A Repeatability- and Error-Centered Validation Protocol for Mobile Device Forensic Workflows

Younghun Chae[1,*], Angela Guercio[1] and Timothy Arndt[2]

[1]Department of Computer Science, Kent State University at Stark, 6000 Frank Ave NW, North Canton, OH 44720, USA

[2]Department of Information Systems, Cleveland State University, 2121 Euclid Avenue, Cleveland, OH 44115-2214, USA

Abstract: Mobile device forensic reliability is highly dependent on device model, operating-system version, access state, extraction pathway, parser behavior, application version, and other workflow conditions. As a result, reliability should be demonstrated under defined and repeatable circumstances rather than inferred from tool identity alone. This paper presents a repeatability- and error-centered validation protocol for mobile device forensic workflows and illustrates its application through a bounded Android logical-acquisition pilot. The study is motivated by the growing need for practical validation methods that address configuration-specific evidence access, parser variability, frequent mobile application changes, cloud-linked artifacts, AI-assisted parsing workflows, and timestamp uncertainty in contemporary iOS and Android examinations. The proposed protocol adopts a controlled laboratory design based on documented ground truth, repeated acquisitions, fixed workflow conditions, explicit treatment of access-state constraints, and stage-based anomaly classification. It defines a compact measurement framework for artifact recovery, false positives, false negatives, timestamp deviation after justified normalization, repeatability across repeated runs, and reproducibility across independently repeated subsets. To demonstrate the reporting model, the paper includes a small scoped pilot using an unlocked Samsung Galaxy S8 and an ADB logical-acquisition workflow. Across three repeated runs, the pilot recovered all 35 scoped user-accessible artifacts and produced identical recovered artifact sets, while also showing the importance of bounding claims because timestamp-delta accuracy, app-private data, deleted artifacts, and cross-device generalization were outside the pilot scope. The main result of the paper is a disclosure-ready validation framework that links measured outcomes to the exact technical conditions under which they were produced. Rather than ranking commercial tools, the paper offers a reproducible framework for generating bounded, reviewable, and defensible validation records for mobile forensic workflows. The pilot is used only to demonstrate recovery and repeatability reporting; timestamp-delta validation remains part of the broader protocol and requires trusted creation-time records in a full deployment.

Keywords: Mobile Device Forensics, Forensic Workflow Validation, Repeatability, Reproducibility, Timestamp Reliability.

1. INTRODUCTION

Mobile devices are high-value sources of digital evidence because they preserve communications, location traces, authentication material, and application activity that can help reconstruct user behavior. Their forensic examination remains difficult because modern mobile operating systems are intentionally designed to restrict offline access and to bind decryption to hardware protections, device state, and user authentication. On iOS, evidence availability is shaped by Data Protection classes and keychain protections, so file access may differ depending on whether the device is locked, unlocked, or only recently unlocked [1] . On Android, file-based encryption separates data into device-encrypted and credential-encrypted domains, which affects artifact availability before first unlock and after first unlock, and complicates comparisons across tools and acquisition conditions [2,3].

*Address correspondence to this author at the Department of Computer Science, Kent State University at Stark, 6000 Frank Ave NW, North Canton, OH 44720, USA; E-mail: ychae@kent.edu

Comparative acquisition studies have likewise shown that different tools may recover different subsets of data from the same smartphone, which reinforces the need for explicit validation rather than assumption [4].

1.1. Defining Reliability

Because mobile forensic output depends on device model, operating-system version, extraction pathway, application version, encryption state, and other testing conditions, validation cannot be treated as a one-time event. Recent work argues that reliability must be demonstrated and maintained across the lifecycle of tools and methods, while newer research on automated reference-data generation shows that application updates alone may justify repeated revalidation [5,6]. This view is consistent with earlier mobile-forensics research calling for structured tool testing, measurable performance comparison, and clearer definitions of tool support and limitations [7–9].

In this setting, reliability should be evaluated as a configuration-specific property of a forensic workflow rather than assumed from vendor identity alone. In

this paper, reliability refers to the extent to which a mobile forensic workflow, under explicitly defined technical and access conditions, produces materially correct, repeatable, and interpretable outputs with bounded error. Accordingly, reliability is not treated as a general property of a tool brand, but as a configuration-specific property of the tested workflow.

These validation concerns are directly tied to evidentiary reliability. Contemporary forensic-science scholarship frames reliability review around empirical testing, error rates, standards, peer review, and the reliable application of methods, which closely aligns with the practical demands of digital forensic evidence [10]. In the digital-evidence context, legal and forensic scholarship has also emphasized that insufficient reliability testing can undermine fairness and the presumption of innocence [11]. This concern is especially important in mobile forensics because timestamp interpretation is often fragile. Clock skew, drift, timezone offsets, daylight-saving transitions, and inconsistent artifact provenance can distort event reconstruction if they are not explicitly examined [12–14].

1.2. Related Work and Gap

Prior work has established the value of structured, measurement-based approaches to mobile-forensic tool evaluation. Baggili *et al.* proposed a database-driven approach to tool testing and error estimation in mobile forensics [7]. Saleem *et al.* introduced a quantitative method for comparing mobile forensic tools [8], and Anobah *et al.* proposed a multi-level testing framework spanning basic tasks through anti-forensic scenarios [9]. More recent work has expanded this discussion into broader validation frameworks and explicit mappings of digital-forensic error sources [6,12]. However, the field still needs a compact and practical protocol that integrates controlled ground truth, repeated acquisitions, error-focused measurement, and reporting guidance in a form that can be reproduced and reviewed under defined laboratory conditions. To reduce the gap between protocol design and practical application, this paper also includes a bounded pilot demonstration. The pilot shows how a controlled ground-truth subset can be acquired repeatedly, scored, and reported in a form that preserves the limits of the tested device, acquisition pathway, artifact categories, and access state.

Table 1: Comparative Positioning Against Prior Validation Frameworks and Tool-Testing Studies

Prior work or framework	Main focus	Remaining gap	Position of the proposed protocol
Baggili *et al.* mobile phone forensic tool testing (7)	Database-driven testing and error estimation for mobile forensic tools	Provides structured tool-testing logic, but does not fully integrate repeated acquisition, timestamp-error reporting, stage-based anomaly classification, and disclosure-ready reporting	Extends tool-testing logic into a repeatability- and error-centered validation record suitable for laboratory review and later evidentiary disclosure
Saleem *et al.* quantitative tool comparison (8)	Quantitative comparison of mobile forensic tools across selected outputs	Emphasizes tool comparison, but less directly addresses configuration-specific workflow validation under bounded device, access-state, and extraction conditions	Uses quantitative metrics while avoiding product ranking; results are interpreted only under the tested device, toolchain, access state, extraction pathway, and artifact scope
Anobah *et al.* mobile device forensic testing framework (9)	Multi-level testing framework for mobile forensic tools, including basic and advanced scenarios	Provides a useful testing structure, but gives less emphasis to timestamp-error summaries, repeated-run stability, and stage-specific anomaly reporting	Adds explicit artifact recovery, FP/FN, timestamp-error, Jaccard repeatability, reproducibility, and anomaly-stage reporting
Reliability Validation Enabling Framework (RVEF) (6)	General digital-forensic reliability validation across technology, method, and application levels	Provides a broad reliability framework, but is not specific to mobile workflow variables such as access state, application-version drift, cloud-linked artifacts, parser behavior, and mobile extraction pathways	Operationalizes reliability validation for mobile forensic workflows by specifying device context, access state, extraction mode, parser behavior, application version, synchronization state, and reporting limits
Digital-forensic error-source research (12)	Identification and classification of error sources across the digital forensic process	Clarifies where errors may arise, but does not by itself provide a mobile-specific measurement and reporting template	Incorporates stage-based anomaly classification into the validation record so that errors are localized to connectivity, acquisition, access/decryption, parsing, presentation, timestamp conversion, duplication/merge, or reproducibility stages
Proposed protocol	Repeatability- and error-centered validation for mobile forensic workflows	Addresses the need for a compact, practical, and reviewable protocol that combines controlled ground truth, repeated acquisition, error metrics, timestamp analysis, anomaly classification, and bounded reporting	Provides a disclosure-ready framework and a bounded Android logical-acquisition pilot showing how measured validation outputs can be reported without overstating tool performance

