IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 9, September 2026

DOI: 10.17148/IJARCCE.2026.15906

A Unified Analysis of Digital Forensics and Machine Learning in Cybercrime Investigation

Muzamil Amin[1], Yatham Thirumalanayudu[2]

Assistant Professor, CSE Cyber Security & Data Science, Guru Nanak Institutions Technical Campus, Hyderabad, India[1,2]

Abstract: The rapid increase in cybercrime and the growing volume of digital evidence have made digital forensics an essential component of modern cyber investigations. Digital forensic tools play a crucial role in identifying, acquiring, preserving, analyzing, and presenting digital evidence from computers, mobile devices, cloud platforms, networks, and Internet of Things (IoT) environments. Over the years, numerous commercial and open-source forensic tools have been developed to address a wide range of investigative requirements. Meanwhile, machine learning (ML) techniques have emerged as effective approaches for improving the efficiency of forensic analysis through automated evidence classification, malware detection, anomaly detection, user behavior analysis, and event reconstruction. This paper presents an analysis of digital forensic tools, and their functions, and the application of machine learning techniques to digital evidence analysis throughout the investigation process. The study also examines the application of different ML algorithms across forensic domains based on the characteristics of the available evidence and extracted features. In addition, the study experimentally evaluates selected forensic tools and proposes a generalized architecture for integrating digital forensic tools with machine learning. The comparative analysis highlights how ML can assist in addressing challenges related to data volume, complexity, timeline analysis, evidence correlation, and analysis efficiency. The findings demonstrate that the integration of digital forensic tools with machine learning can enhance the efficiency and scalability of forensic investigations while maintaining the need for evidence validation and expert interpretation.

Keywords: Digital Forensics, Cybercrime Investigation, Digital Forensic Tools, Machine Learning, Digital Evidence, Forensic Analysis, Evidence Classification.

I. INTRODUCTION

In this modern digital era, where electronic gadgets and the Internet are essential in personal and professional domains, digital forensics has emerged as a fundamental aspect of current investigations [1]. It includes the retrieval, examination, and presentation of evidence preserved in digital formats, which is essential in matters pertaining to cybercrime, corporate espionage, fraud, intellectual property theft, and data breaches [2]. As people become increasingly dependent on digital communication, e-commerce, and online platforms, the significance of digital forensics has markedly increased, rendering it crucial for law enforcement, corporate security, legal proceedings, and civil disputes [3]. Digital forensics encompasses a broad range of technical domains [4], including the recovery of deleted files, the investigation of cyberattacks, and the analysis of data stored on mobile devices, cloud platforms, and network systems. Fig. 1 illustrates the forensic domains with their function. The importance of digital forensics extends beyond solving cybercrimes to protecting organizations from financial losses, data breaches, and reputational damage. By identifying, collecting, and preserving digital evidence, investigators can prevent further criminal activities and provide reliable evidence in legal proceedings. As cyber threats become increasingly sophisticated and society grows more dependent on digital infrastructures, digital forensics has emerged as an indispensable discipline for ensuring security, accountability, and the effective administration of justice [5].The increasing use of intelligent media technologies and the acceptance of social networking platforms have brought new problems to digital forensics, especially in addressing the growing dissemination of misinformation and disinformation [6]. The integration of information from multiple online platforms often blurs the distinction between credible and misleading content, making it increasingly difficult to differentiate factual information from false narratives during forensic analysis [7]. This has led to a global consensus on the necessity for sophisticated forensic methods to tackle the intricacies of misinformation and its impact on cyber governance and societal trust.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 9, September 2026

DOI: 10.17148/IJARCCE.2026.15906

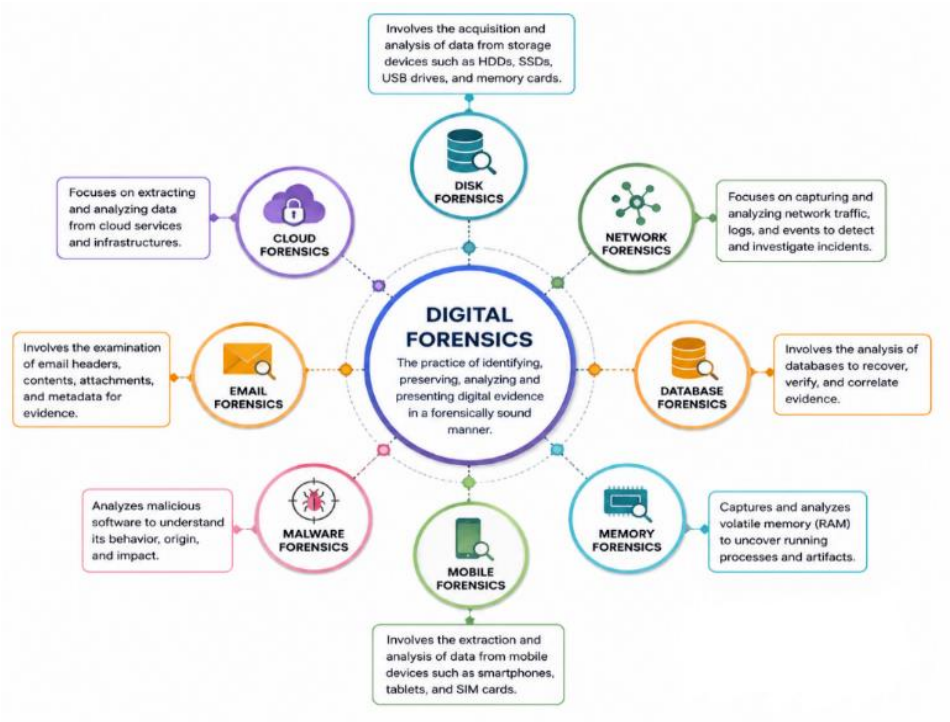

Fig. 1 Digital forensic domains and their functions.

II. DIGITAL FORENSICS AND MACHINE LEARNING ALGORITHMS

A. Digital Forensics

It is a field of forensic science that deals with the identification, preservation, acquisition, examination, analysis and presentation of digital evidence in a forensically acceptable manner. This is the systematic gathering and analysis of data from electronic devices and information systems for use in civil and criminal investigations. Digital forensics is the application of scientific methods and forensic techniques to ensure the authenticity and integrity of digital evidence during an investigation. The main goal of digital forensics is to reconstruct the events, determine facts and provide answers to the fundamental investigative questions of who, what, where, when, how and why. Such information helps investigators, law enforcement agencies and judicial authorities to identify the precise conditions of an event and to submit trustworthy evidence in court.

1) The Digital Forensic Investigation Process

A digital forensic investigation begins as soon as a security incident or crime is reported. Once potential sources of evidence have been identified, investigators methodically acquire relevant digital artifacts from the devices, systems, and other items involved in the incident [8]. The process of digital forensics is shown in Fig. 2.

Identification: The identification phase determines the scope of the investigation by identifying potential sources of digital evidence which may include devices, storage media, networks, cloud services, and user accounts that are relevant to the incident.

Preparation: Preparation involves setting up the investigative environment by documenting case details, preparing forensic tools, logging, and establishing the chain of custody for legal compliance of the investigation.

Strategy: This is the phase in which you decide on your forensic strategy; selecting the right acquisition methods, deciding on the tools you need, planning the procedures for collecting evidence and documenting the overall investigation methodology.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 9, September 2026

DOI: 10.17148/IJARCCE.2026.15906

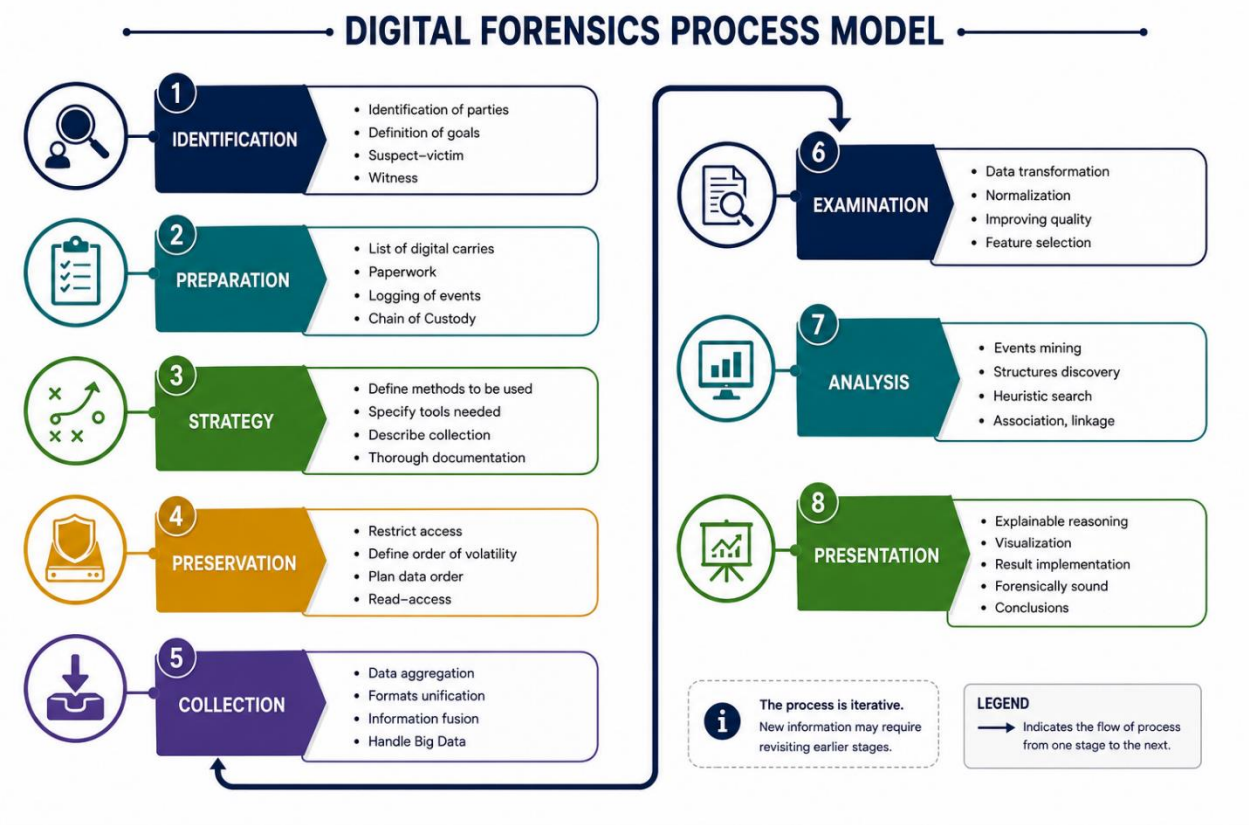

Fig. 2 Digital forensic investigation process model

Preservation: The preservation phase protects the integrity of digital evidence by preventing any changes to the evidence through the use of forensic imaging, write-blockers, secure storage, and documentation of a continuous chain-of-custody.

Collection: The collection stage is where digital evidence is acquired from the sources identified in the preceding phase, employing forensically sound procedures that ensure the authenticity of the data, while minimizing the potential for alteration or loss.

Examination: The examination phase involves processing and organizing the evidence collected through data extraction, normalization, recovery and transformation to prepare the relevant artifacts for detailed forensic analysis.

Analysis: The analysis step consists of interpreting the evidence examined to reconstruct events, establish relationships, detect anomalies, and reach conclusions that support the goals of the investigation.

Presentation: The presentation phase involves communicating the forensic findings through comprehensive reports, visual aids and expert explanations, ensuring the results are accurate, comprehensible and legally acceptable.

