finally, the uppermost tier (Layers H–J) provides system-level corroboration independent of semantic artifact interpretation. Event logs, NTFS metadata structures, and journaling records serve as structural validation mechanisms capable of confirming or contradicting behavioral interpretations derived from lower layers.

This dependency-oriented structure provides a conceptual basis for selective evidence acquisition. Rather than acquiring all artifact domains indiscriminately, investigators may prioritize acquisition of layers that maximize interpretive stability while minimizing redundant data capture. The HVEA model therefore supports acquisition strategies in which evidentiary classes are collected in a dependency-consistent order, enabling early-stage reconstruction while preserving the ability to validate interpretations using independent structural artifacts.

3.2. Implications for Selective Evidence Acquisition

The dependency-oriented organization of the HVEA pyramid has direct implications for evidence acquisition strategies. Because higher evidentiary layers depend on contextual grounding from lower layers, acquisition can be performed in a dependency-consistent sequence that preserves interpretive stability while reducing unnecessary data collection. Under this model, acquisition begins with the attribution and environmental anchor (Layer I), then proceeds to primary evidentiary objects (Layer II), behavioral contextualization artifacts (Layer III), and finally system-level corroboration artifacts (Layer IV). This layered acquisition strategy enables investigators to progressively expand the evidentiary scope while maintaining interpretive coherence. Recent empirical evaluations of privacy-aware selective imaging approaches have demonstrated up to 40% reduction in non-relevant data exposure while retaining 95% of evidentially relevant artifacts [25], supporting the viability of dependency-ordered acquisition strategies of the kind proposed by the HVEA framework.

3.3. Motivating Example: HVEA-Guided Acquisition in Practice

To make the model concrete and operationally relatable, consider a hypothetical insider-threat investigation in which an employee is suspected of exfiltrating proprietary documents before resigning. An investigator responding without an acquisition framework might immediately image the entire hard drive, an operation that on a modern 2TB disk could take several hours and yield hundreds of gigabytes of data with uncertain relevance. Under the HVEA model, the investigator proceeds differently. The acquisition begins with Layer I (attribution anchor): within minutes, the SAM hive and SYSTEM registry key are extracted, establishing the primary user account SID, machine name, and timezone configuration. These artifacts cost kilobytes to collect and immediately bound the attribution context for all subsequent analysis. The investigator then targets Layer II (primary

evidentiary payload): the user's documents, desktop, downloads, and pictures directories are imaged selectively. A scan reveals recently modified spreadsheets and a zip archive in the downloads folder; these artifacts are consistent with bulk data collection. Layer II collection takes minutes rather than hours and focuses storage on investigatively relevant material. Layer III (behavioral contextualization) is acquired next. Prefetch files (Layer E) confirm that an archiving utility was executed seventeen times in the week before the resignation date, a pattern not visible in the user content layer alone. Browser history (Layer F) reveals repeated navigation to a personal cloud storage provider. Recycle bin artifacts (Layer G) show that several files were deleted the day before the employee's last day. These behavioral artifacts, each tiny in footprint, collectively reconstruct a coherent activity sequence linking identity to payload handling. Finally, Layer IV (structural corroboration) is acquired selectively to anchor the behavioral reconstruction temporally. The \$UsnJrnl journal (Layer J) confirms the precise creation and deletion timestamps for the archived files, and Windows event logs (Layer H) record the logon session during which the archiving tool was executed. The three-source corroboration—SAM last-login (Layer B), NIC LastWrite (Layer D), and event log session record (Layer H)—converges within seconds, making the reconstructed timeline defensible in a court proceeding.

This walkthrough illustrates the practical value of dependency-ordered acquisition: the investigator obtains an investigatively coherent reconstruction in a fraction of the time required by full-disk imaging, with a clear evidentiary dependency chain that supports both analysis and courtroom presentation. The same dependency logic generalizes across investigation types, as operationalized in the NIJ priority matrix in Section 6.

3.4. Practical Implementation: Tool Integration and Automation

The HVEA model is designed to be tool-agnostic at the level of evidentiary function, while accommodating specific artifact implementations at the operational level. In practice, forensic practitioners can map HVEA layers directly to capabilities within widely used tool-sets. For example, Autopsy and Exterro FTK support targeted extraction of registry hives (Layers A–B), user directories (Layer C), browser databases (Layer F), Prefetch and Shimcache entries (Layer E), and event logs (Layer H) as discrete artifact categories rather than requiring full-disk processing before analysis begins. Command-line tools such as the Kroll Artifact Parser and Extractor (KAPE) [26] support direct HVEA-layer-aligned collection through configurable target definitions that match artifact paths to the substrate classes defined in Table 2.

The increasing role of automation in forensics workflows further motivates the need for structured acquisition models that can guide automated tools in prioritizing evidence classes [27]. The HVEA dependency hierarchy is explicitly designed to support such integration: by encoding acquisition priority at the tier level, the model provides a stable interface between the investigative framework and tool-level implementation. Future work should address the formal specification of HVEA-compatible acquisition profiles for existing triage platforms, as well as the development of automated dependency-checking mechanisms that flag potential attribution failures before Tier II–IV extraction begins.

4. Methodology and Empirical Analysis

The empirical component of this study evaluates the stability and investigative utility of artifact classes within the proposed HVEA pyramid across legacy and contemporary operating system environments.

The methodology combines two complementary data sources: (1) retrospective analysis of legacy forensic case images and (2) a controlled behavioral experiment conducted in a modern Windows environment. This dual-source design enables observation of artifact

behavior across both realistic investigative datasets and controlled user activity. Together, these datasets support evaluation of artifact persistence, behavioral trace consistency, and evidentiary dependency relationships across operating system generations.

4.1. Why This Design? Methodological Motivation

A central challenge in evaluating forensic artifact models is the tension between ecological validity and experimental control. Purely synthetic environments may generate artifacts with incorrect provenance; purely archival case studies constrain the range of observable activity patterns. The dual-source design used here deliberately addresses both constraints. The M57-Patents dataset provides historically grounded investigative scenarios in which artifact generation arises from realistic user behavior over a simulated organizational period. The controlled Windows 11 experiment complements this by enabling direct causal observation. Because the researcher performs the activity, the ground truth of what occurred is known, allowing verification that artifact generation is consistent with the HVEA model's dependency predictions. The fourteen-day controlled experiment was structured around the four HVEA acquisition tiers rather than arbitrary user behavior. Activities were selected to trigger artifact generation across each tier: account configuration (Layer I), document creation and file downloads (Layer II), application execution and browser interaction (Layer III), and system events such as installation, reboot, and deletion (Layer IV). This design choice means the experiment is not merely observational; it constitutes a functional test of whether the HVEA dependency structure correctly predicts artifact co-occurrence patterns across tiers. The complementarity between the two data sources therefore provides both external validity (M57) and internal validity (Windows 11 experiment) for the proposed model.

4.2. Data Sources and Case Selection

Legacy forensic datasets were drawn from the *M57-Patents* digital forensic scenario dataset provided by the Digital Corpora repository [9]. The scenario simulates several weeks of activity within a corporate environment and includes disk images, memory captures, and network traces generated during realistic user interactions. The forensic validity of scenario-based datasets for empirical evaluation have been examined through quantitative metric frameworks [28], supporting the use of the M57-Patents dataset as a representative basis for cross-system artifact analysis.

For this study, the analysis focused on the Windows XP and Windows Vista workstation disk images associated with the user environments commonly identified in the dataset as Charlie, Pat, Terry, and Jo. These systems represent independent workstations within the simulated organization and contain organically generated activity, including document creation and modification, file transfers, web browsing, application execution, and archive handling.

All disk images were examined using a standardized forensic workflow designed to preserve evidentiary integrity while enabling the extraction of artifacts across multiple evidentiary domains, including system configuration structures, user activity traces, application interaction records, and filesystem metadata. These datasets therefore provide the historical empirical basis for evaluating artifact classes represented within the HVEA model.

4.3. Controlled Behavioral Induction (Windows 11)

To complement the legacy datasets, a controlled behavioral experiment was conducted in a Windows 11 virtual machine environment. While the M57 dataset provides historically grounded investigative scenarios, the controlled environment enables direct observation of how specific user actions generate artifacts across evidentiary layers.

The system was configured with commonly used applications including a web browser, office productivity tools, and compression utilities. Over a fourteen-day observation period, routine user activities were performed, including document creation and editing, file downloads, archive compression and extraction, browser interaction, application execution, file movement, and deletion operations.