This comparison in Table **1** clarifies that the proposed protocol is not intended to replace prior validation frameworks or tool-testing studies. Instead, it translates their shared emphasis on empirical testing, repeatability, reproducibility, and error disclosure into a mobile-specific workflow that can be applied under documented device, access-state, extraction, parser, and reporting conditions.

1.3. Contemporary Mobile Forensic Workflow Challenges

Contemporary mobile forensic workflows face additional validation pressure from application parser drift, frequent mobile application updates, cloud-linked mobile artifacts, and AI-assisted parsing workflows. Parser drift occurs when an application changes its database schema, file structure, metadata conventions, or synchronization behavior while forensic tools continue to rely on earlier parsing assumptions. This concern is consistent with recent work showing that mobile application updates can trigger renewed reference-data generation and revalidation needs [5], as well as app-specific studies showing that mobile artifacts are highly dependent on application design and version behavior [15].

Cloud-linked mobile artifacts further complicate validation because local device records may represent cached, synchronized, deleted, or partially available data rather than a complete source of truth. AI-assisted parsing workflows add another layer of uncertainty because automated classification, entity extraction, or artifact grouping may improve triage efficiency while also creating risks of opaque interpretation, unstable labeling, or insufficiently explained outputs. These issues align with broader digital-forensic error research that emphasizes stage-specific error sources and systematic tool-error mitigation [12,16]. Accordingly, validation records should identify the exact app version, parser version, synchronization state, AI-assisted processing settings when used, and tool configuration under which forensic conclusions were produced.

1.4. Study Objective and Contributions

This paper addresses the need for a practical, repeatable, and disclosure-ready validation approach for mobile device forensic workflows. The objective is to present a protocol that allows laboratories and researchers to measure workflow reliability under explicitly defined technical conditions, while also demonstrating how the protocol can be applied to a small controlled validation scenario. This paper provides a structured method for generating bounded reliability statements that identify what was tested, what was recovered, what was missed, what kinds of errors or uncertainties were observed, and under which conditions the results should be interpreted.

The contributions of the paper are fourfold. First, it proposes a practical end-to-end workflow for configuration-specific mobile forensic validation using controlled ground truth, fixed acquisition conditions, repeated runs, and explicit access-state documentation. Second, it defines a compact measurement framework for artifact recovery, false positives, false negatives, timestamp deviation, repeatability, reproducibility, and stage-based anomaly classification. Third, it provides a disclosure-ready reporting structure that records device state, operating-system version, extraction pathway, toolchain components, parser settings, time-handling assumptions, anomaly categories, and interpretive limits. Fourth, it includes a bounded Android logical-acquisition pilot to demonstrate how the protocol can be applied in practice. The pilot is intentionally narrow, but it shows how scoped ground truth, repeated acquisition, category-level recovery, and Jaccard-based repeatability can be reported without overstating the findings beyond the tested device, artifact categories, acquisition pathway, and access conditions. These contributions position the paper as both a validation framework and a case-based demonstration of measured forensic reporting. The proposed approach is designed to support laboratory quality assurance, technical peer review, and later evidentiary scrutiny by replacing generalized claims of product-level reliability with measured, configuration-specific, and reviewable validation records.

2. METHODS

2.1. Validation Design and Scope

This study adopts a controlled laboratory protocol to validate mobile device forensic workflows under defined, repeatable conditions. The protocol assumes lawful authority, institutional approval, and policy-compliant handling of test devices. It is intended for laboratory validation of forensic acquisition and interpretation workflows, not for destructive hardware attacks, unauthorized access, or claims about universal tool capability. Because

modern mobile operating systems bind data access to encryption state, device state, and user authentication, extraction outcomes must be interpreted relative to the tested device, operating-system version, access state, and extraction pathway rather than as absolute indicators of tool quality [1–3,6,7].

The protocol is tool-agnostic in design, but each validation cycle should instantiate multiple independent toolchains so that differences in acquisition, parsing, normalization, and reporting can be observed directly. Prior work has shown that mobile forensic output can vary substantially across tools, recovery methods, device types, and software versions, and that such variation should be treated as an empirical property of the workflow rather than as an exceptional case [4–9]. Accordingly, all validation outcomes produced by this protocol are configuration-specific and version-specific. They apply only to the tested devices, tool versions, operating-system states, and documented settings used during the validation runs [5,6].

The objective of the protocol is not to produce a definitive ranking of commercial tools, but to generate a reproducible validation record that measure's reliability under stated conditions. In this paper, reliability is operationalized through artifact recovery, false positives, false negatives, timestamp deviation after justified normalization, repeatability across repeated runs, reproducibility across independently repeated subsets, and stage-based anomaly classification. This framing treats validation as a continuing measurement activity rather than a one-time certification event.

2.2. Testbed and Procedure

The testbed should include both iOS and Android devices to capture the effects of platform security design and vendor-specific implementations on artifact availability. At a minimum, the device set should include one current iPhone, one older iPhone when a broader extraction pathway is lawfully feasible, one Pixel-class Android device, and one Samsung-class Android device. This mix is justified by the fact that iOS Data Protection and key hierarchy design affect artifact access on Apple devices, while Android file-based encryption and OEM-specific implementations affect pre-unlock and post-unlock evidence availability on Android devices [1–3]. For pilot use, four devices may be sufficient. For formal validation, the sample should be expanded across hardware generations and operating-system minor versions to reduce device-specific bias [4,6].

Ground truth should be created through scripted or tightly documented user actions that produce known artifacts across common evidentiary categories. At minimum, the populated dataset should include messaging activity, browser use, location-generating actions, media capture, and native personal-information-management data such as contacts, calendars, and call records. Because mobile app artifacts are often application-specific and may change as app versions change, application versions should be pinned and recorded for every run [5,15]. Each action should be logged against a trusted time source and supported by a secondary source of confirmation, such as screenshots, exported logs, or observer notes. Then, a controlled subset of artifacts should be deleted so that the protocol can evaluate both active and recoverable-deleted data.

For each device-tool-mode combination, the protocol should perform three repeated acquisitions under identical conditions. When feasible, a second examiner should repeat a subset of the runs on an independent workstation to assess reproducibility across operators and workstations, rather than simple rerun consistency [5,6,8,9]. Device lock state, timezone settings, network isolation, cable or interface path, workstation configuration, software versions, parsing options, and export formats should all be fixed and recorded because prior studies have shown that these factors can materially influence forensic output [4–6].