B. Machine Learning

Machine Learning (ML) is a prominent subfield of Artificial Intelligence (AI), where systems learn from data, recognize patterns and make predictions or decisions without being explicitly programmed for every task [9]. ML models, using data-driven algorithms [10], can automatically classify information, find complex patterns, and derive useful insights from large and heterogeneous data. Machine learning is increasingly being used as a valuable tool in digital forensics, aiding in the automation of evidence analysis, malware detection, anomaly identification, cyberattack attribution, file classification, user behavior analysis, and the detection of manipulated digital content. Such capabilities greatly enhance the accuracy, efficiency and scalability of forensic investigations, allowing investigators to process large volumes of digital evidence more effectively.

Machine learning is generally divided into four major paradigms: supervised learning, unsupervised learning, semi-supervised learning and reinforcement learning. In supervised learning, models are trained on labeled datasets to define a mapping between input features and respective outputs. Classification and regression are the most commonly used

DOI: 10.17148/IJARCCE.2026.15906

techniques [11]. One of the main approaches, clustering, is used in unsupervised learning to analyze unlabeled data and to uncover hidden structures, relationships and patterns [12]. Semi-supervised learning makes use of a small amount of labeled data and a larger amount of unlabeled data to improve the performance of the model and reduce the large-scale manual annotation [13]. Reinforcement learning is a method that allows an agent to learn the best actions by interacting with the environment in a continuous process where the agent maximizes the total reward and minimizes the penalties, proving to be suitable for sequential decision-making and adaptive problem solving [14].

III. LITERATURE REVIEW

A. Parveen et al. [15] have discussed various digital forensic tools. They have particularly focused on software forensic tools to detect forged digital images. The various features of five software forensic tools namely, FotoForensics, JPEGsnoop, Ghiro, Forensically and Izitru have been evaluated. The authors have observed that the selected tools do not provide any information about the basic concepts which have been used in the tools for detection of the forged images.

Raff et al. [16] introduced MalConv, an end-to-end deep learning framework that directly operates on raw executable files for malware detection without manual feature engineering. The proposed convolutional neural network has shown that raw binary data can be processed effectively to distinguish malicious from benign software, thus improving the automation and efficiency of malware forensic analysis.

Ghabban et al. [17] present a comparative study on Network Forensic Tools and Network Forensic Processes. Four tools (Xplico, OmniPeek, NetDetector and NetIntercept) are evaluated focusing on the detection, collection and analysis of the network incidents. They conclude that Xplico is better than the others.

Valluvar et al. [18] evaluate and contrast complimentary forensic software, including Autopsy, FTK Imager, ProDiscover Basic, and Wireshark, focusing on network investigation, data scrutiny, and password decryption. The assessment criteria encompass platform compatibility, file system compatibility, imaging functionalities, data-centric features, reporting functionalities, hash type compatibility, types of attacks, resource consumption, and pattern recognition skills. The research indicates that Autopsy, FTK Imager, and ProDiscover Basic possess distinct advantages and drawbacks in data analysis. The authors determine that John the Ripper and Hashcat excel at password cracking due to their extensive hash type compatibility. Wireshark is advised for the examination of networks.

Bayar and Stamm [19] suggested a constrained convolutional neural network for the detection of image manipulations in digital forensic investigations. The proposed architecture, different from traditional CNNs, suppresses the image content and highlights the forensic artifacts, which allows for the efficient detection of operations like median filtering, resampling, and contrast enhancement. Experimental results demonstrated superior performance over traditional handcrafted feature-based methods.

Ghazinour et al. [20] compared eleven digital forensic tools based on their functionality regarding different file systems, decryption techniques and incident response capabilities. The study emphasized the importance of selecting an appropriate forensic tool for effective evidence collection and processing in digital investigations. The tools evaluated provided useful features for forensic professionals but the authors noted that they mostly targeted traditional forensic settings and lacked new forensic fields or advanced analytical capabilities.

Horsman [21] considered the reliability and authenticity of digital forensics tools in the forensic inquiry process. The paper evaluated the performance of forensic tools in the collection of evidence, file system analysis, and artifacts analysis. It highlighted that the dependability of forensic software is an important issue in the credibility and admissibility of digital evidence. The author stressed the need for rigorous testing, standardized validation protocols, and accreditation systems to ensure that forensic tools produce accurate, consistent, and legally acceptable results.

Sachdeva et al. [22] have presented a comparative study of various commercial and open-source digital forensic tools for the forensic domains. The study assessed an array of instruments according to their platform compatibility, license structure, purpose and core features to support researchers in choosing appropriate tools for different research environments. The authors concluded that no one forensic tool can fulfill all investigative requirements and emphasized the significance of continuous improvements in forensic software to accommodate emerging digital technologies, operating systems, and cyber threats.

Cozzolino et al. [23] proposed a convolutional neural network that translates traditional residual-based forensic descriptors into a deep learning architecture for image forgery detection. The model was able to successfully detect the

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 9, September 2026

DOI: 10.17148/IJARCCE.2026.15906

manipulated images with higher accuracy and robustness compared to traditional handcrafted forensic techniques, thereby demonstrating the effectiveness of CNN-based feature learning for multimedia forensics.

IV. CONTRIBUTION

Digital forensics increasingly requires the integration of diverse forensic tools and intelligent analytical techniques to address the growing volume and complexity of digital evidence. In this context, this research provides a systematic and practical review of digital forensic tools and their integration with machine learning. The primary contributions of this research are as follows:

- Cross-domain analysis of digital forensic tools in terms of database, network, disk/computer, mobile, cloud, multimedia and memory forensics with a focus on their functions, versions, licensing and platform compatibility [20], [22].
- Practical evaluation of selected forensic tools to demonstrate their practical capabilities in evidence acquisition, artifact extraction, integrity verification, activity reconstruction, live-system triage, file identification and memory analysis.
- Structured analysis of machine learning techniques in digital forensics, mapping ML algorithms to their forensic applications, domains, investigation phases, advantages, and limitations.
- A Generalized architecture for integrating digital forensic tools with machine learning, connecting forensic collection and acquisition, evidence preservation, artifact extraction, preprocessing, feature engineering, ML-based analysis and prediction, forensic interpretation, and investigation reporting.

V. METHODOLOGY

The aim of this study is to investigate the role of digital forensic tools and machine learning techniques in modern cybercrime investigation and digital forensic analysis. The methodology is presented in two consecutive phases as shown in Fig. 3. The first phase deals with the identification and analysis of the major digital forensic domains and the tools related to each domain. The second phase is about the application of machine learning methods in different areas of digital forensics and compares the algorithms based on standard evaluation criteria.

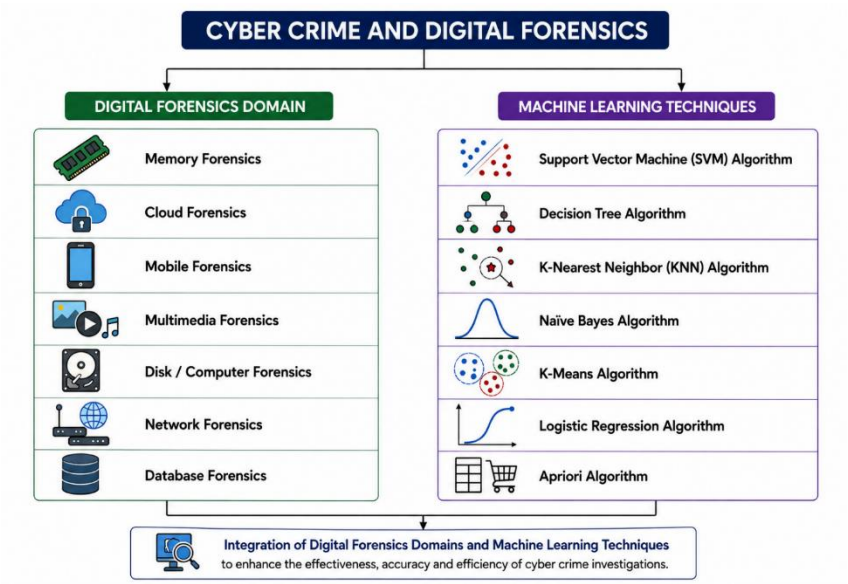

Fig. 3 Methodology framework of the study.

A. Analysis of Digital Forensic Domains and Tools

Digital forensics consists of several specialized areas, each focusing on analysis of specific types of digital evidence and computing environments. The growth in diversity and quantity of digital evidence has resulted in the creation of specialized forensic tools to assist investigators in acquiring, preserving, examining, analyzing and reporting the evidence. These tools vary in their supported platforms, evidence formats, forensic applications, and capabilities. The major digital forensic domains and the tools normally used in these domains are discussed in this paper. The analysis includes memory forensics, cloud forensics, mobile forensics, multimedia forensics, disk/computer forensics, network forensics and database forensics. The analysis in each domain discusses the relevant forensic tools, their primary

DOI: 10.17148/IJARCCE.2026.15906

functions, supported platforms, and licensing models, providing a comparative overview of the current digital forensic tool landscape.

1) Database forensics

Database Forensics is a branch of computer forensics that deals with the investigation of databases stored in physical storage. This includes retrieving data from database files, but also examining transactions (committed and rolled-back), failures, and backups that are part of routine database activity [24],[26]. The main aim of Database Forensics is to establish the existence of a data security breach and if yes, then to what extent the attack has been committed. It also involves database operations monitoring and database restoration that are deleted [27]. Databases often contain sensitive information such as bank account details, transaction records, server logs, health-related information and user IDs and passwords. Thus, Database Forensics is an important tool for investigating data breaches in organizations. Similarly to computer forensics, Database Forensics investigates database server timestamps, metadata, and cache data. Artifact analysis is also important to establish correlations between different timestamps. For a Database Forensics investigation, the investigator gathers evidence from multiple sources, which include volatile database data, transaction logs, system event logs, database files, server error logs, trace files and audit logs [26]. These artifacts enable investigators to reassemble database activity and detect unauthorized or suspicious operations.

TABLE I: OVERVIEW OF COMMERCIAL AND OPEN-SOURCE DATABASE FORENSIC TOOLS

Tool Name	Current Version	Supported Database	Open Source / Paid	Trial Version	Platform
Belkasoft Evidence Center	10.0	SQLite	Paid	Yes (30-day trial)	Windows
ElcomSoft Distributed Password Recovery	5.0	Oracle	Paid	Yes (Free demo)	Windows
Intella	3.2	SQLite	Paid	Yes (30-day trial)	Windows
Cellebrite Physical Analyzer	7.68	SQLite	Paid	Yes (Demo available)	Windows
DBR	21.0	MySQL, Oracle, SQL Server	Paid	Yes (30-day trial)	Windows
DB3F	—	SQLite3x	Open Source	N/A	Windows
DBCarver	—	SQL Server, Oracle, MySQL, PostgreSQL, SQLite	Open Source	N/A	Windows
SQLite Forensic Reporter	1.2	SQLite	Open Source	N/A	Windows
Digital Detective Blade	2.0	SQLite	Paid	Yes (14-day trial)	Windows
DCFLDD	1.4.1	SQLite	Open Source	N/A	Windows, Linux
WFT (Windows Forensic Toolchest)	3.2.0	SQLite	Open Source	N/A	Windows
ApexSQL Recover	2024.01.0129	SQL Server	Paid	Yes (15-day trial)	Windows
Stellar Repair for MS SQL	11.0	SQL Server	Paid	Yes (Demo available)	Windows

2) Network forensics

Network Forensics (NF) is a branch of digital forensics that deals with the collection, analysis and recovery of network traffic and information exchanged among network nodes in an effort to obtain evidence about criminal activities. It can be used to extract information such as emails, file transfers, chat conversations and web activities from network traffic as a part of forensic investigations. Network forensics is focused on live analysis, but offline analysis is also possible by taking a physical memory dump from the relevant devices. Packet analyzers can be used to capture, identify and analyze system logs, network logs and network events during live investigations [28]. Data about the network, such as open

DOI: 10.17148/IJARCCE.2026.15906

connections, active ports, visited web pages, cookies, and server- and client-side logs, may also reside in physical memory. The information can aid forensic analysis even after the device has been unplugged. Network forensics can help to monitor and investigate anomalous traffic, intrusion attempts, active and passive attacks, phishing, spamming, and Denial-of-Service (DoS) attacks. With the continuous growth of the network infrastructures, the number and types of network attacks are increasing as well. It becomes difficult for forensic investigators to analyze large numbers of network packets. To address this problem, packet-analysis methods have been designed to minimize the network data and facilitate the detection of the nature and origin of network attacks [28]. Generally, network forensic investigation can be performed in two ways: capturing and analyzing individual packets, and capturing packets sequentially for later analysis, which is often called Deep Packet Inspection (DPI) [29]. However, the use of DPI may have legal and regulatory consequences, especially when network monitoring is associated with surveillance and national investigations, and may therefore require the approval of appropriate governmental authorities [30]. Moreover, the implementation of multiple levels of encryption [31] in technologies such as TOR browsers, Virtual Private Networks (VPNs) and proxy servers, may obscure the real origin and destination of network communications. Covert communication methods can introduce an additional layer of complexity to network forensic analysis by hiding communication channels and network activities [32].

