AI Did It: Why Digital Investigative Analysts Must Not Outsource Judgment to Artificial Intelligence

As AI agents begin to read, write, and move files on their own, the profession needs to be ready for a new defense, and for a renewed commitment to validation.

By Ovie Carroll
SANS Principal Instructor and Co-Author, FOR500
June 2026

The Coming SAIDI Defense

For as long as there have been trials, there has been the claim that someone else is responsible. In digital evidence cases the argument is so common that it has a nickname: **SODDI**, short for "Some Other Dude Did It." The account was shared. The WiFi was open. There was malware on the machine. A roommate, a coworker, or an unknown intruder, anyone but the defendant, must have been at the keyboard.

SODDI is about to get a successor. As artificial intelligence agents, copilots, and automation frameworks begin to read files, draft documents, move data between local folders and cloud storage, and drive browsers and command shells on a user's behalf, a new version of the argument is taking shape. I'm calling it **SAIDI**: "Some Artificial Intelligence Did It." The defense will not be that an unknown human acted. It will be that an AI tool acted: that it misunderstood an instruction, executed a task no one intended, invented a step that was never requested, modified a file on its own, or generated artifacts that merely look like the product of human activity.

This is not a distant problem, and it is not a thought experiment. The tools that make such a defense plausible are already running on ordinary computers, installed by ordinary users for ordinary reasons. The question for our profession is whether we are ready to answer the argument, not with speculation, but with disciplined analysis of what the artifacts do and do not establish. To show why that matters, I want to walk through a small, controlled test I conducted on a Windows 11 system, then use it to make a larger argument about attribution, validation, and the role of AI inside our own tools.

From SODDI to SAIDI: Why the New Defense Is Different

SODDI has always been answered with attribution evidence. We place a person at a device through a combination of artifacts: a login session, a shortcut (LNK) file created when a document is opened, a jump list entry, a RecentDocs or OpenSavePidMRU registry value, an Office most-recently-used list, a reading location that shows where the user left off. No

single artifact proves identity, but together they build a picture of interactive human use: a person sitting in front of the machine, opening files through Windows Explorer and Microsoft Word, making choices that the operating system dutifully records.

SAIDI is different in a way that should concern every examiner and legal professional. AI agents frequently do not act through the interactive user interface at all. They read and write files programmatically. They issue commands through non-interactive shells. They call software libraries that manipulate documents directly, never opening the application a human would use. When an AI agent works this way, the familiar attribution artifacts are not weak or ambiguous; they are simply never created. The activity is real and the file genuinely changes, yet the traces we have spent careers learning to interpret are absent, scattered, or replaced by something unfamiliar.

That creates two distinct risks. The first is that an examiner sees the absence of the usual artifacts and concludes, wrongly, that nothing happened. The second is the mirror image: an examiner sees a modified file in a user's personal folder and concludes, just as wrongly, that the user personally made the change. SAIDI lives in the gap between those two errors. To close the gap, we have to understand what agentic activity actually looks like on disk.

The Test: An AI Agent, a OneDrive Document, and a Word File

The test was simple by design. On a Microsoft Surface running Windows 11, version 25H2, build 26200, a Microsoft Word document, created on a separate system, named Protein-Integration-with-Alien-Tech.docx, was placed in the OneDrive Documents folder of the test system under a user profile named "ryans." OpenClaw, an open-source autonomous AI agent framework, configured to use Anthropic's Claude as its language model was running on the system under that user's account, with a local web control panel through which instructions could be issued. The instruction, preserved verbatim in the agent's own session log, was to open the named document, add an executive summary paragraph, save it, and leave it open.

What followed was the kind of sequence we will increasingly be asked to reconstruct. The agent's session log recorded the instruction at 15:41:01 UTC. The agent read the document twice, composed a summary, inserted it, and saved the file, recording a "Saved OK" result at 15:41:44 UTC. The file system agreed: the NTFS USN change journal recorded the document's data being rewritten at 15:41:44.38 UTC, matching the agent's save to the same second. A few seconds later the agent launched the saved file in its default application, and Microsoft Word 2016 started at 15:41:48 UTC, eventually recording the document, anomalous among its jump lists artifact by reference to its OneDrive cloud address.

So far, through the change journal, this could look like an ordinary edit. The decisive detail is how it was done. The change was not typed into Microsoft Word by a human. It was made programmatically by the agent using the python-docx software library, executed silently through Windows PowerShell launched with the "`-NoProfile -NonInteractive -Command`" switches,

with no visible window, application, or user interaction. The Windows PowerShell event log, Event ID 600, captured the Python that imported the document and appended the summary. The agent's workspace contained a Python environment with python-docx installed. Word opened only afterward, as instructed by the agent, and the evidence indicates it did so to display a file that had already been saved, not to author the change.