Extraction pathways are treated operationally as logical, file-system, and physical, while recognizing that specific tools may implement subtypes such as agent-based or chained workflows. The protocol does not treat encryption-imposed limits as tool failure when those limits are expected under the tested access conditions. Instead, it distinguishes expected incompleteness caused by access state from failures arising during acquisition, parsing, normalization, or reporting. This distinction is necessary for contemporary iOS and Android devices, where encryption state and lock state strongly constrain artifact availability [1–3].

2.3. Evaluation Metrics and Analysis

Analysis is conducted at both the artifact level and the run level. For clarity, the main validation

metrics are defined using set-based and time-based measures. Let G denote the scoped ground-truth artifact set for a given validation condition, and let R denote the recovered artifact set produced by a workflow run. Artifact Recovery Rate (ARR) is defined as the proportion of scoped ground-truth artifacts that are correctly recovered:

$$ARR = \frac{|G \cap \mathrm{R}|}{|G|}$$

False positives and false negatives are defined relative to the scoped ground-truth set:

$$\mathrm{FP} = |\mathrm{R} \setminus \mathrm{G}|$$

$$FN = |G \setminus R|$$

In these definitions, false positives are recovered artifacts that are not supported by the scoped ground truth, while false negatives are ground-truth artifacts that were not recovered. Expected background artifacts generated by the operating system, application cache behavior, thumbnails, or other routine device activity should not be counted as false positives unless they are incorrectly attributed to the scoped ground-truth set.

Timestamp correctness is evaluated by comparing recovered event times against documented ground-truth times after conversion to UTC and explicit documentation of timezone assumptions. Let t_i^{rec} denote the recovered timestamp for artifact i, and let t_i^{gt} denote the corresponding ground-truth timestamp. The absolute timestamp error for artifact i is defined as:

$$TE_i = |t_i^{rec} - t_i^{gt}|$$

Because prior work has shown that clock skew, drift, time zone misconfiguration, provenance differences, and interpretation errors can distort event reconstruction, timestamp reliability should be reported using robust summary measures rather than a single average value [13,14]. The recommended summaries are median absolute timestamp error, 95th-percentile timestamp error, and maximum observed timestamp error:

$$Median\ Timestamp\ Error = median(TE_i)$$

$$P95\ Timestamp\ Error = P_{95}(TE_i)$$

$$Maximum\ Timestamp\ Error = max(TE_i)$$

Repeatability is assessed across repeated acquisition runs by comparing recovered artifact sets. For two recovered artifact sets R_a and R_b, the Jaccard similarity index is defined as:

$$J(R_a, R_b) = \frac{|R_a \cap R_b|}{|R_a \cup R_b|}$$

A Jaccard value of 1.000 indicates that the recovered artifact sets are identical across the compared runs, while lower values indicate run-to-run differences. When three repeated runs are performed, the report may disclose both the mean pairwise Jaccard similarity and the minimum pairwise Jaccard similarity:

$$J_{min} = \min\left(J(R_1, R_2), J(R_1, R_3), J(R_2, R_3)\right)$$

Reproducibility is assessed through a second-examiner or independent-workstation subset when feasible. This subset should evaluate whether materially similar outputs are obtained when the same scoped ground truth, device state, extraction pathway, and tool settings are independently repeated. These measures align with prior validation and tool-testing literature that emphasizes repeatability, reproducibility, and measurable error over binary pass-fail judgments [5–9].

Anomalies are classified by process stage, including connectivity, acquisition, access or decryption, parsing, presentation, timestamp conversion, and duplication or merge behavior. This stage-based treatment is consistent with recent digital-forensics work that maps error sources across the full investigative workflow rather than attributing all observed problems to a single final output [6,12]. The final product of the analysis is a measurement-based validation record that documents what the workflow recovered, what it missed, how stable the output was across repeated runs, where anomalies arose, and under which technical conditions the results should be interpreted.

3. CASE-BASED DEMONSTRATION AND VALIDATION REPORTING OUTPUTS

3.1. Purpose and Interpretation of Validation Reporting Outputs

This section demonstrates how the proposed protocol can be translated into a disclosure-ready validation report. Rather than presenting a vendor ranking or broad performance benchmark, it shows how a controlled validation cycle should document the tested conditions, summarize recovery outcomes, report repeatability measures, and identify limits that affect interpretation. The purpose

Table 2: Android Pilot Scope and Category-Level Recovery

Category	Scoped Artifacts	Recovery Check	Result Across Three Runs	Interpretation
Documents/Text files	10	Filename/path match	10/10 in each run	File-level recovery from user-accessible storage
Camera photos	5	Filename/path match	5/5 in each run	File-level recovery, not automated image-content interpretation
Screenshots	10	Filename/path match; timestamp metadata recorded	10/10 in each run	File-level recovery; timestamp delta accuracy not computed
Downloaded images	5	Filename/path match	5/5 in each run	File-level recovery from user-accessible storage
Contact VCF entries	5	Exact identifier match inside Contacts.vcf	5/5 in each run	Entry-level VCF recovery, not broad contact-parser validation
Total	**35**		**35/35 in each run**	Scoped Android logical-acquisition subset

Table 3: Android Pilot Metric and Repeatability Summary

Metric	Run 1	Run 2	Run 3	Interpretation
Scoped artifacts	35	35	35	Same ground-truth set evaluated across repeated runs
Recovered artifacts	35	35	35	Complete scoped recovery under tested conditions
False positives	0	0	0	No unsupported artifact was attributed to scoped ground truth
False negatives	0	0	0	No scoped artifact was missed
Artifact Recovery Rate (ARR)	1.000	1.000	1.000	Complete recovery for the scoped Android subset
Screenshot timestamp metadata available	10	10	10	Metadata was available, but independent timestamp-delta accuracy was not computed
Minimum pairwise Jaccard	1.000			Recovered artifact sets were identical across repeated runs

is to make the validation output reproducible, reviewable, and suitable for later technical or legal scrutiny.

The reporting examples in this section are based on the metric definitions introduced in Section 2 and on a small controlled Android logical-acquisition pilot. The pilot provides measured outputs for a scoped set of user-accessible artifacts, while the accompanying tables illustrate how the same reporting structure can be used in broader validation deployments. This organization separates two related goals: demonstrating the practical application of the protocol and providing a general reporting template for future validation studies.

The need for this reporting structure follows from prior mobile-forensics research showing that extraction and parsing outcomes may vary by device model, operating-system version, application version, extraction pathway, access state, and parser implementation [5–9]. Recent work has also shown that frequent mobile-application updates can affect parser behavior and may require renewed reference-data generation or revalidation [5,6,15]. For this reason, validation outputs should preserve the technical context in which measurements were produced rather than presenting recovery rates or error measures as product-level claims.

The remainder of this section presents the pilot application, the configuration record, the metric-reporting structure, worked metric examples, timestamp and statistical reporting guidance, and stage-based limitations. Together, these components show how the protocol supports measured and reviewable forensic reporting while avoiding overgeneralization beyond the tested conditions.