These activities were intentionally designed to trigger artifact generation across multiple HVEA layers, allowing observation of how ordinary user behavior propagates through the operating system artifact ecosystem. Unlike purely simulated environments, in which artifact provenance may lack causal grounding in real system events [29], the controlled experiment in this study was conducted on a live Windows 11 installation performing genuine user interactions, ensuring that all extracted artifacts reflect authentic OS-generated evidence.

Following completion of the fourteen-day activity simulation, the Windows 11 virtual machine was forensically acquired to preserve the system state for artifact analysis. The virtual disk was imaged using Exterro FTK Imager (version 8.2), producing an Expert Witness Format (E01) forensic image.

The acquisition process included automated cryptographic verification using MD5 and SHA1 hashing to ensure data integrity. All subsequent artifact examination was performed on the verified forensic image rather than on the live virtual machine in order to maintain evidentiary integrity and analytical reproducibility.

4.4. Deterministic Descriptor-to-Artifact Mapping

Figure 2 summarizes the end-to-end HVEA acquisition workflow; the descriptor-to-layer mapping stage (Stage B) is detailed here. To ensure reproducible artifact classification, artifact descriptors extracted during forensic examination were deterministically mapped to the operational HVEA layers (A–J). Descriptors were first normalized to reduce variation arising from naming differences across tools, documentation sources, and artifact labeling conventions.

Normalized descriptors were then mapped using a rule-based phrase-to-prefix dictionary (PHRASE_TO_PREFIX), which associates artifact phrases with their corresponding HVEA layer identifiers. Layer identifiers were processed using a fixed ordering sequence (PREFIX_ORDER) defined as {A, B, C, D, E, F, G, H, I, J} to ensure consistent representation across datasets.

This deterministic mapping procedure reduces subjective artifact categorization and provides a transparent mechanism for translating heterogeneous artifact descriptions into the structured evidentiary framework required for cross-case analysis.

4.5. Cross-Case Artifact Stability Analysis and Dependency Validation

Artifact stability was assessed by examining the presence or absence of artifact descriptors mapped to each HVEA operational layer across the analyzed systems. Layer recurrence was evaluated by comparing whether artifacts associated with each layer appeared in the legacy Windows XP datasets and the controlled Windows 11 experimental environment.

The purpose of this comparison was to determine whether the proposed evidentiary classes remain observable across independent environments despite differences in operating system generation and activity patterns. To support interpretation of cross-system recurrence, observed artifacts were examined in relation to the four dependency tiers defined by the HVEA model, assessing whether recurring artifacts served attributional, payload, behavioral, or corroborative evidentiary functions. This layer-oriented analytical logic links empirical observation to conceptual validation by examining whether recurring artifact classes occupy stable evidentiary roles across environments.

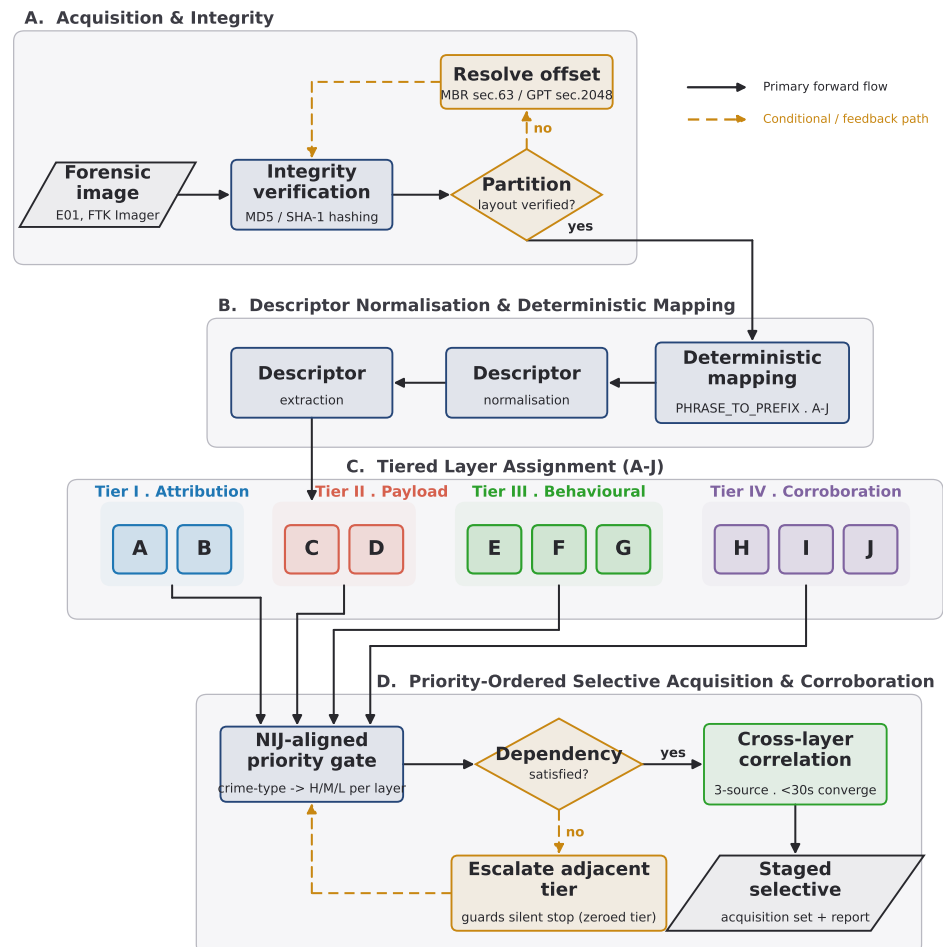


Figure 2. End-to-end HVEA selective acquisition workflow. Stage A acquires and integrity-verifies the image, resolving partition offset failures before processing. Stage B normalizes artifact descriptors and maps them deterministically to layers A–J via PHRASE_TO_PREFIX and PREFIX_ORDER. Stage C assigns the ten layers to the four dependency tiers (attribution, payload, behavioral, corroboration). Stage D performs NIJ-prioritized selective acquisition, in which a dependency-check gate escalates to the adjacent tier rather than terminating when a high-priority layer yields below threshold, preventing silent under-collection, and closes with cross-layer corroboration.

Table 4 summarizes the fourteen-day activity protocol used to intentionally trigger artifact generation across multiple HVEA layers.

Table 4. Fourteen-day controlled activity timeline and targeted artifact generation.

Day	Simulated Activity	Artifact Sources	Artifact Trigger	HVEA Layers
1	System initialization and account configuration	System configuration files, user profile structures	Identity and environment baseline	A,B
2	Initial document creation and file storage	User documents, filesystem metadata	User-generated evidentiary objects	C,I
3	Application execution and document editing	Prefetch entries, recent file lists	Program execution traces	E,I
4	Web browsing and search activity	Browser history, cache, cookies	Browser interaction artifacts	F

Table 4. *Cont.*

Day	Simulated Activity	Artifact Sources	Artifact Trigger	HVEA Layers
5	File downloads and archive handling	Download records, archive metadata	User content acquisition	C,I
6	File deletion and recycle bin interaction	Recycle bin artifacts, filesystem journal entries	Deletion and recovery traces	G,J
7	System reboot and idle system behavior	Windows event logs, system startup traces	System logging artifacts	H
8	Additional document editing and file modification	File timestamps, filesystem metadata	Metadata lifecycle changes	I
9	Web browsing, search queries, and document downloads	Browser history, download records	Network-mediated user activity	F,C
10	Software installation and application usage	Program execution artifacts, installation records	Execution and installation telemetry	E,H
11	Software removal and configuration updates	Uninstall logs, registry modifications	System configuration change artifacts	H,I
12	File organization, renaming, and archive creation	Filesystem metadata, journal entries	Metadata mutation and journaling	I,J
13	Additional browsing and file downloads	Browser artifacts, user file traces	Combined browser and payload traces	F,C
14	System shutdown and session termination	Event logs, system state artifacts	System lifecycle termination traces	H

Artifact observations were aggregated by HVEA layer and compared across systems to evaluate cross-case stability and the distribution of evidentiary roles.

5. Empirical Validation and Comparative Yield

The following section moves beyond theoretical abstraction to ask a simpler but harder question: when investigators open nine independent forensics images spanning two operating system generations and fifteen years of platform evolution, do the ten HVEA layers reliably surface? And if they do, do they surface with enough evidentiary substance to justify selective acquisition strategies built around them?

Artifact observations extracted from the eight M57-Patents disk images (Charlie, Pat, Terry, and Jo, each captured at two distinct points in time) were examined alongside those generated during the controlled Windows 11 behavioral experiment described in Section 4.