TABLE II: LIST OF NETWORK FORENSIC TOOLS

Tool Name	Current Version	Supported Protocols / Network Data	Open Source / Paid	Trial Version	Platform
TCPdump	4.99.6	IP, UDP, DHCP, DNS, ICMP, TCP and others	Open Source	N/A	Linux, Unix, Windows*
Wireshark	4.6.7	Broad protocol support	Open Source	N/A	Windows, Linux, macOS
NetDetector	—	TCP, UDP, SCTP, IP, DNS, ICMP, HTTP, HTTPS, SSL/TLS, SMB, GTP, SIP, RADIUS, FTP, SSH	Paid	Contact vendor	Appliance / Alpine-based
Xplico	1.2.2	HTTP, HTTPS, SIP, IMAP, POP, SMTP, TCP, UDP, IPv6, RTP, IRC, FTP, TFTP and others	Open Source	N/A	Linux
Metasploit	5.0.0-2026051301**	Network services and security protocols	Open Source / Paid	Yes	Windows, Linux, macOS
Nessus	10.12.0***	Network vulnerability/security assessment	Paid	Yes	Windows, Linux, macOS
Wikto	—	HTTP/Web traffic analysis	Freeware	N/A	Windows
NetworkMiner	3.1	HTTP, HTTP/2, FTP, TFTP, SMB/SMB2, SMTP, POP3, IMAP, SSH, TCP, UDP and others	Open Source / Paid	Free version available	Windows, Linux
Ngrep	1.45	IPv4, IPv6, TCP, UDP, ICMP, Ethernet and others	Open Source	N/A	Linux, Windows, Unix-like
Nmap	7.99	TCP, UDP, IP, ICMP, SCTP, FTP, DNS, SNMP, DHCP and others	Open Source	N/A	Windows, Linux, macOS
Wireless Network Watcher	2.44	Network/device traffic monitoring	Freeware	N/A	Windows
Snort	3.12.2.0	TCP, UDP, ICMP, ARP, IP and others	Open Source	N/A	Windows, Linux
Zeek	8.2.1	HTTP, HTTPS, SMTP, SSH, SSL, DNS, FTP, TCP, UDP, ICMP	Open Source	N/A	Linux, macOS

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 9, September 2026

DOI: 10.17148/IJARCCE.2026.15906

3) Computer forensics

Disk/Computer Forensics is a fundamental area of computer forensics that deals with the examination of data stored on digital storage devices [33]. Modern storage devices comprise hard disk drives (HDDs), solid state drives (SSDs), secure storage drives (SEDs), optical disks, magnetic tapes, RAID systems, flash drives, memory devices, and Blu-ray disks [34]–[35]. Disk forensics is important because of the increasing popularity of personal storage devices and USB drives that can be a large source of digital evidence. The main aim of disk/computer forensics is to find, retrieve, and collect useful forensic evidence from digital storage devices while maintaining the integrity of the original evidence. Modern disks, however, can hold huge amounts of information, making manual inspection difficult. Thus, machine learning methods such as topic modeling, clustering, and data classification have been investigated for helping to identify relevant information and to automate parts of the forensic analysis process [36].

Forensic datasets may have a lot of high-dimensional and unstructured data. Because of this, the classical classification methods may not be efficient in extracting the relevant information. Forensic data can also be analyzed using Natural Language Processing (NLP) techniques such as Named Entity Recognition (NER), relation extraction, and Part-of-Speech (PoS) tagging to extract useful information. A further significant challenge for disk/computer forensics is the increasing use of anti-forensic techniques. Deleted data can often be recovered with forensic recovery software, but file shredding, data overwriting, drive erasing, and disk wiping can be used to deliberately destroy or conceal evidence. Attackers can also hide malicious code or other evidence in disk slack, RAM slack, and Master File Table (MFT) slack. Such covert artifacts can be challenging to detect by conventional forensic tools.

Bit-by-bit disk imaging is a method for creating a complete copy of a storage device for forensic examination. It copies all contents of the disk from the beginning to the end, including hidden areas such as MBR and MFT slack. However, full disk image creation requires a lot of disk space, and disk errors or bad sectors may impact the acquisition process. Disk forensic investigators are most likely to encounter FAT, exFAT, NTFS, HFS, HFS+, HPFS, UFS, Ext2, Ext3, Ext4, XFS, Veritas, VMFS, ZFS, and Reiser file systems.

TABLE III: OVERVIEW OF COMMERCIAL AND OPEN-SOURCE DISK/COMPUTER FORENSIC

Tool Name	Current Version	Primary Evidence / Function	Open Source / Paid	Trial Version	Platform
SANS SIFT Workstation	Rolling Build (Ubuntu 22.04 LTS Base, cast v1.x)	Disk images, file systems, memory and digital evidence	Open Source / Free	N/A	Linux, Windows (WSL)
ProDiscover Suite	—	Disk imaging, live-disk preview, file-system and digital-evidence analysis	Paid	Demo available	Windows
The Sleuth Kit (TSK)	4.15.0	Disk images, file systems, deleted files and forensic artifacts	Open Source	N/A	Windows, Linux, macOS
FTK Imager	4.7.x	Forensic imaging, evidence acquisition, preview and verification	Freeware	N/A	Windows
GNU dd	9.10	Bit-by-bit disk imaging and data duplication	Open Source	N/A	Linux, Unix-like
CAINE	14.0 Lightstream	Forensic acquisition, disk imaging and analysis	Open Source / Free	N/A	Linux
Oxygen Forensic Detective	18.3	Disk/partition imaging, evidence acquisition and digital-evidence analysis	Paid	15-day free trial	Windows
DEFT Linux	8.2 (Legacy)	Forensic acquisition, disk imaging and digital investigation	Open Source / Free	N/A	Linux
PALADIN Forensic Suite	Version 9	Disk imaging, hashing, file recovery, logical imaging and forensic triage	Free / Open Source components	N/A	Linux
LastActivityView	1.37	Windows activity artifacts, including Registry, Event	Freeware / Proprietary	N/A	Windows

DOI: 10.17148/IJARCCE.2026.15906

		Logs, Prefetch, Minidump, and related system sources			
Autopsy	4.23.1	Disk images, hard drives, mobile devices, file systems, and forensic image formats	Open Source / Free	N/A	Windows, Linux, macOS

4) Multimedia forensics

Multimedia forensics is an important field of digital forensics (DF) since documents, audio, video, and images are increasingly used in digital communication. These multimedia files may include personal, business, or financial information. Therefore, it is crucial to protect and analyze the multimedia data for its security and authenticity [37]. Multimedia forensics is a specialized area of data forensics. It uses scientific methods to analyze multimedia content such as audio, video, and images. The main goals are to identify the source device, recover the useful information, and verify the integrity of the recovered data [38]. The main topics of multimedia forensics are document audio-video-image counterfeiting, document audio-video-image manipulation, and image carving [39], and manipulated images [40]. Advanced encryption and morphing algorithms for multimedia files can extend the time needed for forensic examination. This topic raises an important question: "Can we trust what we see?" [41] Several techniques have been proposed to detect video forgery, including deepfake detection, pixel-motion-based methods, inter-frame analysis, and intra-frame analysis. With the development of advanced image morphing techniques, it has become more and more difficult to identify the original image. Government agencies and forensic experts have a very difficult time determining whether images are real or fake, where they came from, and who owns them, especially in cases involving child pornography.

Although deep neural networks have been used to develop many forensic techniques, attackers might try to fool Convolutional Neural Networks (CNNs) if they know the forensic algorithms used to detect forgeries [42]. Multimedia forensic tools are designed to detect attacks that alter the integrity of multimedia files. Some of these tools use deep learning methods, e.g., the Fast Gradient Sign Method (FGSM). Multimedia forensic systems are, however, vulnerable to a variety of attacks, including universal attacks, laundering attacks, and surrogate detector-based attacks. These attacks can change images in a way that the introduced changes are difficult to perceive.

TABLE IV: COMPARATIVE ANALYSIS OF MULTIMEDIA FORENSIC TOOLS

Tool Name	Current Version	Supported Media	Open Source / Paid	Trial Version	Platform
ExifTool	13.40	Images, videos, audio, documents	Open Source	N/A	Windows, macOS, Linux
Foremost	1.5.7	Images, documents, multimedia files	Open Source	N/A	Linux, Unix, Windows
Scalpel	2.0	Images, documents, multimedia files	Open Source	N/A	Linux, Windows
Ghiro	0.2.0	Images	Open Source	N/A	Linux
iZotope RX	11.4.0	Audio	Paid	Trial/Demo available	Windows, macOS
Phonexia Voice Inspector	5.2.0	Audio/voice	Paid	Demo available	Windows, Linux
Amped Five	Latest release	Video	Paid	Demo/Trial available	Windows
Amped Authenticate	Update 40960	Images, video	Paid	Demo/Trial available	Windows
Cognitech Tri-Suite	Latest release	Images, video	Paid	Demo available	Windows
VideoInspector	Latest release	Video	Freeware	N/A	Windows
FTK	8.2	Images, video, audio, documents and other digital evidence	Paid	Trial/Demo available	Windows

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 9, September 2026

DOI: 10.17148/IJARCCE.2026.15906

FotoForensics	Web service	Images	Free / Commercial options	N/A	Web
Palantir	Commercial platform	Documents and multimedia evidence	Paid	Contact vendor	Windows/Linux
Analyst's Notebook	Latest release	Documents and investigative data	Paid	Trial/Demo available	Windows
Foxton Forensics – Browser History Capturer (BHC) / Browser History Examiner (BHE)	BHC: 1.4.5; BHE: 1.23.4	Chrome, Microsoft Edge, Firefox, Internet Explorer 10/11, Safar	BHC: Free; BHE: Paid	BHC: Free; BHE: Paid	Windows

5) Mobile forensics

Mobile Forensics (MoF) is a sub-branch of digital forensics that deals with the identification, collection, extraction and documentation of digital evidence from mobile and other small electronic devices using scientific methods [25]. It includes devices like smartphones, electronic gadgets, PDAs, GPS devices, tablets, gaming consoles and messaging applications. Mobile forensics can be further categorized into fields such as Android forensics, iOS forensics, smartwatch forensics, GPS forensics, drone forensics, and social media forensics, which involves investigating various social media applications. Mobile devices are very important in digital investigations because many people access the internet using smartphones and other mobile devices. Smartphones have become an indispensable part of everyday life. Hence, they contain a lot of useful information such as addresses, bank card details, personal photos, emails, official documents, contacts, and messages [43]. Phones can also hold data on where a person travels and what they do during the day. This information can create a unique pattern of an individual's life. If a mobile device is lost, stolen or accessed by an unauthorized person, sensitive personal information could be compromised. This can be problematic for privacy issues and can lead to identity theft.