3.2. Pilot Application: Android Logical-Acquisition Subset

To demonstrate the practical use of the proposed protocol, a small pilot application was conducted on an owned, unlocked Samsung Galaxy S8. The pilot was intentionally limited in scope and was not designed to benchmark commercial forensic tools or to generalize across Android devices. Its purpose was to show how a controlled ground-truth set can be acquired repeatedly, scored, and reported in a bounded and reviewable form. The pilot operationalizes the proposed validation metrics using measured outputs from a controlled logical-

acquisition workflow. This scoped approach follows prior mobile-forensic validation and tool-testing work that emphasizes repeatability, measurable error, and configuration-specific interpretation rather than unsupported claims of general tool correctness [6–9].

The pilot device was a Samsung Galaxy S8 running Android 9 with security patch level December 1, 2020. The acquisition workstation was a Windows 11 system using Android Debug Bridge (ADB) version 1.0.41. USB debugging was enabled on the device, and the device was unlocked during acquisition. The same USB cable, workstation, ADB executable, device access state, and acquisition procedure were used across all repeated runs. Before each run, device recognition was confirmed using `adb devices`, and the acquisition was limited to user-accessible storage exposed through the Android shared storage path. The pilot did not attempt full-file-system extraction, physical acquisition, deleted-data recovery, app-private database acquisition, or bypass of Android security controls.

The Android pilot used ADB logical acquisition of selected user-accessible storage locations. The scoped ground-truth set contained 35 artifacts across five categories: 10 user-created text files stored under `/sdcard/Documents/`, five camera photos stored under `/sdcard/DCIM/Camera/`, 10 screenshots stored under `/sdcard/DCIM/Screenshots/`, five downloaded image files stored under `/sdcard/Download/`, and five contact entries contained in an exported Contacts.vcf file. The contact file was exported from the device contacts application and placed within the acquired user-accessible storage area before acquisition. Cache files, thumbnails, hidden system-generated files, and background artifacts were excluded from the scoped ground-truth set unless they were explicitly defined as part of the validation set.

For each run, the same source directories were copied from the device to a clean run-specific folder on the workstation. The acquisition procedure followed a simple logical-copy workflow using ADB commands such as

```
adb pull /sdcard/Documents/,
adb pull /sdcard/DCIM/Camera/,
adb pull /sdcard/DCIM/Screenshots/,
adb pull /sdcard/Download/.
```

After each acquisition, a manifest was generated for the recovered files. The manifest recorded the recovered file path, filename, file size, and hash value where applicable. The manifest was then compared against the predefined ground-truth list. This comparison served as the basis for artifact recovery, false-negative, false-positive, and repeatability scoring.

File-based artifacts were scored as recovered only when the expected filename and expected relative path were present in the acquisition manifest. Where file size or hash values were available, they were used as additional confirmation that the recovered file corresponded to the expected ground-truth artifact. Contact artifacts were scored at the VCF-entry level by exact identifier matching inside the exported `Contacts.vcf` file. An artifact was counted as a false negative if it appeared in the scoped ground-truth list but was absent from the recovered output. An artifact was counted as a false positive only if an unsupported recovered item was incorrectly attributed to one of the scoped ground-truth artifacts. Background files, thumbnails, and automatically generated device artifacts were not counted as false positives unless they were incorrectly presented as part of the scoped validation set.

Three repeated acquisition runs were performed under fixed device, workstation, access-state, and collection conditions. Each run copied the same user-accessible storage locations and evaluated the same scoped ground-truth set. The scored subset produced complete recovery across all three repeated acquisition runs. All 35 scoped artifacts were recovered in Run 1, Run 2, and Run 3. No false negatives were observed within the scoped ground-truth set. False positives were recorded as zero because no unsupported artifacts were attributed to the predefined ground-truth artifacts.

Repeatability was evaluated using Jaccard similarity across the recovered artifact sets from the three acquisition runs. The recovered artifact set was identical across Run 1, Run 2, and Run 3. Therefore, the minimum pairwise Jaccard value across the three run pairs was 1.000. This result supports repeatability only for the tested device, Android version, ADB logical-copy pathway, workstation environment, artifact categories, and user-accessible storage scope.

Timestamp information was recorded where it was available in recovered file metadata. In particular, timestamp metadata was available for the

Table 4: Validation Scope and Controlled Conditions

Element	What Should Be Fixed or Documented	Why It Matters
Device class	iPhone, Pixel-class Android, Samsung-class Android	Platform and OEM differences affect artifact availability
OS version	Major and minor versions tested	Results are version-specific
Access state	BFU, AFU, locked, unlocked, recently unlocked	Encryption and lock state constrain evidence access
Toolchain	Acquisition component, parser version, export module	Different stages may affect output differently
Extraction mode	Logical, file-system, physical-style, agent-based	Mode affects recovery depth and artifact visibility
App version	Version for targeted apps or artifact sources	App updates can change storage and parser compatibility
Workstation environment	OS, hardware, interface path, cable/adapter	Environmental differences may affect reproducibility
Parser options	Deleted-item parsing, WAL/sidecar handling, normalization options	Output can vary with settings, even within one toolchain
Time settings	Device timezone, workstation timezone, UTC conversion rule	Timestamp analysis depends on explicit time handling
Repeat-run design	Number of repeated runs and second-examiner subset	Needed to evaluate repeatability and reproducibility
Cloud or synchronization state	Account login status, network state, sync enabled/disabled, cached versus local artifact scope	Cloud-linked artifacts may not represent a complete local source of truth
AI-assisted processing	AI-assisted parsing, classification, grouping, or triage settings; module/model version; review procedure	Automated interpretation may introduce unstable labels or opaque grouping decisions

10 screenshot artifacts in each run. However, independent timestamp-delta accuracy was not computed because separate trusted creation-time records were not available for this pilot. This limitation is important because prior forensic timestamp research shows that timestamp interpretation can be affected by device clock conditions, time-zone handling, provenance uncertainty, and tool-level normalization choices [13,14]. Therefore, the pilot should be interpreted as demonstrating file-level recovery, VCF-entry recovery, and repeatability, not timestamp accuracy. In a full validation deployment, each artifact should be created against a trusted time source, and recovered timestamps should be compared after documented normalization to UTC.

A defensible statement based on this pilot is bounded: under the tested Galaxy S8 device state, Android version, Windows workstation environment, ADB version, ADB logical-copy pathway, and user-accessible storage scope, all 35 scoped artifacts were recovered across three repeated acquisition runs, producing ARR and minimum pairwise Jaccard values of 1.000. This statement should not be generalized to app-private databases, deleted artifacts, full-file-system acquisition, physical acquisition, cloud-synchronized artifacts, AI-assisted parsing, image-content interpretation, timestamp-delta accuracy, or other Android devices. The value of the pilot is that it demonstrates how the proposed validation protocol can produce measured, repeatable, and clearly scoped reporting outputs from a controlled mobile forensic workflow. Therefore, the pilot should be read as a protocol demonstration rather than as a benchmark of ADB, Android, or any commercial forensic tool.

3.3. Configuration and Context Record

Table **4** defines the configuration record that should accompany any validation output before recovery rates, timestamp measures, or anomaly counts are interpreted. The table identifies the core conditions that make the results traceable, including device class, operating-system version, access state, extraction mode, toolchain components, application version, workstation environment, parser options, time-handling assumptions, and repeat-run design. These elements are necessary because mobile-forensic output is shaped by the interaction of platform security design, device state, workflow configuration, and software versioning [4–9].