What the Artifacts Show

Read carefully, the artifacts tell a coherent story, and it is worth being precise about what each one supports. *The file system establishes* timing and mechanism. The Master File Table and the USN change journal recorded the document, having been previously created on another system and copied to the OneDrive Documents folder at 15:32:07 UTC, written in a single operation in roughly three milliseconds, followed within seconds by reparse-point changes that are the signature of OneDrive converting the new file into a cloud placeholder. The same journal recorded the later rewrite at 15:41:44 UTC. These are *system-generated records* of what happened to the file and when.

The AI agent and PowerShell logs establish agency. The OpenClaw session log (`bd500b9f-ff43-4642-bb79-5e7fcfd6770e.jsonl`) preserved the human instruction and the agent's step-by-step execution, including the save. The Windows PowerShell log independently recorded the same python-docx operations against the document's full path, along with a save-stage process running in the seconds before the recorded save in an EventID600 record. Two independent sources, the application's own log (the OpenClaw session log) and an independent operating-system record (the Windows PowerShell event log), corroborate one another to the second; neither depends on trusting the other.

The Office and OneDrive artifacts establish what came next. The Microsoft Office alerts log (OAlerts.evtx) recorded a somewhat unusual EventID 300, recording the Word process starting one second after the launch command. The Word jump list (AppID fb3b0dbfee58fac8), instead of recording the typical entry showing the document being opened, instead recorded an atypical launch of the Word application with the document's OneDrive cloud URL (`https://d.docs.live.net/7c7baf0cc732173d/Documents/Protein-Integration-with-Alien-Tech.docx`) in its arguments field, with an "interaction count" of 2. *It is my opinion the "interaction count" represents first, the PowerShell/python interaction of the summary paragraph insertion and saving of the document, and second, the subsequent opening of the document.* OneDrive's own databases and reparse records showed the file being tracked and synchronized at both the creation and the save. Taken together, the available *artifacts are consistent with* a single account: the preexisting document was copied to the file system in OneDrive, *modified programmatically* by the OpenClaw agent acting on a recorded instruction, then opened in Word for display, while stored on and synchronized through the cloud.

Notice the language. “The system recorded.” “The file was modified.” “The application started.” “The available artifacts are consistent with.” That precision is not decoration. It is the difference between describing evidence and overstating it.

What the Artifacts Do Not Prove

A careful examiner is defined as much by what they decline to assert as by what they conclude, and this test offers several useful examples. *The traditional proof-of-opening artifacts were never created.* No LNK shortcut in the user’s Recent folder referenced the document. The NTUSER.DAT registry hive contained not a single entry for the target document. There were no jump list entries for the document as a user-opened file, no RecentDocs or OpenSavePidMRU values, no Office reading location, no file MRU entry. An examiner who relied only on those artifacts, the very artifacts most of us reach for first, would have found nothing and might have reported that the document was never opened. That conclusion would have been false.

Just as important, the artifacts do not establish that a human being personally authored the new text. They establish that an instruction was recorded in the AI agent’s session log and that the agent carried it out. They do not, by themselves, tell us who issued that instruction, whether that person foresaw what the agent would do, or whether the agent’s interpretation matched the operator’s intent. The analysis was careful on a related point: the specific text written during the save, and the process that originally created the document, were not established by the materials provided. The file’s creation at 15:32:07 UTC appears in the timeline as a process unresolved. Holding that line, declining to fill the gap with assumption, is exactly the discipline the SAIDI era will demand.

Why AI Activity Complicates Attribution

The deeper lesson of the test is that the same end state, a created or modified Word document sitting on a computer or found in a user’s personal cloud folder, can arise from very different actors, and each leaves a different signature. As examiners, we increasingly have to distinguish among at least four possibilities.

The first is activity performed directly by a human user, working through Windows Explorer and Microsoft Word. This is the case our traditional artifacts were built to capture, and it generates the LNK files, jump lists, MRU values, and reading locations we know well. The second is activity performed by ordinary software acting under user direction: scripts, scheduled tasks, macros, backup and sync clients. These leave their own traces, often in logs and the file system, but not always in the user-interface artifacts.