The vast variety of mobile devices and operating systems presents another significant challenge in mobile forensics. Different manufacturers use different hardware, firmware and software, which can lead to compatibility and interoperability problems. Even devices that share the same operating system can have different implementations, file systems, and data structures. These differences make forensic investigations more difficult for forensic analysts and mobile forensic tool developers [44].

There are many different manufacturers producing mobile devices. This results in the development of several Mobile Forensics (MoF) tools by using different investigation techniques. These tools are intended to assist in forensic investigations and enhance cooperation of forensic investigators with law enforcement agencies [25]. However, there still exists a need for a universal mobile forensic tool that can effectively support the examination of different types of mobile devices and operating systems.

TABLE V COMPARISON OF MOBILE FORENSIC TOOLS

Tool Name	**Current Version (2026)**	**Primary Function**	**Open Source / Paid**	**Trial Version**	**Platform**
Cellebrite Inseyets	Spring 2026 Release	Mobile acquisition, extraction, decoding and analysis	Paid	Contact vendor	Windows
MSAB XRY	11.5	Logical, physical and file-system extraction	Paid	Contact vendor	Windows
MSAB XRY Pro	11.5	Advanced mobile extraction, RAM acquisition and device access	Paid	Contact vendor	Windows
Magnet AXIOM	10.2.0.49217	Mobile acquisition, processing and analysis	Paid	Contact vendor	Windows
Magnet ACQUIRE	2.97.0.44221	iOS and Android mobile acquisition	Paid	Contact vendor	Windows

DOI: 10.17148/IJARCCE.2026.15906

Oxygen Forensic Detective	18.3	Mobile extraction, decoding, analysis and reporting	Paid	Demo/Trial	Windows
Belkasoft X	2.11	Mobile acquisition, artifact extraction and analysis	Paid	Trial available	Windows
MOBILedit Forensic	9.8	Mobile-phone and smartwatch acquisition and analysis	Paid	Trial/Demo	Windows
Magnet Verakey	Current / continuously updated	Consent-based iOS/Android logical and full-file-system extraction	Paid	Contact vendor	Windows
Elcomsoft Mobile Forensic Bundle	Discontinued	Mobile evidence acquisition/recovery	Paid	—	Windows
DEFT Linux	Legacy	Mobile and computer forensic acquisition	Open Source / Free	N/A	Linux

6) Cloud forensics

Cloud Forensics (CF) is the application of digital forensics (DF) to cloud computing and can be viewed as a subset of network forensics. It is the combination of the field of cloud computing and digital forensics. The main goal of cloud forensics is to acquire digital artifacts from remote servers, analyze them without unauthorized alteration or third-party intervention and document the evidence collected properly. The process involves different cloud actors namely the cloud broker, cloud provider, cloud auditor, cloud end user and cloud carrier. The challenges and existing solutions in cloud forensics are systematically studied in [45]. The studies described cloud forensic research and categorized different cloud forensic solutions. They also stress the importance of continually upgrading forensic tools to combat ever more sophisticated cybercrimes.

Cloud forensics can be considered a subset of network forensics as network forensics deals with forensic investigations across different kinds of networks such as private and public networks. Cloud computing requires extensive network access and applies many of the concepts of network forensic investigations, as well as additional techniques designed specifically for cloud environments. Cloud services run on servers that utilize virtual or physical hardware. Therefore, cloud environments typically use virtualization either through bare-metal implementations or hardware-assisted virtualization [46]. The forensic investigation process is also different on the basis of different cloud service models such as Platform-as-a-Service (PaaS), Software-as-a-Service (SaaS), Storage-as-a-Service (StaaS) and Infrastructure-as-a-Service (IaaS)[47]. Cloud forensics can also be studied at three levels: client side, network side and server side. This study is more interested in the major issues related to cloud forensics than traditional cloud computing. Dropbox, OneDrive and Google Drive and other cloud services may present challenges in obtaining evidence from during a cyberattack, due to Service Level Agreements (SLAs) and other restrictions.

TABLE VI: COMPARATIVE ANALYSIS OF CLOUD FORENSIC TOOLS

Tool Name	Current Version	Supported Cloud Service Model	Open Source / Paid	Trial Version	Platform
Digital Forensics Framework (DFF)	1.3.0 (Legacy)	IaaS	Open Source	N/A	Linux, Windows
FROST	Legacy / Research Prototype	IaaS (OpenStack)	Open Source / Research Tool	N/A	OpenStack / Ubuntu Linux
FTK (Forensic Toolkit)	8.2	IaaS	Paid	Trial/Demo available	Windows
NetworkMiner	3.1	IaaS, PaaS, SaaS	Free / Paid Professional	Free version available	Windows
X-Ways Forensics	21.8	SaaS	Paid	Demo available	Windows
Omnipeek	22.4.1	IaaS, PaaS, SaaS	Paid	Trial/Demo available	Windows
VNsnap	Not currently maintained/verified	IaaS	Open Source / Research Tool	N/A	Linux

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 9, September 2026

DOI: 10.17148/IJARCCE.2026.15906

7) Memory forensics

Memory Forensics (MeMF) is a subfield of digital forensics that addresses the acquisition and analysis of volatile memory from a computer system [48]. It is also known as RAM forensics, physical memory forensics, ROM forensics, and volatile memory forensics. Memory forensics is a field closely related to malware and ransomware investigations. Primary memory can contain useful evidence such as malicious processes, information about the attack, and keys for encryption or decryption. The main goal of MeMF is to recover and analyze relevant information and forensic artifacts from the primary memory and the data structures maintained by the operating system kernel [49]. Memory contains knowledge of the current state of the system and can be used as evidence of activities performed during or immediately prior to an attack, unlike persistent storage. It may contain data such as running processes, network configurations, user credentials, registry information, event logs, scheduled tasks, and Dynamic Link Libraries (DLLs). These artifacts can help investigators to identify the source of an attack and to understand how a system got compromised [50].

Memory forensics can be divided into three main steps: memory acquisition, memory analysis, and report generation. During memory acquisition, volatile memory is copied from the system using tools such as Belkasoft Live Capture, FTK Imager, and Mandiant. Once the memory image is obtained, it is analyzed using open-source and commercial forensic tools such as Volatility, Belkasoft, Bulk Extractor, Redline, and IDA Pro. The last step is to write a forensic report that explains what was found and how and when the system was compromised. Memory acquisition can be categorized into two approaches: Hardware Acquisition (HA) and Software Acquisition (SA). Hardware-based acquisition techniques often use PCI devices and Direct Memory Access (DMA) to access system memory. Software-based acquisition techniques acquire memory using kernel- or user-level mechanisms. Memory analysis can be done by static analysis and dynamic analysis. Static analysis is the analysis of the acquired memory image without actually interacting with the original system. Static analysis methods can include hash comparison with trusted third-party vendor values. Dynamic analysis is concerned with the activity of the system and may include integrity checking, process monitoring, and registry analysis.

TABLE VII: COMPARATIVE ANALYSIS OF MEMORY FORENSIC TOOLS

Tool Name	Version	Primary Function	Open Source / Paid	Trial Version	Platform
Volatility 3	2.28.0	Memory-dump analysis	Open Source	N/A	Windows, Linux, macOS
Belkasoft Live RAM Capturer	1.0	Live RAM acquisition	Free	N/A	Windows
FTK Imager	4.7.1	Evidence and memory acquisition	Free	N/A	Windows
Magnet RAM Capture	1.2.0	Live RAM acquisition	Free	N/A	Windows
Bulk Extractor	2.1.1	Artifact extraction from forensic images and memory	Open Source	N/A	Windows, Linux, macOS
Belkasoft X	2.11	Memory and digital-evidence analysis	Paid	Yes	Windows
IDA Pro	9.3	Reverse engineering and malware analysis	Paid	Evaluation available	Windows, Linux, macOS
CrowdResponse	1.0.5.0	Windows host artifacts: processes, registry, files, Prefetch, scheduled tasks, YARA/memory artifacts	Freeware / Proprietary	N/A	Windows

B. Experimental Analysis Of Digital Forensic Tools

To evaluate the practical applicability and forensic capabilities of digital forensic tools, controlled experiments were conducted across multiple forensic domains. The evaluation included evidence acquisition, recovery and analysis of artifacts, and reliability of forensic evidence acquired. The selected tools covered browser artifact analysis, Windows Registry and boot-time analysis, disk imaging and cloning, file and folder access-history analysis, network traffic analysis, live incident-response triage, file-type identification, and live memory (RAM) forensics. The tools evaluated were Browser History Capturer and Examiner, Process Monitor, FTK Imager, LastActivityView, CrowdResponse, and Autopsy. The experimental results provide a practical basis for the following comparison of forensic tools and evaluation of machine-learning-assisted forensic analysis.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 9, September 2026

DOI: 10.17148/IJARCCE.2026.15906

1. Browser Forensics using Foxton Forensics

Foxton Forensics' Browser History Capturer (BHC) and Browser History Examiner (BHE) were used to acquire and analyze web browser artifacts from a Windows 11 test system. BHC was configured to capture History, Cache, and Deleted History (recovered from Volume Shadow Copies) for the Edge and Internet Explorer/Edge-Legacy browsers, with the captured files preserved in their native format at a designated destination folder as shown in Fig 4. This preserves forensic soundness, as the original artifact structure is retained for independent verification rather than being altered during acquisition.

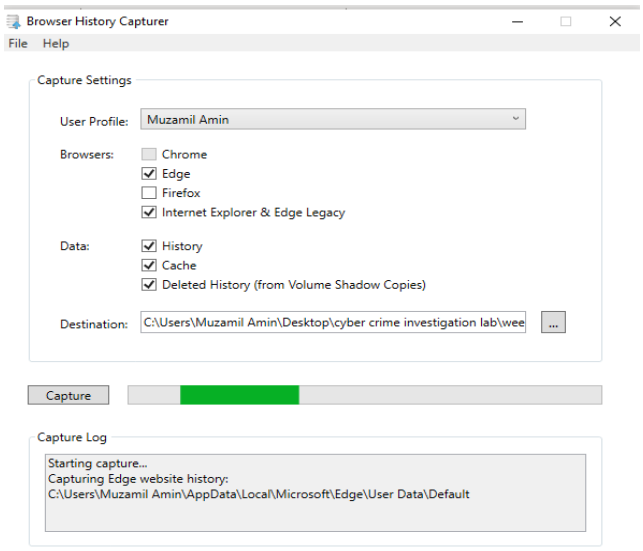

Fig. 4 Browser History Capturer configuration

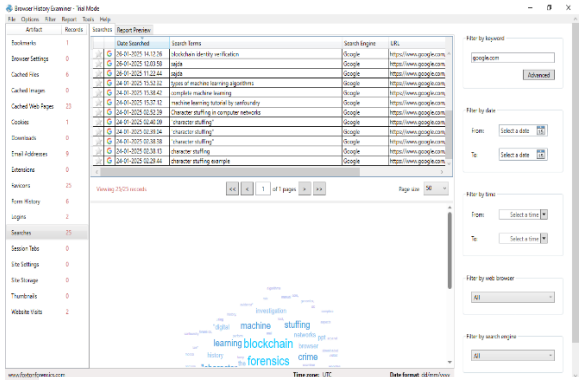

Fig. 5 Browser History Examiner search-artifact

The captured data set was subsequently loaded into Browser History Examiner as shown in Fig 5, which parsed and indexed the artifacts into eighteen distinct categories, including Bookmarks, Cached Files, Cached Web Pages, Cookies, Downloads, Email Addresses, Form History, Logins, Searches, Session Tabs, Site Storage, Thumbnails, and Website Visits. In the experimental run, the Searches artifact alone yielded 25 recoverable entries spanning a range of search engines and timestamps, and the Website Visits artifact returned corresponding domain-level visit records.