The configuration record provides the context needed to distinguish a measured workflow limitation from a change in test conditions. For example, a missing artifact may reflect encryption

state, pre-unlock versus post-unlock access, unsupported device conditions, parser settings, environmental differences, or application-version changes rather than a simple acquisition failure [1–3,5,6]. For app-specific artifacts, the application version and relevant synchronization state should be documented because frequent application updates and cloud-linked behavior can change local artifact availability and parser compatibility [5,15]. When AI-assisted parsing or automated artifact grouping is used, the tool setting, model or module version, and output-review procedure should also be recorded so that later interpretation does not depend on undocumented automation.

The entries in Table **4** also support repeatability and reproducibility analysis. Device class, operating-system version, access state, and extraction mode define the technical scope of the validation run. Toolchain version, parser options, export settings, and workstation environment help identify whether differences across runs arise from the forensic workflow or from uncontrolled environmental variation.

Time settings, including device timezone, workstation timezone, and UTC normalization rules, provide the basis for later timestamp-error analysis. Repeat-run design records the number of repeated acquisitions and any second-examiner or independent-workstation subset used to assess reproducibility.

A disclosure-ready report should present the configuration record before reporting artifact recovery, temporal error, repeatability, or anomaly results. This ordering ensures that the numerical metrics are interpreted as measurements produced under documented technical conditions rather than as context-free statements of tool performance.

3.4. Validation Metrics and Reporting Interpretation

A validation report should present a compact set of measurements that describe artifact recovery, temporal reliability, output stability, and remaining interpretive risk.

Table **5** summarizes the core metrics used in this protocol and shows how each metric should be reported. The table is intended to translate the methodological definitions in Section 2.3 into a reporting format that can be used in laboratory documentation, peer review, and later evidentiary examination. This structure is consistent with digital-forensics reliability work that emphasizes documented testing, repeatability, reproducibility, and error disclosure [6,12].

The first group of metrics concerns artifact recovery. Artifact Recovery Rate (ARR), false positives, and false negatives should be reported by artifact class rather than as a single overall score. Category-level reporting is important because mobile forensic workflows may recover some artifact types more consistently than others. For example, a workflow may recover contacts, media files, or call records while missing application-specific messages, location artifacts, sidecar databases, transaction logs, or version-sensitive parser outputs [5,8,15]. Reporting recovery by category helps identify which findings can be interpreted with greater confidence and which require additional caution.

The second group of metrics concerns timestamp reliability. Timestamp correctness should be evaluated separately from artifact recovery because an artifact may be present while its displayed time is shifted or uncertain. As described in Section 2.3, this protocol reports median absolute timestamp error, upper-percentile error, and maximum observed error. These summaries are preferable to a single average because timestamp discrepancies may arise from clock skew, timezone handling, daylight-saving interpretation, provenance ambiguity, or parser normalization choices [13,14]. A validation report should disclose both the typical timestamp behavior and upper-tail error that may affect event reconstruction.

The third group of metrics concerns output stability. Repeatability is measured by comparing recovered artifact sets across repeated runs. In this protocol, the Jaccard similarity index is used for that purpose because it measures the overlap between recovered artifact sets. A value of 1.000 indicates that the recovered artifact sets were identical across runs, while lower values indicate run-to-run differences. Reproducibility should be assessed, when feasible, through a second-examiner or independent-workstation subset. Together, these measures show whether the workflow produces materially similar results when the same test conditions are repeated [5–9].

Table **5** provides the minimum metric structure for a validation summary: what was recovered, what was missed, what unsupported artifacts appeared, how stable the output was, how reliable the

Table 5: Core Metrics and Reporting Interpretation

Metric	What It Measures	Recommended Reporting Form	Interpretation
Artifact Recovery Rate (ARR)	Recovery of in-scope ground-truth artifacts	Percentage by artifact class and workflow	Shows category-level recovery strength
False Positives (FP)	Reported artifacts not supported by the scoped ground truth	Count with a short note on probable cause	Indicates parsing, duplication, or attribution issues
False Negatives (FN)	Ground-truth artifacts not recovered	Count by artifact class	Shows omissions relative to the scoped claim
Median Absolute Timestamp Error	Typical absolute time deviation	Seconds or minutes	Robust summary of the usual temporal shift
P95 Timestamp Error	Upper-tail timestamp deviation	Seconds or minutes	Shows how large timestamp errors become in difficult cases
Maximum Timestamp Error	Largest observed timestamp deviation	Seconds or minutes	Highlights the worst-case temporal risk
Jaccard Repeatability	Similarity of normalized artifact sets across runs	0–1 similarity value	Measures run-to-run stability
Reproducibility Result	Agreement across independently repeated subsets	Percentage or qualitative summary	Indicates whether findings remain stable across examiner/workstation changes
Stage-Based Anomaly Count	Number of issues by process stage	Count by stage	Helps localize where problems arise in the workflow

Table 6: Worked Example of Validation Metric Calculation

Metric	Example Ground Truth / Output	Calculation	Interpretation
Artifact Recovery Rate	45 recovered / 50 ground-truth artifacts	45 / 50 = 0.90	90% of scoped artifacts were recovered
False Positives	3 unsupported reported artifacts	FP = 3	Possible parsing, duplication, or attribution issue
False Negatives	5 known artifacts not recovered	FN = 5	Omission relative to the scoped ground truth
Timestamp Error	Recovered: 10:05:43 UTC; Ground truth: 10:05:20 UTC	|23 sec|	Small but reportable temporal deviation
Jaccard Repeatability	Run 1 = 45 items, Run 2 = 44 items, overlap = 43, union = 46	43 / 46 = 0.935	High run-to-run similarity
Reproducibility Subset	28 matching items / 30 independently repeated items	28 / 30 = 0.933	Stable across examiner/workstation subset

timestamps were, and what anomaly categories remained relevant. Table **6** illustrates how these metrics can be calculated from simple validation outputs. Tables **5** and **6** connect the protocol's measurement concepts to practical reporting. The resulting metrics should be interpreted together with the configuration record and the stage-based anomaly taxonomy.

3.5. Timestamp-error Reporting and Statistical Summaries

Figure **1** summarizes the proposed validation workflow as a seven-stage reporting pipeline. The process begins with controlled ground-truth creation and documented device and workflow setup, then proceeds through repeated acquisition, artifact matching, metric computation, anomaly classification, and reporting and disclosure. This numbered workflow clarifies how the tested conditions, recovered artifacts, computed metrics, observed anomalies, and final interpretive statement are connected. The feedback loop indicates that validation should be repeated when tool versions, operating-system or application versions, parser behavior, access state, acquisition conditions, or observed anomalies materially change the tested workflow. Timestamp reliability remains one important part of this workflow and should be reported separately from artifact presence because an artifact may be recovered while its displayed time is shifted by timezone handling, clock skew, daylight-saving interpretation, provenance ambiguity, or parser assumptions [13,14].