Throughout this section, operational layer identifiers (A–J) refer to the substrate-level evidentiary classes defined in Table 3: Layers A–B form the attribution anchor, C–D the primary evidentiary payload, E–G the behavioral contextualization tier, and H–J the structural corroboration tier. The analysis addresses two related questions: whether the proposed evidentiary layers appear consistently across independent systems, and whether their density and character support the dependency-ordered acquisition logic the HVEA framework proposes.

5.1. Cross-System Layer Presence and Artifact Density

The most direct test of the HVEA model is whether its ten operational layers produce observable, extractable artifacts across independent forensics environments. Table 5 reports both layer presence and the artifact counts recovered from each system during forensics

examination. Rather than reducing the evidence to binary indicators, the table reflects actual extraction yield.

All ten HVEA layers produced extractable artifacts across every analyzed system, confirming layer-level presence. However, the more consequential finding lies in how artifact density distributes across layers and subjects.

Table 5. Artifact density across HVEA operational layers and analyzed systems. Values represent extracted artifact counts per layer. A dash (—) indicates zero yield; ✓ indicates confirmed presence with a single structural instance. Approximate values (~) reflect estimated counts from partial extraction. Italicized tier summaries provide interpretive context for observed density patterns.

Layer	Charlie		Jo		Pat		Terry		Win11
	Nov	Dec	Nov	Dec	Nov 16	Nov 17	Nov 12	Nov 19	14 d
Layer I—Attribution Anchor									
A (system baseline)	✓	✓	✓	✓	✓	✓	✓	✓	✓
B (logins)	13	51	24	20	16	18	8	4	33
Layer A is structurally invariant across all systems; Layer B login counts reflect observed session activity. Together these layers provide a compact but indispensable attribution anchor at minimal acquisition cost.									
Layer II—Primary Evidentiary Payload									
C (user files)	2	42	277	20	5	11	1	—	30
D (network)	✓	✓	✓	✓	✓	✓	✓	✓	✓
Layer C varies substantially by user role and machine history (0–277 files). Low Layer C counts do not imply low activity; Terry’s near-zero yield reflects a reimaged system rather than user inactivity. Layer D presence is universal.									
Layer III—Behavioral Contextualization									
E (Prefetch)	36	84	85	68	65	84	93	50	503
F (browser URLs)	~6	~50	~25	~40	~25	~30	13	~20	200+
G (deleted items)	—	—	3	6	—	—	—	19	34
Layer E is the most consistently dense behavioral layer across all systems, persisting even where Layer C is sparse, a critical asymmetry signaling activity that content-first triage would miss. Layer G absence indicates no recovered deletions, not necessarily that none occurred.									
Layer IV—Structural Corroboration									
H (event records)	11	59	30	27	23	23	319	~40	33
I (MFT entries)	721	2918	2169	1751	23	30	2105	2730	32,813
J (journal MB)	29	78	44	41	36	38	185	121	307
Tier IV dominates raw artifact volume, particularly Layer I MFT entries and Layer J journal size. This tier is generated independently of user intent and provides temporal corroboration resistant to the same concealment strategies that degrade Tiers II and III.									

Attribution layers (A–B) behave as structural anchors rather than variable evidence sources. Layer A remains invariant across all systems, while Layer B varies in proportion to observed login activity but remains consistently populated. These layers, therefore, provide a stable identity context with minimal acquisition cost.

Layer C (user content) exhibits the greatest variance across the dataset, ranging from zero artifacts on Terry’s reimaged system to 277 files on Jo’s older workstation. This variation is not noise. Rather, it reflects differences in user role, machine history, and investigative relevance. Sparse Layer C content may indicate limited user-created payload, but does not imply low activity.

Layer E (execution telemetry) is the most consistently dense behavioral layer across the analyzed systems. Even where Layer C is sparse, Layer E remains populated, demonstrating that execution activity persists independently of user-generated file volume. This cross-layer

asymmetry is itself an investigative signal. This finding is consistent with Kondapally's observation that execution artifacts such as Prefetch files and filesystem structures such as the USN journal capture system activity independently of user-generated content, providing investigative value even when primary payload artifacts are minimal [7].

Layer J similarly demonstrates that structural corroboration persists even when higher-level user artifacts are thin. System-managed journals capture filesystem activity independently of user intent and therefore preserve evidence that may not appear in content or interaction layers.

Taken together, these observations show that artifact distribution is not uniform across the HVEA pyramid. Instead, the density pattern supports the dependency structure proposed by the model: attribution layers remain compact but essential, behavioral layers provide rich contextualization, and structural layers preserve high-volume corroborative evidence independent of direct user behavior.

Figure 3 makes the structural regularity visible. Reading down any single column, the four tiers reproduce the same vertical signature on every system: a compact, near-constant Tier I band; a volatile Tier II in which Layer C swings across two orders of magnitude; a persistently populated Tier III; and a volume-dominant Tier IV. Reading across rows, the Windows 11 column is uniformly brighter yet preserves this signature unchanged, so its higher counts shift the absolute scale without altering the relative distribution. The pattern that matters is therefore vertical, not horizontal: evidentiary organization is governed by layer dependency rather than raw artifact volume, and the same acquisition ordering applies whether a system yields hundreds of artifacts or tens of thousands.

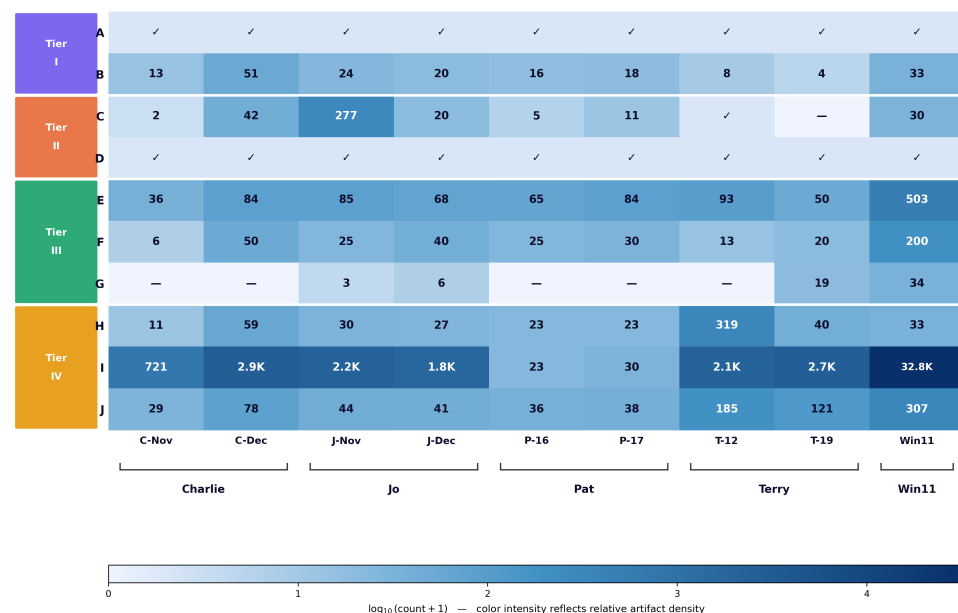


Figure 3. Log-scaled artifact density across HVEA operational layers and analyzed systems ($\log_{10}(\text{count} + 1)$ transformation applied). Cell annotations report raw extracted artifact counts; a dash indicates zero yield and a tick indicates confirmed single-instance presence. Horizontal lines demarcate the four evidentiary tiers. Color intensity reflects relative density on the log scale, preserving cross-system comparability despite the substantially higher artifact volumes observed in the Windows 11 environment. Tier I (A–B) remains compact and invariant, providing a stable attribution anchor at minimal acquisition cost. Tier II (C–D) exhibits high variance in Layer C (0–277 files), reflecting differences in user role and machine history rather than absence of activity. Tier III (E–G) is the most evidentiary-dense tier relative to its acquisition footprint: Layer E persists even where Layer C is sparse, a cross-layer asymmetry that content-first triage would miss. Tier IV (H–J) dominates raw volume through MFT and journal records, generated independently of user intent and resistant to the concealment strategies that degrade Tiers II and III.

5.2. Functional Consistency Across Operating System Generations

While the previous analysis confirms that HVEA layers produce observable artifacts across systems, an additional challenge arises from structural differences in how these artifacts are implemented across operating system generations. Although the underlying artifact structures vary between systems, the evidentiary roles they fulfill within the HVEA framework remain consistent.