The experimental analysis showed that the acquisition and analysis processes were separated into distinct tools, with Capturer used for acquisition and Examiner used for analysis, thereby limiting the risk of the analysis process altering the source artifacts. Deleted browser history was recoverable through Volume Shadow Copy parsing. Furthermore, the eighteen artifact categories enabled a single acquisition to support multiple lines of forensic inquiry from the same dataset.

2. Disk Imaging and Cloning using FTK Imager

FTK Imager was used to perform forensically sound disk imaging and cloning of a source storage device. The workflow followed the standard acquisition sequence as shown in figure 6. Upon completion, FTK Imager computed MD5 and SHA-1 hash values for both the source and the resulting image and compared them in a verification report; the computed and report hash values matched exactly for both algorithms as shown in figure 7.

FTK Imager was then used to mount the created image via Image Mounting, producing a cloned logical volume shown in figure 8, and comparing the capacity values of the original and mounted clone confirmed structural equivalence.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 9, September 2026

DOI: 10.17148/IJARCCE.2026.15906

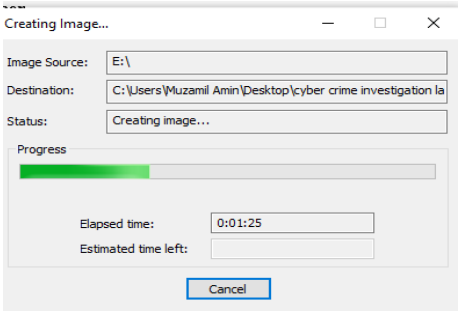

Fig. 6 FTK Imager disk imaging

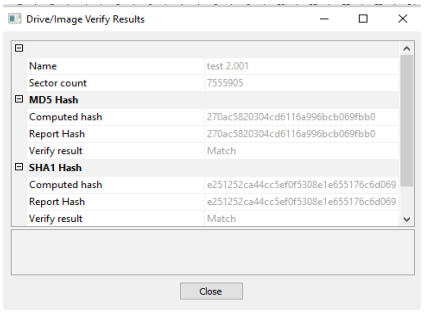

Fig. 7 FTK Imager hash verification

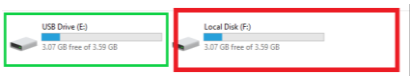

Fig. 8 Storage capacity comparison of source drive and mounted image.

The experimental analysis showed that cryptographic hash verification using MD5 and SHA-1 provided an objective and reproducible basis for demonstrating evidentiary integrity. Capturing case metadata at the time of acquisition directly supported chain-of-custody requirements. Furthermore, image mounting allowed the verified image to be examined or cloned without modifying the original evidence device.

3. File and Folder Access History using LastActivityView

LastActivityView was used to reconstruct a history of file, folder, and program-execution activity on a Windows test system by aggregating multiple low-level Windows data sources Prefetch files, jump lists, MRU (most-recently-used) registry lists, and application-execution records into a single unified timeline. Following file/folder interactions performed on the test system (opening a document, launching browsers, running system utilities), the tool was run to scan the system and populate its activity table.

The resulting output listed 4,028 discrete actions, each entry recording an Action Time, a Description, the Filename, Full Path, associated file extension, and a Data Source field identifying the underlying artifact shown in figure 9.
The experimental analysis showed that a single scan aggregated 4,028 actions from multiple underlying Windows artifact stores, demonstrating the value of consolidating disparate low-level records into a single investigator-readable timeline. Prefetch files served as the dominant corroborating data source, linking the recorded activities to independently verifiable on-disk evidence.

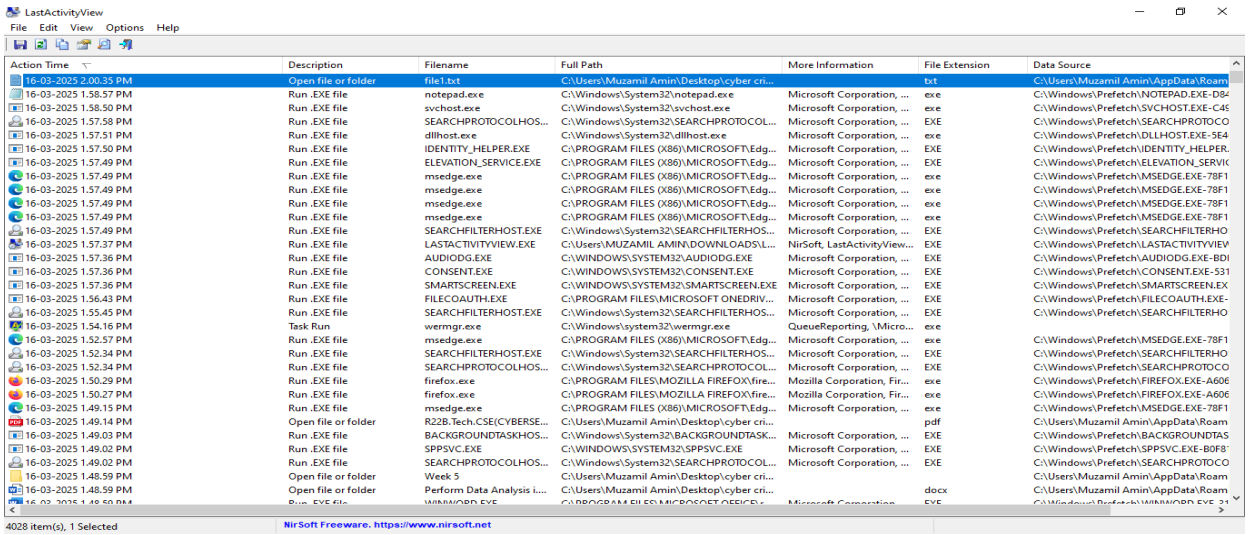

Fig. 9 LastActivityView reconstructed activity timeline.

4. Incident-Response Triage using CrowdResponse

CrowdResponse, a lightweight, portable incident-response utility, was used to demonstrate rapid, installation-free forensic data collection from a live Windows system. The tool exposes a set of modular collectors including PSList (running processes), RegDump (registry dump), Prefetch, SuperFetch, Shim (AppCompatCache), Tasks, Drivers, and

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 9, September 2026
DOI: 10.17148/IJARCCE.2026.15906

Yara (malware pattern matching) — each invocable independently from the command line and configurable to emit output in JSON, CSV, or XML.

Two modules were exercised experimentally. The PSList module (CrowdResponse.exe -o C:\process_info.xml @PSList) produced a structured XML enumeration of running processes shown in figure 10, including PID, parent PID, session ID, and 64-bit/admin flags, together with host metadata such as hostname, domain, OS version, and IPv4/IPv6/MAC addresses. The RegDump module (@RegDump against the Run key) produced an XML dump of the Run key's contents shown in figure 11.

Fig. 10 CrowdResponse PSList output of running

Fig. 11 CrowdResponse RegDump output of Registry Run

The experimental analysis showed that a single portable executable, requiring no installation, collected both process-state and registry evidence directly into a machine-readable XML format, making it suitable for time-constrained live-system triage.

5. File Type Detection using Autopsy

Autopsy, the graphical front end to The Sleuth Kit, was used to perform signature-based file-type detection on a forensic disk image. A new case was created with case number, examiner, and description metadata recorded for chain-of-custody purposes, and a sample.E01 image was added as a data source. During ingest, the File Type Identification module was enabled alongside Hash Lookup and Keyword Search, and the image was processed to build a searchable case database. Once processing completed, the File Views > File Types view categorized the recovered content by detected signature rather than by file extension alone shown in figure 12. In the experimental run, the Images category returned 36 files, each with associated Modified, Change, Access, and Created (MAC) timestamps and file size, viewable individually alongside hex, text, and metadata panes for verification of true file type against any claimed extension.

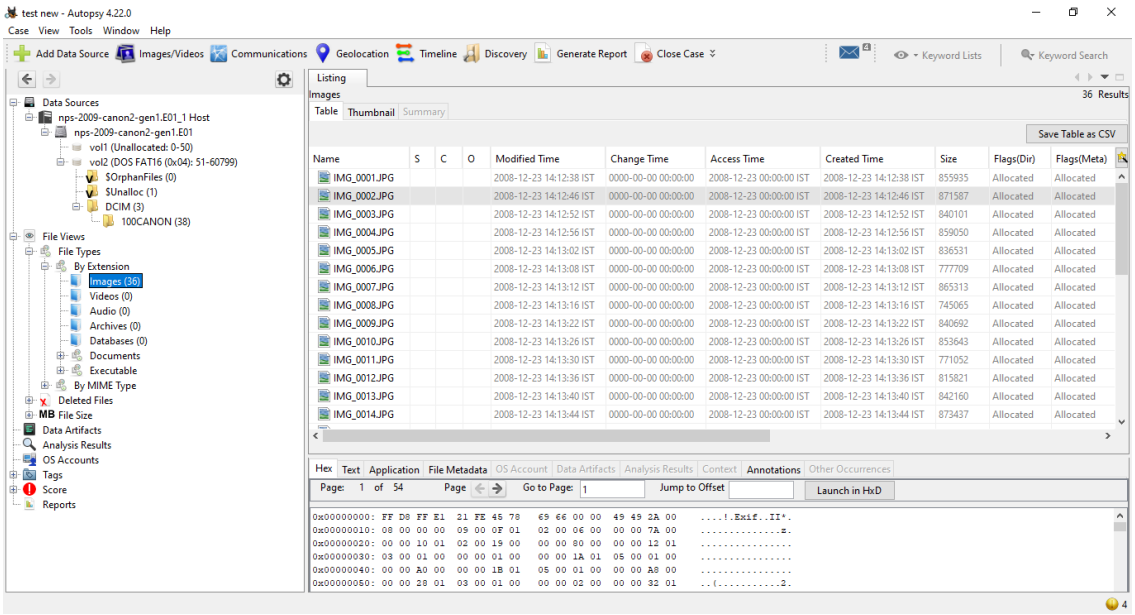

Fig. 12 Autopsy file-type classification results.

DOI: 10.17148/IJARCCE.2026.15906

The experimental analysis showed that signature-based classification using **"magic number" detection** was essential for identifying files that had been deliberately renamed to conceal their actual file types. The ingest-module architecture, with File Type ID, Hash Lookup, and Keyword Search operating concurrently, enabled a single pass over the forensic image to support multiple analytical objectives. Furthermore, the per-file MAC timestamps recovered during file-type triage provided useful information for subsequent timeline analysis.