For measured results, statistical analysis should proceed in three layers. First, paired artifact-level recovery comparisons between workflows on the

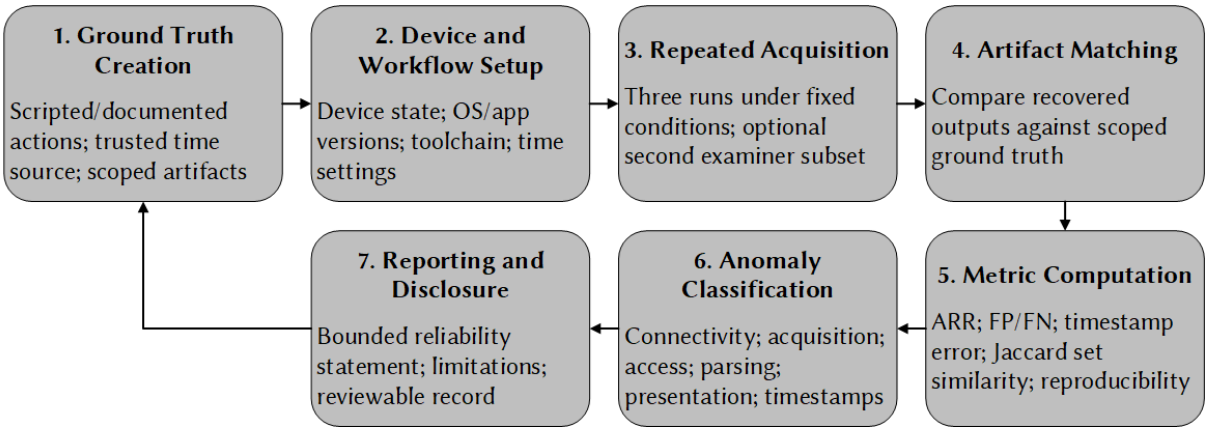

Figure 1: Mobile forensic workflow validation and reporting pipeline. The protocol proceeds through seven stages: ground-truth creation, device and workflow setup, repeated acquisition, artifact matching, metric computation, anomaly classification, and reporting and disclosure. The feedback loop indicates that revalidation may be needed when tool versions, operating-system or application versions, parser behavior, access state, acquisition conditions, or observed anomalies change the tested workflow.

same device and the same ground-truth item can be evaluated with McNemar's test for each artifact class. Second, timestamp-error summaries should rely on robust nonparametric statistics because the expected distributions are typically non-normal and heavy-tailed; the primary descriptive outputs should include median absolute error, median absolute deviation, and upper-percentile summaries rather than the mean alone [13,14].

Group comparisons may then be performed with Kruskal-Wallis tests for more than two groups and Mann-Whitney tests for pairwise contrasts. Third, repeatability across runs should be reported using set-based similarity measures such as the Jaccard index, together with the mean and minimum similarity across repeated extractions. For all inferential comparisons, the report should also disclose sample size, effect size, and confidence intervals where appropriate.

3.6. Limitations and Reporting Cautions

The final part of the reporting structure should make explicit the limits that accompany measured outcomes and should identify where interpretive risk arises within the workflow. Table **7** provides a stage-based anomaly taxonomy for this purpose. Rather than collapsing all problems into a single generic notion of tool failure, it organizes observed issues by workflow stage, including connectivity, acquisition, access or decryption, parsing, presentation, timestamp conversion, duplication or merge behavior, and reproducibility. This structure is consistent with recent digital-forensics research that maps error sources across the investigative process and emphasizes that reliability concerns often arise from different technical layers rather than from one final output alone [6,12,17]. It also matches the anomaly-classification logic defined in Section 2.3.

Several limitations should be disclosed whenever this protocol is used with real measured data, especially scalability, cross-device generalization, dependence on controlled ground truth, access-state constraints, timestamp uncertainty, and workflow-stage anomalies. First, scalability remains a practical limitation. The pilot application demonstrates the protocol on a small Android logical-acquisition subset, but broader validation would require additional devices, operating-system versions, application versions, extraction modes, toolchains, repeated runs, and examiner or workstation combinations. As the number of tested configurations increases, the cost of ground-truth generation, acquisition, artifact matching, timestamp verification, and anomaly review also increases. Therefore, the protocol is scalable in structure, but its full implementation requires careful sampling decisions, automation support, and prioritization of high-risk artifact categories or commonly used workflow configurations.

Second, cross-device generalization is limited. Results obtained from one device, such as the Android pilot in this paper, should not be generalized to other Android models, iOS devices, other operating-system versions, different security patch levels, different lock states, or different extraction

Table 7: Stage-based Anomaly Taxonomy for Reporting

Stage	Example Symptom	Likely Interpretation	Reporting Implication
Connectivity	Device not recognized or unstable connection	Interface, cable, driver, or trust-state issue	Record as a setup/connectivity issue, not parser failure
Acquisition	Extraction incomplete or aborted	Workflow interruption or unsupported path	Separate from later parsing or presentation problems
Access / Decryption	Expected protected content unavailable	Lock state or encryption-limited access	Report as a scope/access limitation rather than an ordinary tool failure
Parsing	Known artifact source present but incompletely decoded	Parser limitation or version mismatch	Flag as a parsing weakness affecting artifact interpretation
Presentation	Duplicate rows, missing fields, inconsistent labels	Export or display-layer issue	Preserve native artifacts and note presentation caveat
Timestamp Conversion	Shifted times, inconsistent zones, DST mismatch	Time normalization or provenance problem	Report both original and derived time when applicable
Duplication / Merge	One artifact is shown multiple times or merged incorrectly	Normalization or deduplication issue	Note the risk of count inflation or attribution ambiguity
Reproducibility	Different outputs under nominally fixed conditions	Workflow instability or environmental variance	Escalate as a repeatability/reproducibility concern

pathways. Mobile forensic output is shaped by device model, OEM implementation, encryption state, application behavior, parser support, and acquisition mode. For this reason, each validation result should be interpreted as applying only to the tested configuration unless additional devices and workflow conditions are independently evaluated.

Third, the protocol depends on the quality and scope of the controlled ground-truth set. Controlled ground truth makes recovery, false-positive, false-negative, timestamp, and repeatability measurements possible, but it may not capture the full complexity of real investigations. Real devices may contain background system artifacts, cloud-synchronized records, deleted remnants, partially cached application data, inconsistent user behavior, and artifacts created outside the examiner's observation. Therefore, ARR, FP, FN, and timestamp-error values should be interpreted relative to the defined ground-truth scope rather than as complete measures of all possible evidence on the device.

Fourth, ground-truth completeness is always scoped rather than absolute. Even in controlled populations, mobile operating systems and applications generate background artifacts that complicate false-positive assessment and can blur the boundary between expected system activity and investigator-created content. For that reason, artifact recovery, false positives, and false negatives should always be interpreted relative to an explicit scope of claim, such as user-generated artifacts only or all artifacts within selected databases [6,12]. Table **7** helps support that interpretation by distinguishing whether an apparent problem should be treated as a scoped limitation, a workflow weakness, or a downstream presentation issue.

Fifth, the encryption state and access state must be separated from ordinary workflow failures. Missing artifacts on iOS or Android may reflect lock state, credential availability, or encryption-imposed limits rather than a defect in the acquisition or parsing process. The report should classify each gap in a way that preserves this distinction, for example, as unsupported by design, inaccessible under tested conditions, acquisition failure, parsing failure, or presentation error. This distinction is essential for contemporary mobile devices, where Data Protection classes, file-based encryption, and device-state-dependent key access directly shape what evidence can be exposed under a given workflow [1–3]. Table **7** is useful here because it gives the report a consistent vocabulary for separating expected incompleteness from measured reliability concerns.