To evaluate whether these structural variations affect the stability of the HVEA abstraction, Table 6 presents empirically observed differences in artifact implementation across Windows XP, Vista/7, and Windows 11.

Table 6. Empirically observed artifact nomenclature and structural changes across operating system generations. All entries reflect changes encountered during examination of the M57-Patents dataset and the Windows 11 experimental image.

Layer	Evolution (XP → Vista/7 → Windows 11)	Forensic Impact
B (identity)	XP: SAM RID sequence begins at 1003 following built-in accounts. Vista and later: first user account typically assigned RID 1000. Windows 11 follows the Vista convention.	RID values affect SID construction and profile path resolution. Assuming a fixed RID for the primary user account produces incorrect SID lookups in cross-generation investigations.
E (Prefetch)	XP and Vista use uncompressed MAM-format Prefetch files with no Amcache support. Windows 8 and later introduce compressed Prefetch and Amcache .hve. Windows 11 further enriches Amcache with executable SHA-1 hash metadata.	Compressed Prefetch requires decompression before parsing; tools reading XP-era files directly will fail silently on Windows 10/11 images. Amcache provides execution evidence unavailable in XP or Vista. The Windows 11 image yielded 503 Prefetch entries versus 36 on Charlie's XP November image, a 14× density increase.
F (browser)	XP: IE index.dat binary store and Firefox places.sqlite. Vista and later add Chrome History SQLite. Windows 11 is dominated by Chrome and Edge SQLite traces; IE is absent; Edge WebView2 cache artifacts are introduced.	Three distinct timestamp epochs persist: IE FILETIME, Firefox Unix microseconds ÷ 10 ⁶ , and Chromium timestamps relative to 1601-01-01 requiring subtraction of 11,644,473,600. Windows 11 introduces Edge WebView2 artifacts absent in earlier generations.
F/G (email)	XP: Outlook Express .dbx binary stores. Vista: Windows Mail individual .eml files. Windows 11: no native local mail store; email resides in webmail or cloud storage.	On-disk email artifacts shift completely across generations. Windows 11 email evidence is not absent but lies outside local disk scope—recovery requires Layer F browser artifact analysis or cloud acquisition, both beyond the boundary of local disk imaging.
G (deletion)	XP: RECYCLER/{SID}/Dc# with binary INF02 index. Vista and later: per-file \$I/\$R pairs under \$Recycle.Bin. Windows 11 retains \$I/\$R handling; \$I records add an explicit deletion timestamp field absent in Vista.	Extraction method differs completely across generations. XP requires binary INF02 parsing; Vista and later require per-file \$I/\$R handling. Windows 11 enables more precise deletion chronology. Mixed-OS investigations require all three parsing approaches.
H (event logs)	XP: binary .evt format, EID 528 logon and EID 538 logoff. Vista and later: XML-based .evtx, EID 4624 logon and EID 4634 logoff. Windows 11 retains .evtx and adds EID 4648 explicit credential logon.	The EID namespace was remapped at Vista and has continued to expand. XP EID 528 is not equivalent to Vista EID 4624. Queries written for one generation silently miss events on another. Parser selection and EID lookup must be OS-generation-aware.

Table 6. *Cont.*

Layer	Evolution (XP → Vista/7 → Windows 11)	Forensic Impact
I/J (partition)	XP uses sector 63 offsets with MBR partition layouts. Vista and later use sector 2048 with 4K-aligned partitions. Windows 11 enforces GPT with UEFI layouts and sector 2048 alignment; legacy MBR layouts are eliminated.	Incorrect offset causes silent mount failure. XP: <code>offset=\$((63x512))</code> ; Vista and later: <code>offset=\$((2048x512))</code> . Manual partition inspection was required to resolve extraction failures on Terry's Vista images. Automated workflows must verify partition layout before extraction.
J (journal)	XP and Vista retain both <code>\$LogFile</code> and <code>\$UsnJrnl</code> . Windows 11 retains both; <code>\$UsnJrnl</code> version 3 record format introduces an extended header, and journal volumes are substantially larger (307 MB observed).	<code>\$LogFile</code> and <code>\$UsnJrnl</code> are distinct artifacts and must not be cited interchangeably. <code>\$LogFile</code> supports crash recovery; <code>\$UsnJrnl</code> records file change reasons and supports evidentiary chronology. Windows 11 version 3 records may be truncated by parsers written for version 2. This study extracted exclusively from <code>\$UsnJrnl</code> across all systems.

Table 6 shows that artifact implementations evolve substantially across operating system generations even when their evidentiary function remains stable. This distinction is operationally important because selective acquisition workflows must accommodate generation-specific parsing requirements without abandoning functionally equivalent evidence classes.

Artifact presence is a necessary but insufficient condition for validating the HVEA model. Investigators working across heterogeneous environments must also contend with structural changes in how artifacts are implemented across operating system generations. The dataset used in this study spans Windows XP (2001), Windows Vista (2007), and Windows 11 (2021), and the examination process surfaced several consequential nomenclature and structural divergences that the HVEA model must account for.

Two observations are particularly significant. First, the partition offset divergence between Windows XP and later systems creates a silent failure point in automated workflows. In this investigation, Terry's Vista images initially returned empty extraction results until the correct partition layout was identified through manual inspection. This demonstrates that acquisition pipelines must verify partition structure before layer extraction begins.

Second, the distinction between `$LogFile` and `$UsnJrnl` is conceptually important for reporting Layer J findings. Although both artifacts were present across examined systems, only `$UsnJrnl` was used in this study for file lifecycle reconstruction. Treating the two artifacts interchangeably obscures their different forensics roles.

Despite these implementation changes, the evidentiary functions of the HVEA layers remain stable. Deletion evidence persists whether encoded through INF02 or `$I/$R`; event chronology persists whether the log format is `.evt` or `.evt×`; browser evidence persists despite changes in storage format and timestamp conventions. The model is therefore stable at the level of forensics function even as artifact structures evolve, requiring only parser-level updates rather than structural redesign.

5.3. Layer E as a Counter-Forensics Detection Surface

Layer E (execution telemetry) emerges as a critical detection surface for activity not recoverable from user content alone. Table 7 summarizes confirmed executions of forensics and counter-forensics tools extracted from Prefetch evidence. Prefetch files have been independently identified as reliable indicators of program execution in Windows forensics

practice [7], and the present investigation confirms their value as a detection surface for counter-forensics activity not recoverable from user content layers alone.

Table 7. Confirmed execution of forensics and counter-forensics tools grouped by behavioral function (Layer E Prefetch evidence).

Behavior Class	Systems/Tools Observed	Count	Interpretation
Memory Acquisition Activity			
Charlie December 11	MDD_1.3.EXE, WIN32DD1.EXE	3	Active memory acquisition indicating live forensics interaction.
Jo November 20	MDD_1.3.EXE, HELIX.EXE	3	Live response acquisition consistent with investigative activity.
Pat November 16/November 17	MDD_1.3.EXE, HELIX.EXE	8	Repeated acquisition behavior across sessions suggests deliberate monitoring or investigation.
Terry November 19	MDD_1.3.EXE	1	Acquisition activity on reimaged system indicates post-reimage analysis.
Encryption and Obfuscation			
Jo December 7	TRUECRYPT.EXE, TRUECRYPT FORMAT.EXE	2	Encryption deployment consistent with concealment or secure storage.
Custom and Unknown Execution			
Jo December 7	JO.EXE	1	Non-standard binary execution indicates user-generated or non-attributed tooling.
Steganography and Data Concealment			
Charlie December 11	ISECRETS2.EXE	2	Data hiding activity consistent with covert storage techniques.

The execution summary includes repeated use of memory acquisition utilities, encryption tools, and custom binaries across multiple subjects and dates. In several cases, these executions did not yield corresponding artifacts in Layer C. No memory dumps were recovered on the live filesystem, encryption tools suppressed volume history, and custom binaries operated in non-standard locations. In such cases, Prefetch evidence provided the only persistent indicator of activity.

Figure 4 makes this asymmetry explicit. Systems with minimal user content still exhibit substantial execution activity, reinforcing that behavioral evidence persists independently of user-generated artifacts. The Windows 11 system appears as a high-volume outlier but follows the same structural pattern, indicating that the observed asymmetry is not an artifact of scale but a consistent property of the evidentiary relationship between Layers C and E.