The third is activity performed by an AI-enabled tool or agent. In the test, this looked like python-docx writing a document through a non-interactive PowerShell process, leaving rich agent and process logs but none of the shell artifacts of human use. A different agent might drive the Word application directly, leaving artifacts that look far more human. There is no

single signature for agentic activity, which is precisely the problem. The fourth, and the most dangerous, is activity that only appears user-driven because the resulting artifacts landed in familiar places. A file in OneDrive, an Office cache entry, a registry key, a fresh timestamp in a user's profile: each of these feels like evidence of personal action, and each can be produced by automation the user never consciously directed. The location of an artifact is not proof of the actor behind it. Only temporal analysis of the whole can tell the story of what happened and give context to the actions/evidence.

A defense built on this reality will not be exotic. It will simply ask the question we should be asking ourselves: which of these four types of activity does the evidence actually support, and which has the examiner merely assumed?

Why AI Inside Our Own Tools Deserves Caution

If AI complicates the evidence, it is also arriving, quickly, inside the tools we use to examine that evidence. Nearly every major forensic vendor is now attaching, integrating, or marketing artificial intelligence: natural-language search, automated artifact summarization, suggested timelines, conversational interfaces layered over case data. Some of this will be useful. AI that surfaces a relevant artifact in a terabyte of data or drafts a first-pass timeline an examiner then ***validates***, can save real time.

But there is a difference between a tool that helps an examiner look and a tool that purports to tell the examiner what to conclude, and we must not blur it. **AI will and does fail**. How many variations of "*I apologize for the confusion*," "*You're absolutely right*," or "*I made an error, thank you for pointing that out*," have you heard while using your AI of choice? **AI will hallucinate** artifacts, relationships and events that are not there. It will summarize a chat thread or a registry hive and quietly omit the line that mattered. It will miss context that any experienced examiner would have caught. Most troubling, it may steer an analyst confidently toward the inculpatory reading of an artifact while failing to detect or flag the exculpatory artifact sitting beside it. An AI summary that announced, "the user opened and edited this document" would have been wrong in this very test, and wrong in the direction that does the most harm.

The point is not whether AI has a place in forensic tools. It does. The question is whether AI can be trusted as a forensic analyst. *It cannot*. The examiner remains responsible for every word, every finding, and every assertion in the report and testimony. A tool's confidence or reliability is not the examiner's validation. That distinction is not a technicality; it is the foundation of forensic integrity. Recently, a longtime friend asserted an idea over lunch that AI could simply take the witness stand, testify and be subjected to cross-examination. I admit the image was equal parts amusing and thought-provoking. Imagine: a laptop or phone perched in the witness booth, microphone pulled to the speaker, voice feature engaged, ready to testify. But the U.S. legal system does not find it amusing. Federal Rule of Evidence 603, titled "Oath or Affirmation to Testify Truthfully," requires that before testifying, a witness must give an oath or affirmation to testify truthfully, in a form designed to impress that duty on the witness's conscience. Perjury by a witness is a crime under 18 U.S.C. § 1621. AI cannot qualify, since AI has no conscience which could be bound by a sense

of honor or of legal accountability for falsehood. Without accountability, there is no testimony, only output.

The Risk of Button-Pushing Forensic Practice

We have to be honest about this profession as it actually operates, not as we would describe it in a brochure. Having worked around lawyers daily for the last 20 years, I rarely make definitive statements. I joke sometimes that I even give myself wiggle room when I tell my wife that "generally" I love her. But let me break my rule and say, "All examiners are overworked, under-resourced, and buried in backlog." Devices keep getting larger, cases keep getting more numerous, and the pressure to clear work is constant. Under that pressure, a quiet erosion has already occurred. Many examiners have been reduced to pushing buttons: running the tool, reading the parsed output, and reporting what the tool displays. Sadly, some attorneys have even uttered to me, "well, it's good enough." Words I thought after working 40 years in law enforcement I would never hear cross the lips of an attorney.

For years, I have seen far too often when a tool says an artifact is not present, the source is not always checked. The SQLite database, the registry hive, the ESE database, the raw log, the underlying record, these often go uninspected. When a keyword search returns nothing, the absence is taken as proof that the term is not there, rather than as a prompt to ask whether the search reached the right data in the right encoding. This is the validation problem, and it exists today, before AI has fully arrived. That history should make us skeptical of an easy promise. If we are not consistently validating tool output now, what reason is there to believe validation will improve when a vendor offers an AI feature that appears to do the analyst's thinking for them (an offer which sounds like that of a snake oil salesman)? The more capable and confident the tool appears, the stronger the temptation to defer, and the higher the cost when it is wrong.

AI as Triage, Not as Truth

None of this is an argument against the technology. The right way to think about AI in our work is as an instrument of triage, not a source of conclusions. For the initial review of a large evidence set, AI can be valuable. It can help prioritize artifacts, surface candidate leads, cluster related items, suggest where to look first, and get an examiner onto a productive trail faster than manual review alone. In a discipline where the volume of data routinely exceeds the time available, that assistance is worth having.