Sixth, timestamp errors must not be over-attributed to the workflow alone. Some deviations arise from device configuration, clock drift, ambiguous timezone history, or provenance uncertainty rather than from defective parsing. A defensible report should preserve both the original recovered timestamp and any derived corrected field, together with the justification for the correction method [13,14]. In practice, this means that timestamp conversion issues should be described not merely as "wrong time," but as a stage-specific

interpretive risk whose source may lie in device settings, parser logic, export behavior, or unresolved provenance uncertainty. The timestamp-conversion row in Table **7** is intended to support that kind of disciplined disclosure.

The value of Table **7** is practical as well as conceptual. It gives the validation record a compact way to document where anomalies arose, what they may imply, and how they should affect interpretation. These limitations reinforce the main reporting principle of the protocol: validation results should be stated as bounded, configuration-specific findings rather than generalized claims about a tool, platform, or workflow.

4. DISCUSSION

The central implication of this protocol is that mobile-forensic reliability should be treated as a measurable property of a specific workflow configuration rather than as a general attribute of a vendor or product family. In mobile forensics, output depends on the tested toolchain, version, device model, operating-system state, access condition, extraction pathway, parser behavior, application version, timestamp handling, and reporting logic. Earlier mobile tool-testing studies already argued that the diversity of devices, proprietary implementations, and rapid technical turnover make stable error estimation difficult unless validation is structured and continuously documented [7–9]. More recent validation research extends that position by showing that reliability information must be documented at the levels of technology, method, and application, and that inadequate documentation can leave neither the court nor opposing experts in a position to evaluate the trustworthiness of the reported findings [6,18]. The primary value of the present protocol is not that it produces a single performance score, but that it structures how laboratories generate, bound, and explain reliability under tested conditions [6,12].

This framing has direct implications for admissibility and expert testimony. A Daubert-like reliability inquiry asks whether a method has been tested, whether known or potential error can be described, whether standards or controls guide its use, and whether the method has been applied reliably under the relevant conditions. In mobile forensics, these questions cannot be answered by naming a commercial tool alone. Artifact availability and interpretive accuracy are shaped by device model, operating-system version, access state, application version, extraction pathway, parser behavior, timestamp handling, and reporting logic. A result may be incomplete for reasons that are technically expected, or it may be inaccurate because of acquisition, parsing, presentation, duplication, or temporal-conversion error. Treating all missing, shifted, duplicated, or unsupported artifacts as a single undifferentiated “tool problem” is analytically weak. The protocol proposed here responds to this admissibility concern by requiring version-specific validation records, repeated acquisitions, timestamp-delta analysis, category-specific recovery measures, and stage-based disclosure of unsupported, inaccessible, omitted, and ambiguously interpreted artifacts [6,7,11,17].

A related implication concerns the structure of the expert report and the basis for expert testimony. Peer-reviewed reliability studies have shown that many digital-forensic reports do not document enough information to trace what actions were performed, what tools and settings were used, how findings were linked to their source, and whether the method had been validated for the precise conditions of use (6). That observation is particularly important for mobile devices, where different extraction paths may yield materially different evidence sets even when the same handset is examined. For that reason, a report based on mobile-tool output should disclose, at a minimum, five classes of information: the toolchain used, the device context, the operational extraction pathway, the integrity trail, and the time-handling assumptions and known limitations [6,12,14,16,18]. These disclosure elements do not merely improve presentation quality. They help the examiner explain the factual basis, tested conditions, known limitations, and potential error sources behind the opinion. In that sense, the validation record supports expert testimony by allowing conclusions to be stated as bounded technical findings rather than as broad assertions of tool accuracy.

The protocol also has methodological implications beyond courtroom use. Recent digital-forensics research argues that validation should move away from binary notions of “tool passed” or “tool failed” and toward structured measurement of repeatability, reproducibility, and stage-specific error propagation [6,12,14]. This perspective fits mobile forensics research showing that category-level performance matters more than a single overall recovery score. A workflow may recover contacts and media consistently while missing app-specific

message records or translating timestamps inconsistently across exports. By requiring controlled ground truth, repeated runs, metric-based analysis, and explicit anomaly classification, the present framework encourages laboratories to treat validation as an ongoing empirical program rather than a one-time procurement exercise [6–9]. It also supports more defensible expert testimony because conclusions can be stated in bounded form, for example, that under the tested software version, device state, access condition, and extraction pathway, the workflow recovered a specified proportion of ground-truth artifacts with a documented range of timestamp error. That form of statement is more scientifically supportable than an unqualified claim that a tool is accurate or an industry standard [6,7,11].

Several limitations should nevertheless be stated clearly. First, this manuscript is a protocol and reporting framework, not a definitive comparative evaluation of vendor performance. Its illustrative tables, figure, and pilot demonstration are intended to show the form of a defensible validation record, not to establish universal rankings across tools. Second, the Android pilot is intentionally narrow. It demonstrates the protocol on a scoped logical-acquisition subset from one unlocked Samsung Galaxy S8, but it does not support generalization to other Android models, iOS devices, other operating-system versions, different security patch levels, locked or before-first-unlock states, full-file-system acquisition, physical acquisition, deleted artifacts, app-private databases, cloud-synchronized records, or AI-assisted parsing outputs. Third, any broader deployment of the protocol will remain bounded by lawful access, supported devices, feasible acquisition modes, available toolchains, and the technical realities of encrypted platforms. On modern devices, some forms of incompleteness arise from the technical environment rather than from a tool defect alone [1–3,17,18].

Fourth, the protocol depends on the quality and scope of the controlled ground-truth set. Controlled ground truth makes recovery, false-positive, false-negative, timestamp, and repeatability measurements possible, but it may not capture the full complexity of real investigations. Real devices may contain background system artifacts, cloud-synchronized records, deleted remnants, partially cached application data, inconsistent user behavior, and artifacts created outside the examiner's observation. Therefore, ARR, FP, FN, timestamp-error, and repeatability values should be interpreted relative to the defined ground-truth scope rather than as complete measures of all possible evidence on the device. Fifth, timestamp reliability must remain a separate analytical concern. Recent work on time anchors and timestamp interpretation shows that a recovered artifact can still be misinterpreted if the device clock, time zone history, provenance chain, or parser logic is wrong [13,14]. For that reason, the protocol's requirement to report both recovered timestamps and justified derived-time fields should be treated as a core part of expert disclosure rather than an optional refinement [14,16].

This discussion supports a practical conclusion: mobile-forensic validation should be expressed as measured, configuration-specific reliability rather than assumed product-level trust. A defensible laboratory protocol should document tested conditions, quantify recovery and temporal error, distinguish expected access limits from workflow failures, classify anomalies by workflow stage, and disclose interpretive constraints in a form that can be reviewed and challenged. That shift from general assurance to bounded empirical disclosure is the main methodological contribution of this paper [6,12,16,18].

5. CONCLUSION

This paper frames mobile forensic validation as a problem of measured, configuration-specific reliability rather than assumed product-level trust. The main contribution is not a new extraction method, but a practical validation protocol that defines controlled ground truth, repeated acquisitions, artifact-level and run-level metrics, timestamp-focused error analysis, stage-based anomaly classification, and disclosure-ready reporting under stated laboratory conditions [6–9,12,18]. By organizing validation around artifact recovery, false positives, false negatives, timestamp deviation, repeatability, reproducibility, and workflow-stage anomalies, the protocol provides a structured way to document what a workflow recovered, what it missed, how stable the output was, where interpretive risk arose, and under which technical conditions the results should be understood [6,12,14,16].