A representative example of cross-layer dependency is observed in the execution of steganography tools. The utility ISECRETS2.EXE was identified on Charlie's system (Layer E), while related tooling distributed via email was recovered from Layer G. The tool was received, executed, and used to embed data within image files present in Layer C. No single layer independently exposes this sequence; only cross-layer correlation reconstructs the full behavioral chain.

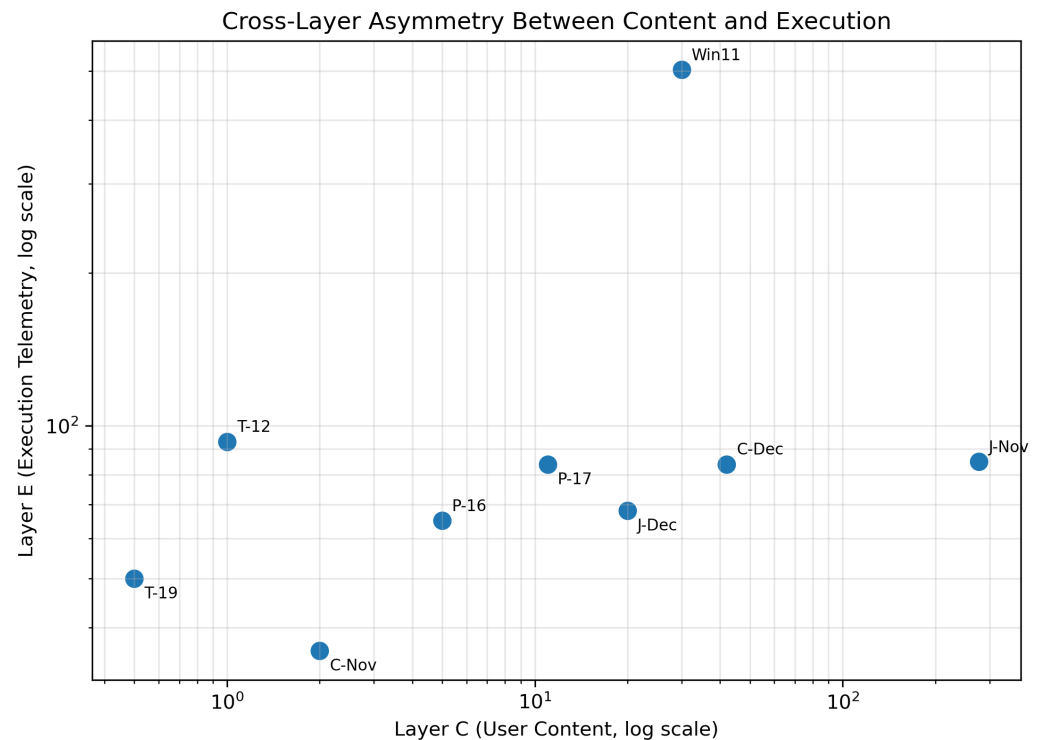


Figure 4. Cross-layer asymmetry asymmetry between Layer C (user content) and Layer E (execution telemetry). On a log–log scale, systems that fall well below the diagonal are those that produced substantial Layer E activity despite very little Layer C, the exact scenario where content-first triage breaks down. The asymmetry is directional: high Layer E with low Layer C is common and often the strongest indicator of real activity, while the reverse pattern is rare. This makes Layer E the more dependable activity signal under concealment. Even the Windows 11 system, although much larger in overall volume, follows the same trend, showing that the C–E imbalance is a structural property of the Layer C/Layer E relationship rather than an artifact of scale.

These observations demonstrate that evidentiary completeness emerges from inter-layer linkage rather than isolated artifact recovery, and that Layer E provides a reliable detection surface for transient or concealed activity.

5.4. Three-Source Corroboration and the Structural Corroboration Tier

The structural corroboration tier (Layers H–J) was used throughout this investigation to validate timestamps and activity patterns inferred from lower layers. The evidentiary value of structural corroboration rests on the independence of artifact generation mechanisms across operating system components such as authentication services, filesystems, and event logging infrastructure [29]: each artifact class records the same underlying event through a distinct OS pathway, ensuring that convergent timestamps reflect genuine system activity rather than outputs of a single logging mechanism. A recurring validation pattern emerged across multiple subjects consistent with this principle: three independent artifact sources converging on the same event timestamp with differences of fewer than thirty seconds.

This three-source corroboration pattern is a direct and empirically demonstrated consequence of the HVEA dependency structure and to the authors' knowledge, no prior artifact cataloging approach, forensic triage framework, or layered investigation methodology has explicitly identified or operationalized this convergence property. Existing approaches such as Kondapally [7] acknowledge the need for cross-artifact correlation but stop short of specifying which artifact classes should converge, across which dependency boundaries, or what convergence threshold constitutes a defensible reconstruction standard. The HVEA model provides all three: the dependency structure predicts which layers will record the

same event independently, the empirical results confirm convergence within thirty seconds across five independent subject systems and two OS generations, and this threshold is proposed here as a minimum defensibility standard for forensic event reconstruction.

Table 8 documents representative corroboration instances from the investigation, each involving a timestamp derived from Layer B (SAM), Layer D or F (network/browser), and Layer H (event log) or Layer J (journal).

Table 8. Representative three-source timestamp corroboration instances across HVEA layers. All times are UTC. Discrepancies of under thirty seconds are expected due to OS clock resolution differences across operating system components.

Subject	Layer B (SAM)	Layer D/F	Layer H/J	Discrepancy
Jo Nov-20 seizure	Last login 18:18:15Z	NIC LastWrite 18:18:10Z	EID 6005 boot 18:18:10Z	5 s
Charlie Dec-11 seizure	Last login 17:09:53Z	DHCP lease obtained 16:43:51Z	EID 6006 shutdown 17:09:53Z	0 s
Pat Nov-17 shutdown [†]	Last login 02:27:47Z	NIC LastWrite 01:43:54Z	EID 6006 shutdown 01:43:47Z	7 s
Terry Nov-19 shutdown	Last login 00:02:30Z	NIC LastWrite 00:02:24Z	EID 6006 shutdown 00:19:56Z	6 s
Win11 last session	Last login 23:44:41Z	Chrome last activity 23:44Z	EID 4634 logoff 23:44:41Z	<1 s

[†] For Pat Nov-17, Layer B reports the final login timestamp (02:27:47Z) corresponding to the subsequent boot session. The 7 s discrepancy is measured between the NIC LastWrite (Layer D) and the shutdown event (Layer H) at 01:43:47–01:43:54Z.

These instances in Table 8 demonstrate that structural layers provide independently generated evidence rather than redundant duplication. When timestamps from identity, interaction, and structural layers converge within seconds, reconstruction becomes more defensible. When divergence occurs, it introduces new investigative insight rather than contradiction, a property that distinguishes the HVEA corroboration tier from simple redundancy checking.

Layer J provided additional corroboration where event logs were sparse. In Charlie's November image, journal records confirmed installation and file activity patterns suggested by Prefetch and browser artifacts, demonstrating the value of cross-layer redundancy. This finding further illustrates that the HVEA structural corroboration tier does not merely duplicate lower-layer inferences; it provides an independent evidentiary pathway that remains valid precisely when adversarial activity has degraded content and behavioral layers.

5.5. Implications for Selective Acquisition Efficiency

The empirical results support a more precise conclusion than general advocacy for targeted collection: evidentiary yield is unevenly distributed across HVEA layers, and acquisition efficiency depends on prioritizing layers based on investigative role rather than storage volume.

Attribution layers (A–B) remain compact yet indispensable. Primary payload layers (C–D) vary significantly depending on user activity. Behavioral layers (E–G) consistently provide high evidentiary value relative to their footprint, frequently exposing activity not visible in user content.

Structural layers (H–J) exhibit the greatest volume but serve a corroborative role, supporting validation and temporal reconstruction. As shown in Figure 5, this tier dominates artifact counts but does not proportionally dominate investigative value. Recent empirical evaluations of privacy-aware selective imaging approaches have demonstrated up to 40% reduction in non-relevant data exposure while retaining 95% of evidentially relevant artifacts [25], supporting the viability of dependency-ordered acquisition strategies of the kind proposed by the HVEA framework.