VI. MACHINE LEARNING IN CYBER CRIME AND DIGITAL FORENSICS

The application of machine learning has become more prevalent in cybersecurity and digital forensics, making it easier in the analysis of large and complex datasets originating from various digital environments. Digital forensic investigators can utilize machine learning algorithms to evaluate extensive datasets gathered from cloud environments and interconnected systems [51]. The analyzed information can help pinpoint user behaviors and identify patterns that may indicate suspicious or criminal activities. This section outlines several machine learning algorithms that facilitate the discovery and analysis of digital evidence, thus enhancing the overall efficiency of the digital forensic investigation process.

A. Support Vector Machine Algorithm in Cyber Crime and Digital Forensics

Islam et al. proposed a method for detecting copy-move and splicing attacks in color images using Local Binary Pattern (LBP) and Discrete Cosine Transform (DCT) operators. The LBP and DCT operators capture the differences in the local frequency distribution and recognize the micro-patterns in images. It considers inter-cell values of LBP blocks and constructs feature vectors. These features are classified by a Support Vector Machine (SVM) with Radial Basis Function (RBF) kernel to discriminate between authentic and tampered images. The experimental results indicate that the proposed approach is suitable for image forgery detection, and promising classification performance is achieved [52].

Ferreira et al. proposed a method for the distinction of genuine and manipulated digital photographs and videos using DFT based feature extraction technique and SVM classifier. The extracted features were used to build an SVM based classification model which was then used to classify test images. We have developed a set of Python programs to process image features, extract frames from videos and build the SVM classification model. The proposed approach has been implemented on two Autopsy modules for forensic analysis of images and videos by DFT-SVM method. The results showed that the proposed approach required less analysis time than convolutional neural network (CNN)-based analysis [53].

B. Decision Tree Algorithm in Cyber Crime and Digital Forensics

Chhabra et al. [54] proposed an architectural framework using the MapReduce framework, Hadoop Distributed File System (HDFS) and a Decision Tree algorithm for processing large volumes of network data. The framework includes several steps: network traffic capturing, converting captured traffic to a human-readable format, filtering packets, analyzing for malicious activities, and presenting threat analysis and visualization results. The Decision Tree algorithm was used to classify the network traffic as malicious and non-malicious which improved the efficiency and accuracy of the analysis process. The reported results showed that the proposed framework detected the 99% of malicious and non-malicious network traffic [54].

Usman et al. presented a hybrid approach to address the challenges of IP reputation systems by integrating machine learning, dynamic malware analysis, and cyber threat intelligence. The proposed approach utilizes big data forensics to forecast the probability of an attack and then classify the attack on the basis of its behavioral features. The proposed system was compared with existing IP reputation systems using various machine learning algorithms such as Decision Tree (DT), Support Vector Machine (SVM) and Naïve Bayes (NB). Decision Tree algorithm showed better performance in recall, F-measure and precision among techniques evaluated this shows that this algorithm is efficient in behavioral attack classification [55].

C. Logistic Regression Algorithm in Digital Forensics

Ali et al. studied malware that targeted the Windows Registry and studied how different malware types interacted with the registry locations during digital forensic investigations. The study evaluated various machine learning classifiers, such as Neural Network, Decision Tree and Logistic Regression, to determine if modified registry timestamps could be used together with machine learning techniques for forensic analysis. The researchers identified 47 registry locations that were commonly targeted by malware. Of the evaluated approaches, the Boosted Tree classifier correctly classified more than 72% of the malware samples, indicating its ability to aid investigators in fast identifying the presence and type of malware during forensic analysis. [56]

DOI: 10.17148/IJARCCE.2026.15906

Del Mar-Raave et al. proposed a machine-learning-based digital forensic tool to automatically classify images acquired during forensic investigations. The study included pre-trained deep learning models into the development lifecycle of a digital forensic tool, and evaluated InceptionV3, Xception, ResNet, and VGG16 models for image classification. The chosen model was then integrated into a prototype forensic tool for handgun image identification. The study showed that pre-trained models in open source can be integrated in digital forensic tools to automate evidence classification and help investigators in processing a large amount of seized digital content [57].

TABLE VII: SUMMARY OF MACHINE LEARNING ALGORITHMS IN DIGITAL FORENSICS INVESTIGATION

Focused Area	ML Algorithm	Forensic Type	DF Phase	Advantage	Disadvantage
Image forgery detection [52]	SVM	Multimedia Forensics	Analysis	Effective classification of authentic and manipulated images using extracted image features	Performance depends on feature quality and training data
Multimedia evidence classification [53]	SVM	Multimedia Forensics	Analysis	Can be integrated into forensic applications such as Autopsy for automated analysis of manipulated images and videos	Feature extraction and model performance are dataset-dependent
Windows Registry malware identification [56]	Decision Tree / Boosted Tree	Disk/Computer Forensics	Analysis	Provides interpretable classification of malware based on Registry activity	Performance depends on selected Registry features and training samples
File-system crime classification [57]	Enhanced Multiclass SVM	Disk/Computer Forensics	Analysis	Supports multiclass classification of file-system activity associated with digital crimes	Requires appropriate parameter selection and representative training data
Forensic artifact classification [58]	K-Nearest Neighbor (KNN)	Computer Forensics	Analysis	Classifies artifacts based on similarity to previously labeled forensic evidence	Computational cost increases with large datasets
Forensic evidence classification [59]	Naïve Bayes	Computer Forensics	Analysis	Computationally efficient for large collections of textual or categorical forensic evidence	Assumes conditional independence among features
Forensic evidence clustering [60]	K-Means	Computer Forensics	Analysis	Groups similar forensic artifacts without predefined classes	Requires selection of the number of clusters and may be sensitive to initialization
Security-log analysis [61]	Logistic Regression	Network/Log Forensics	Analysis	Provides interpretable probability-based classification of security events	Primarily models linear relationships between features
Mobile forensic behavior analysis [62]	K-Means + Apriori	Mobile Forensics	Analysis	Combines clustering and association-rule mining to identify recurring behavioral patterns	Results depend on clustering and association parameters

DOI: 10.17148/IJARCCE.2026.15906

Microsoft 365 forensic analysis [63]	K-Means + Apriori	Cloud Forensics	Analysis	Identifies activity groups and associations among security events	Results depend on clustering parameters and association thresholds
Computer-forensic classification [64]	SVM, KNN, Naïve Bayes, Logistic Regression, Decision Tree	Computer Forensics	Analysis	Enables comparative evaluation of multiple ML approaches	No single algorithm performs optimally for every forensic dataset
Fake-image detection [65]	Logistic Regression, Decision Tree, KNN	Multimedia Forensics	Analysis	Enables automated classification of genuine and manipulated images	Performance depends on feature quality and dataset diversity
Temporal metadata analysis [66]	Learning Classifier System	Disk/Computer Forensics	Analysis/Triage	Generates interpretable rules and reduces the forensic search space	Requires appropriate temporal features and representative training data

VII. INTEGRATION OF DIGITAL FORENSICS WITH MACHINE LEARNING

The combination of Digital Forensics (DF) and Machine Learning (ML) provides a systematic way to automate and improve the analysis of large volumes of digital evidence. Conventional digital forensic investigations rely on forensic tools to acquire evidence and extract artifacts such as file-system information, metadata, logs, browser records, memory artifacts, and network data. However, the increasing volume and complexity of digital evidence make complete manual examination time-consuming. Machine learning can be incorporated after forensic acquisition and artifact extraction to process the resulting evidence, identify relevant features, recognize patterns, classify artifacts, detect anomalies, and prioritize evidence for further investigation. Thus, ML complements the forensic examination process rather than replacing it by helping investigators to identify potentially important evidence while preserving human interpretation and decision making.

The generalized architecture for integrating digital forensic tools with machine learning, presented in figure 13, illustrates this workflow.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 9, September 2026

DOI: 10.17148/IJARCCE.2026.15906

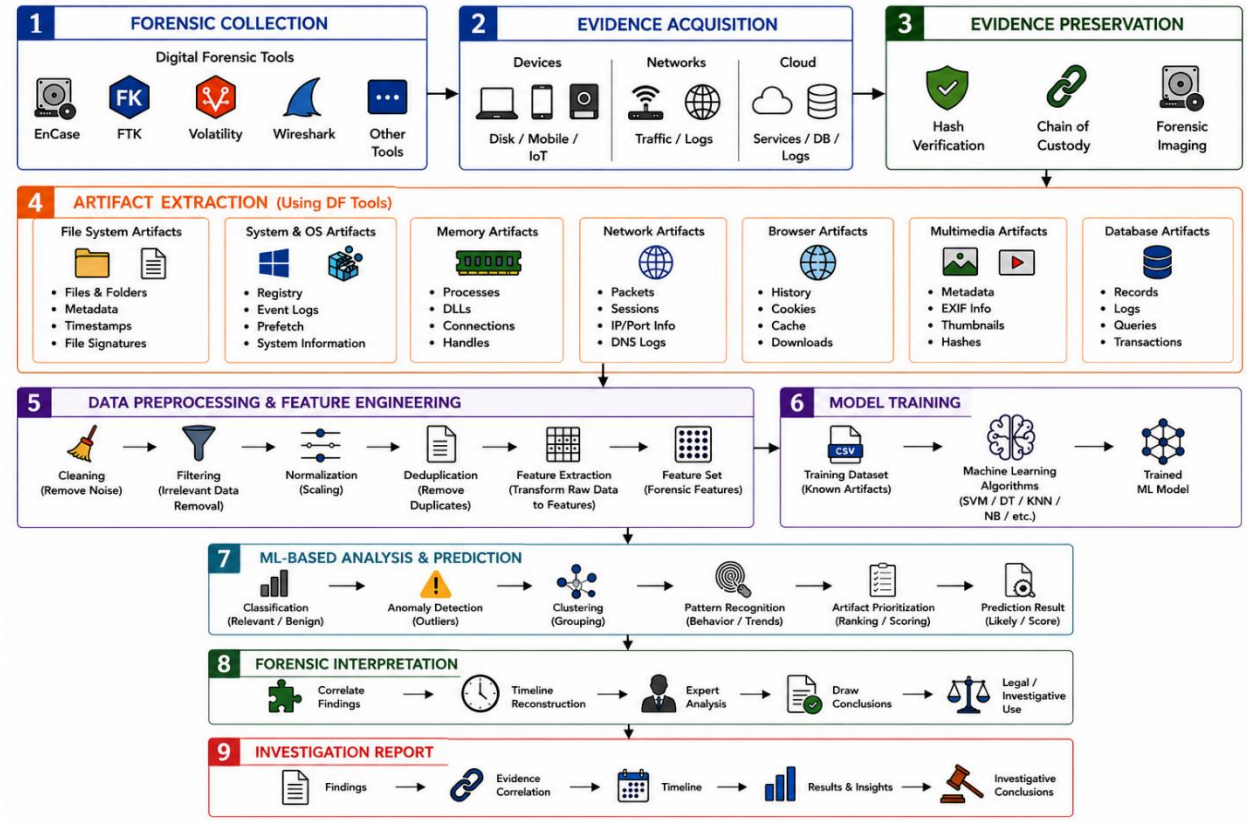

Fig. 13 Generalized Architecture for Integrating Digital Forensic Tools with Machine Learning

A practical implementation of this combination was proposed by Du and Scanlon [66]. They proposed a methodology for automated classification and prioritization of suspicious digital forensic artifacts. The approach integrates forensic artifact extraction and supervised machine learning in a Digital Forensics as a Service (DFaaS) framework. Raw disk images of seized Windows systems were processed using pytsk for filesystem metadata extraction and log2timeline (Plaso) for digital-event and timeline extraction. The resulting metadata and timeline information were combined to construct a dataset for machine learning. Features included file type, filename characteristics, file size, directory depth, creation and access timestamps, and the number of associated digital events. Known artifacts were filtered using hash comparison with a centralized database, and the resulting labeled artifacts were used to train Decision Tree, Gaussian Naïve Bayes, k-NN, SVM, and Logistic Regression models. The trained models were then used to predict whether the unseen artifacts are suspicious or relevant to the investigation. Among the evaluated algorithms, the Decision Tree performed the best overall, and precision, recall and F1-score values of 1.00 were achieved in the second dataset. The study shows how forensic acquisition and artifact-extraction tools can be combined with ML-based classification to automatically prioritize potentially relevant evidence, thus reducing the burden of manual examination [66].