The bounded Android logical-acquisition pilot demonstrates how the protocol can be applied to a controlled mobile forensic workflow without overstating the findings beyond the tested device, access state, acquisition pathway, artifact categories, and user-accessible storage scope. The pilot results

show complete recovery and identical recovered artifact sets across three repeated runs for the scoped artifacts, but they do not support broader claims about deleted data, app-private databases, full-file-system acquisition, physical acquisition, timestamp-delta accuracy, cloud-synchronized artifacts, AI-assisted parsing, or other devices. This bounded interpretation is central to the proposed protocol because validation findings should be reported as measured results under documented conditions rather than as general claims of tool correctness.

This approach supports more rigorous laboratory practice and more defensible forensic reporting because conclusions can be expressed in bounded form rather than as unqualified claims of tool accuracy or industry-standard reliability [6,11,18]. Future work should expand the pilot into broader multi-device, multi-tool, timestamp-validated, and app-private-data experiments. Additional work should also include iOS and Android devices across multiple operating-system versions, different access states, multiple extraction pathways, cloud-linked artifacts, AI-assisted parsing workflows, and stronger inter-examiner reproducibility studies so that the framework can develop into a broader empirical basis for mobile-forensic reliability assessment [5,6,8,9,14,18].

ACKNOWLEDGEMENTS

The authors have no acknowledgements to declare.

DECLARATION OF THE USE OF AI TOOLS

ChatGPT was used only to assist with grammar and phrasing edits to improve clarity. All content was reviewed, verified, and approved by the authors.

CONFLICT OF INTEREST

The authors declare that there is no conflict of interest.

ETHICS

This article was prepared in accordance with ethical principles for scholarly publication. The corresponding author confirms that all authors have read, revised, and approved the final manuscript.

FUNDING

This research received no external funding.

REFERENCES

[1] Teufl P, Zefferer T, Stromberger C, Hechenblaikner C. iOS encryption systems: Deploying iOS devices in security-critical environments. In: 2013 International Conference on Security and Cryptography (SECRYPT). 2013. p. 1–13.

[2] Groß T, Ahmadova M, Müller T. Analyzing Android's File-Based Encryption: Information Leakage through Unencrypted Metadata. In: Proceedings of the 14th International Conference on Availability, Reliability and Security [Internet]. New York, NY, USA: Association for Computing Machinery; 2019 [cited 2026 Mar 17]. p. 1–7. (ARES '19). Available from: https://dl.acm.org/doi/10.1145/3339252.3340340 doi:10.1145/3339252.3340340

[3] Fukami A, Stoykova R, Geradts Z. A new model for forensic data extraction from encrypted mobile devices. Forensic Sci Int Digit Investig. 2021 Sep 1;38:301169. doi:10.1016/j.fsidi.2021.301169

[4] Alshameri F, Khanta K, Boyce S. A comparison study to analyse the data acquisitions of iOS and android smartphones using multiple forensic tools. Int J Electron Secur Digit Forensics. 2024 Jan;16(3):267–83. doi:10.1504/IJESDF.2024.138325

[5] Claij-Swart AA, Oudsen E, Timbermont B, Hargreaves C, Voigt LL. Automatically generating digital forensic reference data triggered by mobile application updates. Forensic Sci Int Digit Investig. 2025 Oct 1;DFRWS APAC 2025 - Selected Papers from the 5th Annual Digital Forensics Research Conference APAC54:301985. doi:10.1016/j.fsidi.2025.301985

[6] Stoykova R, Franke K. Reliability validation enabling framework (RVEF) for digital forensics in criminal investigations. Forensic Sci Int Digit Investig. 2023 Jun 1;45:301554. doi:10.1016/j.fsidi.2023.301554

[7] Baggili I, Mislan R, Rogers M. Mobile Phone Forensics Tool Testing: A Database Driven Approach. Electr Comput Eng Comput Sci Fac Publ. 2007 Jan 1.

[8] Saleem S, Popov O, Appiah-Kubi OK. Evaluating and Comparing Tools for Mobile Device Forensics Using Quantitative Analysis. In: Rogers M, Seigfried-Spellar KC, editors. Digital Forensics and Cyber Crime. Berlin, Heidelberg: Springer; 2013. p. 264–82. doi:10.1007/978-3-642-39891-9_17

[9] Anobah M, Saleem S, Popov O. Testing Framework for Mobile Device Forensics Tools. J Digit Forensics Secur Law. 2014 Jan 1;9(2). doi:10.15394/jdfsl.2014.1183

[10] Scurich N, Faigman DL, Albright TD. Scientific guidelines for evaluating the validity of forensic feature-comparison methods. Proc Natl Acad Sci. 2023 Oct 10;120(41):e2301843120. doi:10.1073/pnas.2301843120

[11] Stoykova R. Digital evidence: Unaddressed threats to fairness and the presumption of innocence. Comput Law Secur Rev. 2021 Sep 1;42:105575. doi:10.1016/j.clsr.2021.105575

[12] Horsman G. Sources of error in digital forensics. Forensic Sci Int Digit Investig. 2024 Mar 1;48:301693. doi:10.1016/j.fsidi.2024.301693

[13] Schatz B, Mohay G, Clark A. A correlation method for establishing provenance of timestamps in digital evidence. Digit Investig. 2006 Sep 1;The Proceedings of the 6th Annual Digital Forensic Research Workshop (DFRWS '06)3:98–107. doi:10.1016/j.diin.2006.06.009

[14] Vanini C, Hargreaves CJ, van Beek H, Breitinger F. Was the clock correct? Exploring timestamp interpretation through time anchors for digital forensic event reconstruction. Forensic Sci Int Digit Investig. 2024 Jul 1;DFRWS USA 2024 - Selected Papers from the 24th Annual Digital Forensics

Research Conference USA49:301759. doi:10.1016/j.fsidi.2024.301759

[15] Aagaard P, Dinyarian B, Abduljabbar O, Choo KKR. Family locating sharing app forensics: Life360 as a case study. Forensic Sci Int Digit Investig. 2023 Mar 1;44:301478. doi:10.1016/j.fsidi.2022.301478

[16] Hargreaves C, Nelson A, Casey E. An abstract model for digital forensic analysis tools - A foundation for systematic error mitigation analysis. Forensic Sci Int Digit Investig. 2024 Mar 1;DFRWS EU 2024 - Selected Papers from the 11th Annual Digital Forensics Research Conference Europe48:301679. doi:10.1016/j.fsidi.2023.301679

[17] Losavio M, Wilson D, Elmaghraby A. Prevalence, Use, and Evidentiary Issues of Digital Evidence of Cellular Telephone Consumer and Small-Scale Digital Devices. J Digit Forensic Pract. 2007 Jun 22;1(4):291–6. doi:10.1080/15567280701418080

[18] Stoykova R, Andersen S, Franke K, Axelsson S. Reliability assessment of digital forensic investigations in the Norwegian police. Forensic Sci Int Digit Investig. 2022 Mar 1;40:301351. doi:10.1016/j.fsidi.2022.301351

Received on 08-04-2026 Accepted on 11-05-2026 Published on 15-05-2026

https://doi.org/10.65879/3070-5789.2026.02.03