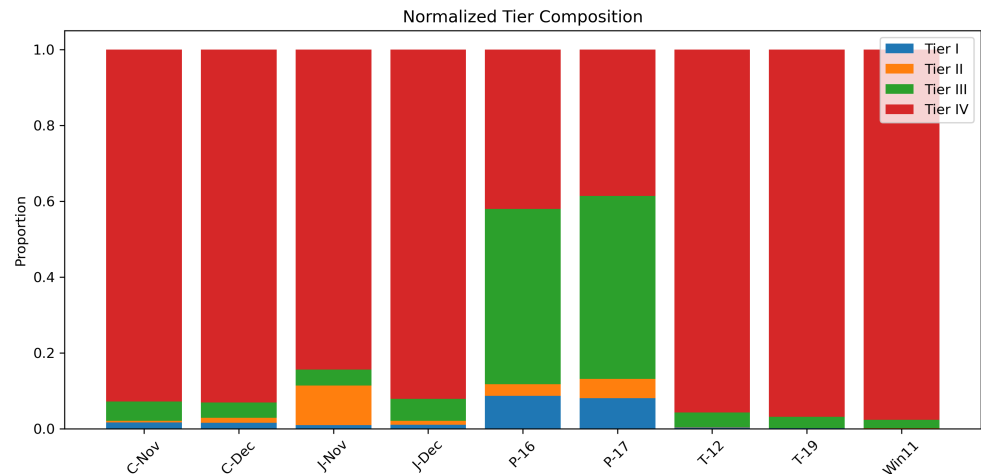


Figure 5. Normalized tier composition across systems. Each bar gives the proportional contribution of Tiers I–IV to total artifact volume per system. The figure exposes an inversion between volume and investigative value: Tier IV (structural corroboration) dominates raw counts on every system yet serves only a validating role, while Tiers I and III, the attribution anchor and the behavioral evidence that carry disproportionate investigative weight, occupy a small fraction of the volume. This inversion is the empirical basis for staged acquisition: the tiers cheapest to collect are not the least valuable, and the tier most expensive in volume is acquired last, for corroboration rather than discovery.

Taken together, these findings support a staged acquisition strategy. Layers A through G can recover the majority of investigatively significant evidence at relatively low cost, while Layers H through J can be acquired selectively to reinforce evidentiary defensibility.

To quantify these efficiency implications directly, we measured the storage footprint of HVEA-guided selective acquisition against full bit-wise imaging on the controlled Windows 11 image. The full source is a 40 GB virtual disk (83,886,080 sectors); the selective footprint is the byte sum of the artifact set targeted by each HVEA tier, measured on the read-only mounted volume. Image integrity was verified at acquisition (FTK Imager, MD5 cbff96d8... , SHA-1 e630f459... , both confirmed); the Windows partition was read at GPT slot 6, sector 444416. NTFS metafile sizes (\$MFT, \$LogFile, \$UsnJrnl:\$J) were read directly from the mounted volume, and the Layer C payload figure was cross-checked against the independent artifact inventory for this case. Table 9 reports the result.

Table 9. Storage cost–benefit of HVEA selective acquisition versus full bit-wise imaging on the controlled Windows 11 image. Reduction is relative to the 40 GB full image; ratio is full ÷ selective.

Acquisition	Footprint	Reduction	Ratio
Full bit-wise image	40.00 GB	–	1.0×
Selective, Tiers I–III	2.03 GB	94.9%	19.8×
Selective, Tiers I–IV	2.62 GB	93.5%	15.3×

Acquiring the attribution, payload, and behavioral tiers (I–III) brought the footprint down from 40 GB to 2.03 GB, a 94.9% reduction, and adding the full structural corroboration tier (I–IV) still produced a 93.5% reduction. The difference between the two rows is telling: the entire structural tier filesystem metadata, the \$LogFile, and the 417 MB change journal adds only 0.59 GB. Even when captured exhaustively, structural corroboration remains an order of magnitude smaller than full imaging.

The more important question is where that remaining data actually sits. Of the 2.62 GB selective footprint, 1.68 GB (64%) is simply the Layer C user payload, material that any acquisition approach would need to collect. The full HVEA dependency context that gives that payload meaning, the Tier I attribution anchors, the Tier III behavioral telemetry, and

the Tier IV structural corroboration, including all NTFS metafiles, adds up to just 917 MB, or 2.24% of the full image. In other words, the interpretive scaffolding that separates HVEA from content-only selective imaging costs roughly one-fortieth of the disk. This directly addresses the main practical concern about dependency-aware acquisition: keeping the artifacts needed for attribution, behavioral reconstruction, and cross-tier corroboration does not recreate the volume problem that selective imaging is meant to solve.

These savings line up with, and on a single controlled image, put concrete numbers to, the order-of-magnitude reductions previously reported for profile-based selective imaging [14] and the privacy-aware reduction figures noted above [25]. The acquisition time benefit follows the same proportional logic: the full 40 GB image required 3 min 8 s to acquire and verify under the conditions logged for this case, while a selective set roughly one-twentieth that size naturally incurs a much smaller acquisition and downstream processing load.

6. NIJ to HVEA Operational Mapping

6.1. Mapping Rationale

The National Institute of Justice (NIJ) organizes digital evidence into investigative categories such as user-generated content, communication records, system artifacts, and web activity [11]. While useful for investigative orientation, this taxonomy does not prescribe acquisition order or account for the interpretive dependencies among artifact classes. The empirical findings in Section 5 illustrate the consequence: Pat’s machine produced only five user files across two image captures, a Layer C density suggesting low investigative priority, yet Layer E yielded 65–84 Prefetch entries and four confirmed memory acquisition tool executions. A content-first triage aligned to NIJ categories alone would have missed the most forensically significant findings on that machine.

The HVEA model addresses this by translating NIJ categories into dependency-ordered acquisition priorities, aligning investigative objectives with the evidentiary dependency structure introduced in Section 3. The mapping preserves NIJ investigative intent while imposing the dependency constraints that make selective acquisition defensible.

6.2. Priority Matrix

Each NIJ crime category is mapped to the HVEA layers using a three-level scheme: High (H) for early-stage acquisition, Medium (M) for contextual reinforcement, and Low (L) for selective structural validation. Prioritization reflects evidentiary dependency order rather than artifact frequency or investigative prominence. Lower priority does not indicate evidentiary insignificance, as it reflects recommended acquisition sequence within staged forensic workflows. The color coding (green = High, amber = Medium) is intended to assist practitioners in quickly identifying which layers require immediate attention for each investigation type (see Table 10).

Table 10. NIJ crime category to HVEA layer prioritization matrix grouped by evidentiary dependency tiers. H = High, M = Medium. Green = High priority; Amber = Medium priority.

NIJ Category	Tier I		Tier II		Tier III			Tier IV		
	A	B	C	D	E	F	G	H	I	J
Crimes Against Persons	H	H	H	M	M	M	M	H	H	H
Financial Crimes	M	H	H	H	M	H	M	H	H	H
Cybercrime	M	M	M	H	H	H	M	H	H	H
Drug Offenses	H	H	M	M	M	M	M	H	H	H
Property Crimes	H	H	H	M	M	M	H	H	H	H

Tier I (A–B): Attribution anchors. Tier II (C–D): Primary evidentiary payload. Tier III (E–G): Behavioral contextualization. Tier IV (H–J): Structural corroboration.

6.3. Interpretation of Layer Prioritization Patterns

Three structural patterns emerge from the matrix, each grounded in the evidentiary dependency relationships of the HVEA model rather than investigative convention.

Attribution layers are universal prerequisites. Layers A and B appear as High or Medium priority across all five categories because no higher-layer artifact can be reliably attributed to a specific actor without them. This is a logical constraint, not a preference: the SAM-derived identity anchors in Layer B grounded every corroboration instance in Table 8, and Layer A established the system baseline against which all artifact timestamps were interpreted.

Primary payload emphasis varies by hypothesis. Layer C dominates content-focused investigations (crimes against persons, property crimes), Layer D dominates network-focused investigations (cybercrime), and both are required for financial crimes where provenance reconstruction demands document and communication artifacts together. This variation confirms that the HVEA framework correctly adapts its Tier II emphasis to the investigative hypothesis while holding Tiers I and IV constant.

Structural corroboration is architecturally constant. Layers H through J receive consistently High prioritization across all categories because they are generated by system subsystems independently of user intent [29], making them resistant to the same concealment strategies that degrade Layers C through G. The three-source convergence documents the following in Table 8: timestamps from Layers B, D/F, and H/J agreeing within thirty seconds across five independent instances demonstrates empirically that independent generation mechanisms produce corroborative evidence that deliberate artifact manipulation cannot simultaneously suppress across all three sources [21].

The matrix encodes a dependency-ordered acquisition sequence with a constant foundational workflow (Tiers I and IV) and a variable primary payload emphasis (Tier II) determined by the investigative hypothesis. Tier III provides the behavioral bridge between them in all cases. This sequencing operationalizes the HVEA dependency structure as actionable acquisition guidance across diverse investigative contexts.