VIII. DISCUSSION

The results from the different forensic domains and the experimental results in Section 6 show that no forensic tool can cater to all the needs of a digital investigation. This is in line with earlier comparative studies [15], [20], [22]. Thus, effective investigations require complementary tools in identification, acquisition, examination, and analysis phases. The experiments with Browser History Capturer/Examiner and FTK Imager/Volatility also demonstrate how the separation of acquisition and analysis can help in maintaining the integrity of evidence while providing multiple analytical views. The experimental results also confirm the importance of evidence integrity and validation. FTK Imager verified the integrity of the acquired disk image using MD5 and SHA-1 hash verification. Acquisition metadata facilitated evidence tracking and chain of custody requirements. This aligns with Horsman [21], who emphasized the need for forensic tool reliability and validation to generate trustworthy and legally defensible evidence.

As shown in Table VIII, the analysis of machine learning applications reveals that the choice of algorithms depends on the forensic task and evidentiary needs. Interpretable approaches such as decision trees, logistic regression, and naïve Bayes can provide more transparent decision-making, while the SVM and deep-learning approaches are more suitable

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 9, September 2026
DOI: 10.17148/IJARCCE.2026.15906

for more complex and high-dimensional tasks, such as multimedia and image-forgery analysis [52], [53], [52], [65]. It emphasizes the importance of a trade-off between predictive performance and interpretability in forensic applications. Machine learning can be more valuable in terms of supporting evidence triage, prioritization, classification, and pattern identification, instead of replacing forensic examination. Du and Scanlon demonstrated the potential of ML-based classification in prioritizing suspicious forensic artifacts [66]. However, the effectiveness of such an approach hinges on the availability of representative training data and appropriate feature selection. These limitations become especially relevant in emerging areas such as cloud, IoT, and multimedia forensics, where standardized forensic datasets are still limited.

The results show that digital forensic tools and machine learning have complementary roles in digital investigations. Forensic tools provide reliable ways of evidence acquisition and artifact extraction, while machine learning enables efficient analysis, classification and prioritization of forensic data. The combination of the two may improve the scalability and efficiency of digital forensic investigations while keeping the important role of expert validation and interpretation.

IX. CONCLUSION

This work demonstrates the complementarity of digital forensic tools and machine learning for digital forensic investigations. The cross-domain analysis and experimental evaluation show that no single forensic tool can address all investigative requirements, while machine learning can support the classification, prioritization, anomaly detection, and pattern analysis of large volumes of forensic data. The proposed generalized DF–ML integration architecture connects evidence acquisition and preservation with artifact extraction, preprocessing, feature engineering, ML-based analysis and prediction, forensic interpretation, and reporting. The results suggest that the integration of forensic tools with the suitable ML techniques may boost the efficiency and scalability of forensic investigations without compromising the validation of evidence and the expert interpretation. Future research will be directed toward expanding the integration of forensic tools with emerging forensic domains, creating representative forensic datasets, and assessing interpretable ML techniques to improve the reliability and practical application of ML-assisted forensic analysis.

REFERENCES

[1]. J. Singaram, S. S. Iyengar, and A. M. Madni, *Deep Learning Networks: Design, Development and Deployment*. Cham, Switzerland: Springer, 2023, doi: 10.1007/978-3-031-39244-3.
[2]. B. Shi and S. S. Iyengar, *Mathematical Theories of Machine Learning—Theory and Applications*. Cham, Switzerland: Springer, 2019, doi: 10.1007/978-3-030-17076-9.
[3]. C. Wang, S. S. Iyengar, and K. Sun, Eds., *AI Embedded Assurance for Cyber Systems*. Cham, Switzerland: Springer, 2023, doi: 10.1007/978-3-031-42637-7.
[4]. Y. Hariprasad, S. Lokesh, N. T. Sharathkumar, K. J. L. Kumar, C. Miller, and N. K. Chaudhary, "AI-ML analytics: A comprehensive investigation on sentimental analysis for social media forensics textual data," in *Intelligent Computing: Proc. 2023 Computing Conf.*, vol. 2, Cham, Switzerland: Springer, 2023, pp. 923–935, doi: 10.1007/978-3-031-37963-5_64.
[5]. G. S. Thejas, Y. Hariprasad, S. S. Iyengar, N. R. Sunitha, P. Badrinath, and S. Chennupati, "An extension of synthetic minority oversampling technique based on Kalman filter for imbalanced datasets," *Machine Learning with Applications*, vol. 8, Art. no. 100267, 2022, doi: 10.1016/j.mlwa.2022.100267.
[6]. Y. Hariprasad, K. J. L. Kumar, L. Suraj, and S. S. Iyengar, "Boundary-based fake face anomaly detection in videos using recurrent neural networks," in *Intelligent Systems and Applications*, Lecture Notes in Networks and Systems, vol. 543, Cham, Switzerland: Springer, 2023, pp. 155–169, doi: 10.1007/978-3-031-16078-3_9.
[7]. Y. Hariprasad, S. S. Iyengar, and N. Subramanian, "Deepfake video detection using lip region analysis with advanced artificial intelligence-based anomaly detection technique," *TechRxiv*, 2024, doi: 10.36227/techrxiv.172114952.25305640/v1.
[8]. C. Altheide and H. Carvey, *Digital Forensics with Open Source Tools*. Burlington, MA, USA: Syngress, 2011, doi: 10.1016/C2009-0-62460-0.
[9]. N. Sultana, N. Chilamkurti, W. Peng, and R. Alhadad, "Survey on SDN based network intrusion detection system using machine learning approaches," *Peer-to-Peer Networking and Applications*, vol. 12, no. 1–2, pp. 1–9, 2019, doi: 10.1007/s12083-017-0630-0.
[10]. D. Dhall, R. Kaur, and M. Juneja, "Machine learning: A review of the algorithms and its applications," in *Proc. ICRIC 2019: Recent Innovations in Computing*, Lecture Notes in Electrical Engineering, vol. 597, Cham, Switzerland: Springer, 2020, pp. 47–63, doi: 10.1007/978-3-030-29407-6_5.

DOI: 10.17148/IJARCCE.2026.15906

[11]. I. H. Sarker, A. S. M. Kayes, and P. Watters, "Effectiveness analysis of machine learning classification models for predicting personalized context-aware smartphone usage," *Journal of Big Data*, vol. 6, Art. no. 57, 2019, doi: 10.1186/s40537-019-0219-y.

[12]. A. Khraisat, I. Gondal, P. Vamplew, and J. Kamruzzaman, "Survey of intrusion detection systems: Techniques, datasets and challenges," *Cybersecurity*, vol. 2, Art. no. 20, 2019, doi: 10.1186/s42400-019-0038-7.

[13]. J. E. van Engelen and H. H. Hoos, "A survey on semi-supervised learning," *Machine Learning*, vol. 109, no. 2, pp. 373–440, 2020, doi: 10.1007/s10994-019-05855-6.

[14]. Z. Wang and T. Hong, "Reinforcement learning for building controls: The opportunities and challenges," *Applied Energy*, vol. 269, Art. no. 115036, 2020, doi: 10.1016/j.apenergy.2020.115036.

[15]. A. Parveen, Z. H. Khan, and S. N. Ahmad, "Classification and evaluation of digital forensic tools," *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol. 18, no. 6, pp. 3096–3106, Dec. 2020, doi: 10.12928/TELKOMNIKA.v18i6.15295.

[16]. E. Raff, J. Barker, J. Sylvester, R. Brandon, B. Catanzaro, and C. K. Nicholas, "Malware detection by eating a whole EXE," in *Proc. Workshops of the 32nd AAAI Conf. Artif. Intell.*, 2018, pp. 268–276.

[17]. F. M. Ghabban, I. M. Alfadli, O. Ameerbakhsh, A. N. AbuAli, A. Al-Dhaqm, and M. A. Al-Khasawneh, "Comparative analysis of network forensic tools and network forensics processes," in *Proc. 2021 2nd Int. Conf. Smart Comput. Electron. Enterprise (ICSCEE)*, Cameron Highlands, Malaysia, 2021, pp. 78–83, doi: 10.1109/ICSCEE50312.2021.9498226.

[18]. A. Valluvar, S. Shetty, S. Pandian, and S. Chaure, "Forensic tools in comparison: An assessment of performance across different parameters," *International Journal of Innovative Science and Research Technology*, vol. 8, no. 9, pp. 485–491, Sep. 2023, doi: 10.5281/zenodo.8363276.

[19]. B. Bayar and M. C. Stamm, "A deep learning approach to universal image manipulation detection using a new constrained convolutional layer," in *Proc. ACM Workshop Inf. Hiding Multimedia Security (IH&MMSec)*, Vigo, Spain, 2016, pp. 5–10, doi: 10.1145/2909827.2930786.

[20]. K. Ghazinour, D. M. Vakharia, K. C. Kannaji, and R. Satyakumar, "A study on digital forensic tools," in *Proc. 2017 IEEE Int. Conf. Power, Control, Signals and Instrumentation Engineering (ICPCSI)*, Chennai, India, 2017, pp. 3136–3142, doi: 10.1109/ICPCSI.2017.8392304

[21]. G. Horsman, "Tool testing and reliability issues in the field of digital forensics," *Digital Investigation*, vol. 28, pp. 163–175, Mar. 2019, doi: 10.1016/j.diin.2019.01.009.

[22]. S. Sachdeva, B. Raina, and A. Sharma, "Analysis of digital forensic tools," *Journal of Computational and Theoretical Nanoscience*, vol. 17, no. 6, pp. 2459–2467, 2020, doi: 10.1166/jctn.2020.8916.

[23]. D. Cozzolino, G. Poggi, and L. Verdoliva, "Recasting residual-based local descriptors as convolutional neural networks: An application to image forgery detection," in *Proc. ACM Workshop Inf. Hiding Multimedia Security (IH&MMSec)*, Philadelphia, PA, USA, 2017, pp. 159–164, doi: 10.1145/3082031.3083247.

[24]. O. M. Adedayo, *Reconstruction in Database Forensics*, Ph.D. dissertation, Dept. Computer Science, Univ. Pretoria, Pretoria, South Africa, 2015.

[25]. A. Al-Dhaqm, S. A. Razak, R. A. Ikuesan, V. R. Kebande, and K. Siddique, "A review of mobile forensic investigation process models," *IEEE Access*, vol. 8, pp. 173359–173375, 2020, doi: 10.1109/ACCESS.2020.3014615.

[26]. K. E. Pavlou and R. T. Snodgrass, "Forensic analysis of database tampering," *ACM Trans. Database Syst.*, vol. 33, no. 4, Art. no. 30, Nov. 2008, doi: 10.1145/1412331.1412342.

[27]. P. Frühwirt, M. Huber, M. Mulazzani, and E. R. Weippl, "InnoDB database forensics," in *Proc. 24th IEEE Int. Conf. Advanced Information Networking and Applications (AINA)*, Perth, WA, Australia, 2010, pp. 1028–1036, doi: 10.1109/AINA.2010.152.