The boundary is simple to state and must be held without exception: AI should help the analyst *find the truth*; it must never replace the analyst's responsibility to determine it. A lead is a hypothesis to be tested, not a finding to be reported or a box we try to make the evidence fit. An AI-generated summary is a starting point for examination, not a substitute for it. The moment a tool's output is copied into a report without independent confirmation,

triage has quietly become a conclusion, and the examiner has ceded the one thing that cannot be delegated: judgment.

Validation, Source Review, and a Return to Old-Fashioned Analysis

The answer to all of this is not to reject the technology. It is to recover a discipline that predates it. Old-fashioned investigative analysis, understanding the artifacts, examining the sources, testing competing explanations. These are the things that the SAIDI era requires. That means understanding what an artifact actually is and how it is generated, so that its presence or absence can be interpreted correctly. It means validating findings, manually or with a second, independently developed tool, before they go into a report. It means always going to the source: opening the SQLite database, the registry hive, the journal, the log, when the finding matters, rather than trusting a parser's summary. It means maintaining a keen eye for exculpatory artifacts as deliberately as inculpatory ones. And it means testing competing hypotheses, human action, automation, synchronization, cloud behavior, application activity, and now AI-agent activity, rather than settling on the first explanation that fits.

The OpenClaw test rewards exactly this approach. The traditional artifacts were absent, but the file system journal, the event logs, the OneDrive databases, and the AI agent's own session files preserved a detailed, corroborated account of what occurred. An examiner who stopped at the missing LNK files would have learned nothing. An examiner who went to the source learned almost everything that could responsibly be known, including where the evidence ran out.

Practical Recommendations

The following practices are not new, but the arrival of AI agents and AI-enabled tools makes them non-negotiable.

- Preserve and examine the original source artifacts; do not rely on a tool's parsed output as if it were the source.
- Maintain forensic images, validate them with hashing, and conduct all analysis on verified working copies.
- Validate every material finding manually or with a second, independent tool before it enters a report.
- Review the underlying SQLite databases, registry hives, ESE databases, logs, and file system records whenever a finding carries real weight.
- Do not treat a keyword search that returns nothing as proof that the term is absent; confirm the search reached the right data, in the right encoding.
- Document both what the tool reported and what you independently confirmed, and keep the two distinct.

- Look deliberately for exculpatory artifacts, not only inculpatory ones.
- Treat AI-generated summaries and suggested leads as hypotheses to be tested, never as conclusions to be reported.
- Test competing explanations for the same evidence: direct user action, scripted automation, synchronization, cloud activity, application behavior, and AI-agent activity.
- State the limits of the evidence plainly in both reports and testimony.
- Avoid "the user did X" unless the artifacts support it; prefer precise, record-based language such as "the system recorded," "the file was modified," "the application created," "the account was used," or "the available artifacts are consistent with."
- Have a technical Peer Review of all reports of findings.

Conclusion: The Analyst's Duty Is to Find the Truth

The purpose of digital investigative analysis has never been to produce an impressive report, to confirm an investigator's theory, or to ratify whatever a tool displays on screen. The purpose is to **find the truth through analysis that is defensible, repeatable, and validated**. AI will not change that purpose. It will only test our commitment to it.

The SAIDI defense is coming, and in many cases it will deserve a serious answer, because AI agents really do act, really do misfire, and really do leave behind artifacts that can mislead. We meet that defense not by fearing the technology and not by surrendering to it, but by doing the work: understanding our artifacts, validating our findings, examining our sources, weighing the alternatives, and stating honestly what the evidence does and does not establish. A person's liberty, livelihood, clearance, and reputation may rest on our analysis. None of those should ever rest on a single unvalidated artifact, a tool-generated summary, or the confident output of a machine that cannot be put under oath. AI can help us find the truth. It cannot be allowed to decide it. That responsibility is, and must remain, the analyst's.

Author's Note

The views, opinions, and analysis expressed in this article are solely those of the author and do not reflect, represent, or constitute the official position, policy, or endorsement of the author's employer or of any agency or organization with which the author is affiliated. The controlled test described herein was conducted independently for research and educational purposes. Nothing in this article constitutes legal advice or an official statement on behalf of any entity.

This article is intended to encourage validation, critical thinking, and defensible digital investigative analysis. It is not an argument against the responsible use of AI or automation in forensic work. Throughout, observations described as facts reflect artifacts recorded on that system, while statements of forensic implication and professional opinion are identified as such.