7. Discussion

The empirical findings presented in Section 5 support a more precise claim than general advocacy for selective acquisition. The HVEA layers are not merely observable across independent systems; they surface in densities that reflect user role, machine history, and investigative significance, and this predictability can guide acquisition decisions before full-disk imaging begins.

The most operationally consequential finding is the persistent asymmetry between Layer C and Layer E. Machines with sparse user-generated content still produced substantial execution telemetry. Pat's machine yielded only five user files yet recorded 65–84 Prefetch entries and four confirmed memory acquisition tool executions. This asymmetry means that triage based on Layer C density alone risks misclassifying high-activity machines as low-priority targets. The counter-forensics tool distribution in Table 7 reinforces this directly: the subjects running the most memory acquisition tools were those with the least recoverable content, consistent with deliberate concealment rather than absence of activity.

The nomenclature evolution documented in Table 6 reveals a structural property not explicitly designed into the model: the HVEA framework is stable at the level of forensics function even as artifact implementations change across platform generations. Deletion evidence persists whether encoded through INF02 or \$I/\$R; event chronology persists whether the log format is .evt or .evtx. Acquisition pipelines built around HVEA layer definitions rather than specific artifact names will remain valid across OS generations, requiring only parser-level updates rather than structural redesign.

The three-source corroboration pattern in Table 8 emerged consistently across subjects and OS generations without being designed into the investigation protocol, reflecting the HVEA dependency structure: because Layers B, D/F, and H/J record the same events through independent OS mechanisms, their agreement constitutes genuine corroboration rather than redundant duplication [29]. This suggests a minimum defensibility standard for the event reconstruction: any timestamp used as an evidentiary anchor should be independently confirmed by at least two artifact sources from different HVEA tiers.

7.1. Positioning HVEA Against Existing Layered Models

The findings above also clarify how the HVEA model differs from prior layered and prioritization approaches, not in the artifacts it examines but in the relationships it imposes among them. Existing models prioritize artifacts along three lines, each of which the present results extend rather than repeat. Sequencing models such as the stepwise methodology of Lee et al. [5] and the Windows artifact prioritization of Ahmadi et al. [6] establish which artifacts to examine first, but order them by acquisition convenience or analytical speed rather than by evidentiary dependency; neither specifies what must accompany a high-priority artifact for it to be interpretable. The Layer C/Layer E asymmetry observed here is precisely the case such ordering mishandles: a convenience-ordered triage that keys on user-content density would deprioritize exactly the machines that carry the richest execution telemetry.

Abstraction models such as the four-level event organization of Bhandari and Jusas [20] demonstrate that progressive structuring improves timeline interpretability. Still, the structure is applied after acquisition to events already collected. HVEA relocates that structuring to the acquisition decision itself: the four tiers determine what is collected and in what order, so that interpretive context is preserved rather than reconstructed. The cross-generation stability documented in Table 6 is a direct consequence; because the tiers are defined by evidentiary function rather than by artifact name or event format, the ordering survives the platform changes that an implementation-level abstraction would not.

Cataloguing approaches come closest. Kondapally's survey of forensically significant Windows artifacts [7] observes that no single artifact is decisive and that reliability emerges through cross-correlation, but it stops at the observation, leaving the dependency relationships unspecified. The three-source corroboration pattern reported in Table 8 supplies the missing structure empirically: it shows not merely that corroboration matters but which tiers corroborate one another (Layers B, D/F, and H/J recording the same events through independent OS mechanisms) and to what threshold (independent confirmation from at least two tiers). This is the operational content that distinguishes a dependency hierarchy from a catalogue.

Across all three comparisons, the distinction is the same. Prior models answer which artifacts are valuable; HVEA answers which artifacts are valuable together. The dependency hierarchy is therefore not a re-labeling of existing prioritization schemes but an additional axis they do not encode, which is the axis that converts a prioritized list of artifacts into an acquisition order, while preserving interpretability.

7.2. Limitations and Future Work

Windows-only evaluation: The current empirical evaluation is limited to Windows XP, Vista, and Windows 11 environments. The evidentiary dependency hierarchy at the tier level is expected to generalize across platforms: macOS, Linux, Android, and iOS systems generate analogous artifact classes serving attributional, payload, behavioral, and corroborative roles, but the specific artifact implementations at each layer differ substantially. Extension to non-Windows environments requires re-evaluation of layer definitions and artifact mappings before the HVEA framework can be applied in cross-platform investigations.

Planned work: Develop a parallel layer-to-artifact mapping for macOS covering unified logs, property lists, and APFS metadata, and validate it against a controlled macOS image using the same deterministic descriptor-mapping procedure used here. The proportion of tiers successfully reproduced will serve as the measure of cross-platform stability. Once the macOS translation is confirmed, the same process will be extended to Linux and mobile ecosystems.

Synthetic and controlled datasets: The M57-Patents dataset is a synthetic scenario rather than a live criminal case, and the Windows 11 experiment involved deliberate rather than naturalistic activity. Both constraints mean the artifact distributions observed may not fully reflect the variability of real-world investigations, including adversarial behaviors not present in the M57 scenario. Future work should validate the HVEA model against real case data, subject to appropriate legal and ethical constraints.

Planned work: Repeat the layer-density and three-source corroboration analyses on de-identified real-case Windows images obtained under an institutional data-sharing and ethics agreement. The goal is to compare tier composition and corroboration-convergence behavior against the controlled results reported here, testing whether the sub-thirty-second convergence threshold persists in naturalistic data. If the pattern holds, it strengthens the claim that the convergence property is not an artifact of controlled imaging but a feature of real-world Windows activity traces.

Expert-derived NIJ prioritization matrix: The NIJ prioritization matrix represents expert-derived judgments rather than empirically optimized acquisition sequences. Future validation against real investigative outcomes would substantially strengthen its operational guidance value.

Planned work: Once real-case data (both synthetic and controlled datasets, above) becomes available, measure the evidentiary yield of each Layer across different crime categories. These results will be used to test and, where necessary, revise the expert-assigned priority weights, shifting the matrix from guidance-driven to empirically calibrated.

Cloud and mobile artifacts: The Windows 11 experimental image was acquired as a local disk image only. Email evidence in modern Windows environments increasingly resides in browser-cached webmail artifacts (Layer F) or cloud storage outside the local filesystem rather than in a native mail store, placing full email content recovery beyond the scope of the local disk acquisition and outside the boundaries of the present study. Extension of the HVEA framework to accommodate cloud acquisition and mobile artifact ecosystems is identified as a priority for future work.

Planned work: Extend the Layer D/Layer F definitions to include cloud-residency indicators such as sync-client databases, OAuth token stores, and browser-cached webmail fragments and evaluate, on a controlled account, how much cloud-resident evidence can be reached from local Windows artifacts alone. The goal is to identify where the HVEA local-acquisition boundary actually lies, and to determine what a cloud-aware extension must recover beyond that boundary.

Automated acquisition pipeline integration: The current study demonstrates the conceptual and empirical basis for HVEA-guided acquisition but does not implement or evaluate an automated acquisition pipeline. Integration of the HVEA dependency hierarchy into automated triage systems, including formal specification of HVEA-compatible acquisition profiles for platforms such as KAPE and Autopsy, represents a concrete near-term research direction with direct practitioner impact.

Planned work: Encode the deterministic layer-to-artifact mapping as HVEA-compatible acquisition profiles for an established triage tool, for example, a KAPE Target/Module set with an equivalent Autopsy ingest pipeline and benchmark automated tiered acquisition against manual examination on the same images. Evaluation will report acquisition

time, storage footprint, and per-tier artifact-miss rate. The dependency-check escalation described in Section 4.4 would serve as the profile's stop-condition guard.

8. Conclusions

This paper introduced the High-Value Evidence Area (HVEA) pyramid, a dependency-oriented abstraction model organizing digital forensics artifacts into ten operational layers across four evidentiary tiers. The HVEA model makes a substantive and novel contribution to the field by providing the first formal evidentiary dependency structure for digital forensics artifact acquisition, explicitly encoding the interpretive relationships among artifact classes that prior cataloging, triage, and selective imaging approaches have left implicit.

Empirical evaluation against eight M57-Patents disk images and a controlled Windows 11 experiment confirmed that all ten layers produce extractable artifacts across independent systems, that execution telemetry exposes investigatively significant activity absent from content layers, and that cross-layer timestamp corroboration within thirty seconds is achievable across OS generations as a defensible reconstruction standard.