[28]. L. F. Sikos, "Packet analysis for network forensics: A comprehensive survey," *Forensic Sci. Int.: Digital Investigation*, vol. 32, Art. no. 200892, 2020, doi: 10.1016/j.fsidi.2019.200892.

[29]. G. Shrivastava, "Network forensics: Methodical literature review," in *Proc. 2016 3rd Int. Conf. Computing for Sustainable Global Development (INDIACom)*, New Delhi, India, 2016, pp. 2203–2208.

[30]. G. A. Pimenta Rodrigues, R. de Oliveira Albuquerque, F. E. Gomes de Deus, R. T. de Sousa Jr., G. A. de Oliveira Júnior, L. J. García Villalba, and T.-H. Kim, "Cybersecurity and network forensics: Analysis of malicious traffic towards a honeynet with deep packet inspection," *Applied Sciences*, vol. 7, no. 10, Art. no. 1082, 2017, doi: 10.3390/app7101082.

[31]. M. R. Arshad, M. Hussain, H. Tahir, S. Qadir, F. I. Ahmed Memon, and Y. Javed, "Forensic analysis of Tor browser on Windows 10 and Android 10 operating systems," *IEEE Access*, vol. 9, pp. 141273–141294, 2021, doi: 10.1109/ACCESS.2021.3119724.

[32]. I. Makhdoom, M. Abolhasan, and J. Lipman, "A comprehensive survey of covert communication techniques, limitations and future challenges," *Computers & Security*, vol. 120, Art. no. 102784, 2022, doi: 10.1016/j.cose.2022.102784.

DOI: 10.17148/IJARCCE.2026.15906

[33]. D. Bem, F. Feld, E. Huebner, and O. Bem, "Computer forensics—Past, present and future," *Journal of Information Science and Technology*, vol. 5, no. 3, pp. 43–59, 2008.

[34]. G. Francis, "Data storage—Trends and directions," Preservation and Archiving Special Interest Group (PASIG), Stanford Libraries, 2011.

[35]. D. Quick and K.-K. R. Choo, "Big forensic data reduction: Digital forensic images and electronic evidence," *Cluster Computing*, vol. 19, no. 2, pp. 723–740, 2016, doi: 10.1007/s10586-016-0553-1.

[36]. L. Ignaczak, G. Goldschmidt, C. A. da Costa, and R. da Rosa Righi, "Text mining in cybersecurity: A systematic literature review," *ACM Computing Surveys*, vol. 54, no. 7, Art. no. 140, pp. 1–36, 2021, doi: 10.1145/3462477.

[37]. H. V. Zhao, M. Wu, Z. J. Wang, and K. J. R. Liu, "Forensic analysis of nonlinear collusion attacks for multimedia fingerprinting," *IEEE Transactions on Image Processing*, vol. 14, no. 5, pp. 646–661, May 2005, doi: 10.1109/TIP.2005.846035.

[38]. W. D. Ferreira, C. B. R. Ferreira, G. da Cruz Júnior, and F. A. A. M. N. Soares, "A review of digital image forensics," *Computers & Electrical Engineering*, vol. 85, Art. no. 106685, 2020, doi: 10.1016/j.compeleceng.2020.106685.

[39]. S. L. Garfinkel and M. McCarrin, "Hash-based carving: Searching media for complete files and file fragments with sector hashing and hashdb," *Digital Investigation*, vol. 14, Suppl. 1, pp. S95–S105, Aug. 2015, doi: 10.1016/j.diin.2015.05.001.

[40]. Y. Tang, J. Fang, K. P. Chow, S. M. Yiu, J. Xu, B. Feng, Q. Li, and Q. Han, "Recovery of heavily fragmented JPEG files," *Digital Investigation*, vol. 18, Suppl., pp. S108–S117, Aug. 2016, doi: 10.1016/j.diin.2016.04.016.

[41]. H. Sako, K. Y. Franke, and S. Saitoh, Eds., *Computational Forensics: 4th International Workshop, IWCF 2010, Tokyo, Japan, November 11–12, 2010, Revised Selected Papers*, Lecture Notes in Computer Science, vol. 6540. Berlin, Germany: Springer, 2011, doi: 10.1007/978-3-642-19376-7.

[42]. M. Barni, Z. Chen, and B. Tondi, "Adversary-aware, data-driven detection of double JPEG compression: How to make counter-forensics harder," in *Proc. IEEE Int. Workshop Information Forensics and Security (WIFS)*, 2016, pp. 1–6, doi: 10.1109/WIFS.2016.7823902.

[43]. D. Aju, A. K. Kakelli, A. S. Varma, and K. Rajendiran, "A comprehensive perspective on mobile forensics: Process, tools, and future trends," in *Confluence of AI, Machine, and Deep Learning in Cyber Forensics*. Hershey, PA, USA: IGI Global, 2021, pp. 1–28, doi: 10.4018/978-1-7998-4900-1.ch001.

[44]. R. Ayers, S. Brothers, and W. Jansen, *Guidelines on Mobile Device Forensics*, NIST Special Publication 800-101, Rev. 1. Gaithersburg, MD, USA: National Institute of Standards and Technology, 2014, doi: 10.6028/NIST.SP.800-101r1.

[45]. B. Manral, G. Somani, K.-K. R. Choo, M. Conti, and M. S. Gaur, "A systematic survey on cloud forensics challenges, solutions, and future directions," *ACM Computing Surveys*, vol. 52, no. 6, Art. no. 113, 2020, doi: 10.1145/3361216.

[46]. P. Purnaye and V. Kulkarni, "A comprehensive study of cloud forensics," *Archives of Computational Methods in Engineering*, vol. 29, no. 1, pp. 33–46, 2022, doi: 10.1007/s11831-021-09575-w.

[47]. C. Karagiannis and K. Vergidis, "Digital evidence and cloud forensics: Contemporary legal challenges and the power of disposal," *Information*, vol. 12, no. 5, Art. no. 181, 2021, doi: 10.3390/info12050181.

[48]. G. Palmer, "A road map for digital forensic research," Digital Forensic Research Workshop (DFRWS), Tech. Rep. DTR-T001-01, Nov. 2001.

[49]. F. Pagani and D. Balzarotti, "AutoProfile: Towards automated profile generation for memory analysis," *ACM Transactions on Privacy and Security*, vol. 25, no. 1, Art. no. 6, 2022, doi: 10.1145/3485471.

[50]. D. Paul Joseph and J. Norman, "An analysis of digital forensics in cyber security," in *Proc. 1st Int. Conf. Artificial Intelligence and Cognitive Computing*, Advances in Intelligent Systems and Computing, vol. 815, 2019, pp. 701–708, doi: 10.1007/978-981-13-1580-0_67.

[51]. E. Nowroozi, A. Dehghantanha, R. M. Parizi, and K.-K. R. Choo, "A survey of machine learning techniques in adversarial image forensics," *Computers & Security*, vol. 100, Art. no. 102092, 2021, doi: 10.1016/j.cose.2020.102092.

[52]. M. M. Islam, G. Karmakar, J. Kamruzzaman, M. Murshed, G. Kahandawa, and N. Parvin, "Detecting splicing and copy-move attacks in color images," in *Proc. 2018 Digital Image Computing: Techniques and Applications (DICTA)*, Canberra, ACT, Australia, 2018, pp. 1–7, doi: 10.1109/DICTA.2018.8615874.

[53]. S. Ferreira, M. Antunes, and M. E. Correia, "Exposing manipulated photos and videos in digital forensics analysis," *Journal of Imaging*, vol. 7, no. 7, Art. no. 102, 2021, doi: 10.3390/jimaging7070102.

[54]. G. S. Chhabra, V. Singh, and M. Singh, "Hadoop-based analytic framework for cyber forensics," *International Journal of Communication Systems*, vol. 31, Art. no. e3772, 2018, doi: 10.1002/dac.3772.

[55]. N. Usman, S. Usman, F. Khan, M. A. Jan, A. Sajid, M. Alazab, and P. Watters, "Intelligent dynamic malware detection using machine learning in IP reputation for forensics data analytics," *Future Generation Computer Systems*, vol. 118, pp. 124–141, 2021, doi: 10.1016/j.future.2021.01.004.

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 9, September 2026

DOI: 10.17148/IJARCCE.2026.15906

[56]. M. Ali, S. Shiaeles, N. Clarke, and D. Kontogeorgis, "A proactive malicious software identification approach for digital forensic examiners," *Journal of Information Security and Applications*, vol. 47, pp. 139–155, 2019, doi: 10.1016/j.jisa.2019.04.013.

[57]. Mohammad, R. M. A. (2022). An enhanced multiclass support vector machine model and its application to classifying file systems affected by a digital crime. *Journal of King Saud University - Computer and Information Sciences*, *34*(2), 179–190. https://doi.org/10.1016/j.jksuci.2019.10.010.

[58]. R. B. Hadiprakoso, W. R. Aditya, and F. N. Pramitha, "Statistical analysis of Android malware detection using supervised machine learning algorithm," *CyberSecurity and Digital Forensics*, vol. 5, no. 1, pp. 1–5, May 2022, doi: 10.14421/csecurity.2022.5.1.3116.

[59]. N. L. Beebe and L. Liu, "Clustering digital forensic string search output," *Digital Investigation*, vol. 11, no. 4, pp. 314–322, 2014, doi: 10.1016/j.diin.2014.10.002.

[60]. I. Indramana and A. Purwanto, "Comparative analysis of supervised learning and unsupervised anomaly detection in security log analysis for post-incident digital forensic investigation," *Journal of Business, Social and Technology*, vol. 7, no. 2, 2026, doi: 10.59261/jbt.v7i2.605.

[61]. H. Li, B. Xi, S. Wu, J. Jiang, and Y. Rao, "The application of association analysis in mobile phone forensics system," in *Intelligence Science II: Third IFIP TC 12 International Conference, ICIS 2018, Beijing, China, November 2–5, 2018, Proceedings*, Lecture Notes in Computer Science, vol. 11189. Cham, Switzerland: Springer, 2018, pp. 126–133, doi: 10.1007/978-3-030-01313-4_13.

[62]. M. S. Rich, "Enhancing Microsoft 365 security: Integrating digital forensics analysis to detect and mitigate adversarial behavior patterns," *Forensic Sciences*, vol. 3, no. 3, pp. 394–425, 2023, doi: 10.3390/forensicsci3030030.

[63]. U. Islam, A. A. Alsadhan, H. S. Alwageed, A. A. Al-Atawi, G. Mehmood, M. Ayadi, and S. Alsenan, "SentinelFusion based machine learning comprehensive approach for enhanced computer forensics," *PeerJ Computer Science*, vol. 10, Art. no. e2183, 2024, doi: 10.7717/peerj-cs.2183.

[64]. A. M. Alashjaee, "Machine learning approach for fake image forensics," *University of Bisha Journal for Basic and Applied Sciences*, vol. 1, no. 1, Art. no. 5, 2025, doi: 10.65073/3122-3508.1005.

[65]. M. C. Todd and G. L. Peterson, "Temporal metadata analysis: A learning classifier system approach," *Forensic Science International: Digital Investigation*, vol. 51, Art. no. 301842, Dec. 2024, doi: 10.1016/j.fsidi.2024.301842.

[66]. X. Du and M. Scanlon, "Methodology for the automated metadata-based classification of incriminating digital forensic artefacts," in *Proc. 14th Int. Conf. Availability, Reliability and Security (ARES '19)*, Canterbury, U.K., Aug. 26–29, 2019, pp. 1–8, doi: 10.1145/3339252.3340517.