The investigation further documented seven consequential artifact nomenclature changes across Windows XP, Vista, and Windows 11 including partition offset divergence, causing silent extraction failures and EID namespace remapping between generations, confirming that the model is functionally stable while requiring generation-specific extraction procedures at the operational level.

For practitioners, the findings support a staged acquisition strategy in which Layers A–G recover the majority of investigatively significant evidence at substantially lower cost than full-disk imaging, with structural corroboration layers acquired selectively to reinforce evidentiary defensibility. For researchers, the HVEA framework provides a formal dependency structure against which artifact prioritization approaches and automated triage systems can be evaluated. Future work should address extension to non-Windows environments, empirical validation of the NIJ prioritization matrix against real investigative outcomes, and integration of the dependency hierarchy into automated acquisition pipeline design.

Author Contributions: Conceptualization, O.O.A., C.V. and N.S.; methodology, O.O.A.; formal analysis, C.V.; original draft preparation, O.O.A.; review and editing, C.V. and N.S.; visualization, O.O.A.; supervision, C.V. and N.S.; project administration, C.V. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The legacy forensics dataset used in this study is publicly available through the Digital Corpora repository as part of the M57-Patents scenario dataset. Controlled experimental artifacts generated within the Windows 11 virtual machine environment are available from the corresponding author upon reasonable request.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Quick, D.; Choo, K.K.R. IoT device forensics and data reduction. *IEEE Access* **2018**, *6*, 47566–47574.
2. Freiling, F.; Groß, T.; Latzo, T.; Müller, T.; Palutke, R. Advances in forensic data acquisition. *IEEE Des. Test* **2018**, *35*, 63–74.
3. Stüttgen, J.; Dewald, A.; Freiling, F.C. Selective imaging revisited. In *Proceedings of the 2013 Seventh International Conference on IT Security Incident Management and IT Forensics*; IEEE: Piscataway, NJ, USA, 2013; pp. 45–58.

4. Du, X.; Ledwith, P.; Scanlon, M. Deduplicated disk image evidence acquisition and forensically-sound reconstruction. In *Proceedings of the 2018 17th IEEE International Conference on Trust, Security and Privacy in Computing and Communications/12th IEEE International Conference on Big Data Science and Engineering (TrustCom/BigDataSE)*; IEEE: Piscataway, NJ, USA, 2018; pp. 1674–1679.
5. Lee, S.; Bang, J.; Lim, K.; Kim, J.; Lee, S. A stepwise methodology for tracing computer usage. In *Proceedings of the 2009 Fifth International Joint Conference on INC, IMS and IDC*; IEEE: Piscataway, NJ, USA, 2009; pp. 1852–1857.
6. Ahmadi, H.R.; Mourad, A.; Tawil, R.; Awada, M.B. A new approach in Digital forensics investigation process. In *Proceedings of the 2018 International Conference on Computer and Applications (ICCA)*; IEEE: Piscataway, NJ, USA, 2018; pp. 1–275.
7. Kondapally, B.P. *Forensically Important Artifacts in Windows Operating Systems*; Academia: San Francisco, CA, USA, 2015.
8. McKeown, S.; Russell, G.; Leimich, P. Fast forensic triage using centralised thumbnail caches on Windows operating systems. *J. Digit. Forensics Secur. Law* **2019**, *14*, 1.
9. Garfinkel, S. *Digital Forensic Scenario Dataset Containing Disk Images, RAM Captures, and Network Traffic Simulating Corporate Intellectual Property Theft Activity*; M57-Patents Scenario Dataset; Digital Corpora, Naval Postgraduate School: Monterey, CA, USA, 2009.
10. Barocas, S.; Selbst, A.D. Big data's disparate impact. *Calif. Law Rev.* **2016**, *104*, 671.
11. Guide, N. *A Guide for First Responders*; National Institute of Justice: Washington, DC, USA, 2001; Volume 4.
12. Al-Dhaqm, A.; Ikuesan, R.A.; Kebande, V.R.; Abd Razak, S.; Grispos, G.; Choo, K.K.R.; Al-Rimy, B.A.S.; Alsewari, A.A. Digital forensics subdomains: The state of the art and future directions. *IEEE Access* **2021**, *9*, 152476–152502.
13. Pedapudi, S.M.; Vadlamani, N. Data acquisition based seizure record framework for digital forensics investigations. In *Proceedings of the 2021 5th International Conference on Electronics, Communication and Aerospace Technology (ICECA)*; IEEE: Piscataway, NJ, USA, 2021; pp. 1766–1768.
14. Grier, J.; Richard, G.G., III. Rapid forensic imaging of large disks with sifting collectors. *Digit. Investig.* **2015**, *14*, S34–S44.
15. Roussev, V.; Quates, C.; Martell, R. Real-time digital forensics and triage. *Digit. Investig.* **2013**, *10*, 158–167.
16. Dangi, S.; Ghanshala, K.; Sharma, S. Approaches to Selective Imaging of Live Systems via Memory Forensics. In *Proceedings of the 2023 3rd International Conference on Intelligent Technologies (CONIT)*; IEEE: Piscataway, NJ, USA, 2023; pp. 1–5.
17. Hansen, R.A.; Seigfried-Spellar, K.; Lee, S.; Chowdhury, S.; Abraham, N.; Springer, J.; Yang, B.; Rogers, M. File toolkit for selective analysis & reconstruction (filetsar) for large-scale networks. In *Proceedings of the 2018 IEEE International Conference on Big Data (Big Data)*; IEEE: Piscataway, NJ, USA, 2018; pp. 3059–3065.
18. Song, J.; Li, J. A framework for digital forensic investigation of big data. In *Proceedings of the 2020 3rd International Conference on Artificial Intelligence and Big Data (ICAIBD)*; IEEE: Piscataway, NJ, USA, 2020; pp. 96–100.
19. Quick, D.; Choo, K.K.R. Impacts of increasing volume of digital forensic data: A survey and future research challenges. *Digit. Investig.* **2014**, *11*, 273–294.
20. Bhandari, S.; Jusas, V. The phases based approach for regeneration of timeline in digital forensics. In *Proceedings of the 2020 International Conference on INnovations in Intelligent SysTems and Applications (INISTA)*; IEEE: Piscataway, NJ, USA, 2020; pp. 1–6.
21. Vanini, C.; Hargreaves, C.; Breitingner, F. Evaluating tamper resistance of digital forensic artifacts during event reconstruction. *Digit. Threat. Res. Pract.* **2025**, *6*, 29.
22. Vanini, C.; Gruber, J.; Hargreaves, C.; Benenson, Z.; Freiling, F.; Breitingner, F. Understanding strategies and challenges of timestamp tampering for improved digital forensic event reconstruction. In *Proceedings of the Digital Forensics Doctoral Symposium, Brno, Czech Republic, 1 April 2025*; pp. 1–8.
23. Weyermann, C.; Vanini, C.; Souvignnet, T.R. A common framework to situate digital and physical traces in time. *Forensic Sci. Int.* **2024**, *360*, 112020.
24. Shayau, Y.H.; Asmawi, A.; Rum, S.N.M.; Ariffin, N.A.M. Digital forensics investigation reduction model (DIFReM) framework for Windows 10 OS. In *Proceedings of the 2019 IEEE 9th International Conference on System Engineering and Technology (ICSET)*; IEEE: Piscataway, NJ, USA, 2019; pp. 459–464.
25. Karadsheh, L.; Al-Najjar, H.; Nehari-Talet, A. Privacy-Preserving Digital Forensics: Balancing Investigation Needs and User Rights. In *Proceedings of the 2025 1st International Conference on Computational Intelligence Approaches and Applications (ICCIAA)*; IEEE: Piscataway, NJ, USA, 2025; pp. 1–7.
26. Zimmerman, E. *Kroll Artifact Parser and Extractor-KAPE*; Kroll: New York, NY, USA, 2020.
27. Michelet, G.; Breitingner, F. Automation for digital forensics: Towards a classification model for the community. In *Proceedings of the Digital Forensics Doctoral Symposium, Brno, Czech Republic, 1 April 2025*; pp. 1–8.
28. Voigt, L.L.; Freiling, F.; Hargreaves, C. A metrics-based look at disk images: Insights and applications. *Forensic Sci. Int. Digit. Investig.* **2025**, *52*, 301874.
29. Beisbart, C. Do Computer Simulations Include Digital Artifacts? *Metaphysics* **2024**, *7*, 37–50.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.