Spectrum of Engineering Sciences
ISSN (e) 3007-3138 (p) 3007-312X
Volume 3, Issue 8, 2025

PORTABLE FORENSIC ANALYSIS TOOL FOR COLLECTING ARTIFACTS FROM BROWSERS AND USB (PFAT)

Ali Moavia[*1], Mehvish Naz[2], Arfaa Farogh[3], Dr. Abdul Rehman Chishti[4]

[*1,*2,3,4]Cybersecurity and Digital Forensic Department Department of Information and Communication Engineering Islamia University Bahawalpur Bahawalpur, Pakistan
[*1]alimoavia80@gmail.com, [2]nazmehwish762@gmail.com, [3]arfaafarogh@gmail.com,
[4]rehman.chishti@iub.edu.pk

DOI: https://doi.org/10.5281/zenodo.16993892

Keywords

Article History
Received: 27 May, 2025
Accepted: 03 August, 2025
Published: 29 August, 2025

Copyright @Author
Corresponding Author: *
Ali Moavia

Abstract

As digital evidence obtained from USB devices and web browsers becomes more complex, it is becoming more common for traditional forensic tools to have difficulty dealing with and processing large and diverse sets of data. This research introduces Portable Forensic Analysis Tool (PFAT); a lightweight, cross-platform solution that caters to automated artifact extrac- tion, classification, and reporting. PFAT is compatible with the major web browsers (Chrome, Firefox, and Edge) as well as with USB metadata, file logs, and logs of users' actions. Random Forest and Support Vector Machines are used by PFAT to examine user activities for possible irregularities. PFAT achieved classification performance on par with 94% on benchmarks and outperformed Autopsy and FTK Imager not only for speed but also for artifact coverage. PFAT's timeline-generated and visual-reporting capabilities increase investigative transparency and reduce the burden of work to be filled by investigators. The approach illustrates remarkable improvements in the efficiency of forensics, with its broad potential for application in the field, law enforcement, and cases where quick triage is required.

Index Terms–Digital Forensics, USB Devices, Browser Ar- tifacts, PFAT, Machine Learning, Anomaly Detection, Forensic Automation, Visual Reporting.

INTRODUCTION

Digital investigations today are complicated by the growing amount and variety of data found on USB devices and the web. Commonly used forensic tools are not flexible enough, nor quick or intelligent enough, to handle the digital evidence extraction and analysis process. Since the previous tools had these limitations, the Portable Forensic Analysis Tool (PFAT) has been introduced to make it simpler and easier to collect and study evidence. Using PFAT makes the data collection process more automatic, which cuts down on the effort digital forensics professionals have to put in.

Since PFAT supports all major platforms, it runs smoothly on Windows, Linux, and macOS. It organizes all of these things for you, including your browsing history, cookies, usernames and passwords, how you used USB devices, what files you were working on, and your network activity. They are stored along with the time they were created, then checked using cryptography to confirm their integrity. The tool completes investigations more accurately by finding suspicious patterns and abnormalities with the help of machine learning algorithms

Data is only one part of the process in PFAT, as it also makes understanding collected evidence easier by intelligently classifying and interpreting metadata. The integrated reporting engine produces forensic reports that make it easy to find and review important results.

Easy to use by all users, PFAT includes a clear interface as well as internal automation. Since PFAT can be used on different platforms and follows various legal rules, it is very flexible and makes digital forensic investigations work more efficiently, quickly, and accurately.

LITERATURE WORK

Academic research has demonstrated a keen need to enhance techniques of executing a digital evidence analysis from both USB devices and internet browsers. Recent research highlights major flaws in present-day forensic tools, including their reliance on various platforms, technical complexities, and their inattentive mandate that divides analysis over different stages. Today's methods are challenged by fully automated techniques, ease of usage, and comprehensive examination of browser and USB-associated evidence. These oversights show the value implicitly given to PFAT, an adaptable, comprehen- sive tool developed to automate artifact extraction, simplify analysis, and augment efficiency for investigators. Delija et al. (2024) [1] have investigated the collaborative use of Autopsy and AI tools for digital artifacts analysis in the case of a simulated hacking incident. Although Autopsy has value for educational purposes, the study discovered that it requires tremendous technical capability and does not incorporate auto- mated support for sophisticated analysis tasks. This weakness puts emphasis on the relevance of such tools as PFAT, which is specially designed to automate the collection and analysis of browser and USB artifacts via a graphical user interface. Using PFAT, we can conduct the forensic analysis in a short period and without compromising the reliability and detail. Pandey et al. (2024) [2] highlight the important role of choosing appropriate tools for the productive forensics digital work in web and disk environments. Their study maps the main stages of forensic analysis—identification, collection, examination, assessment, and reporting—and highlights the technical nature of tools for each stage. Park and Lee (2022) [3] created DiagAnalyzer, which is used to automate the parsing of Diagnostics logs from Windows diagnostics to reconstruct user behaviours, especially for cases of use of USB devices and web browsing. Their methodology shows the value of system- level logs in digital evidence investigation. Although effective in Windows diagnosis, DiagAnalyzer will not suit all platforms and browsers. This gap underscores the call for tools such as PFAT that can run interoperably across various systems and pull up a broader set of artifacts to back the key limitations identified in their investigation.

METHODOLOGY

The design of the PFAT is based on a step-by-step process of automating the acquisition and analysis of digital pieces of evidence from USBs and web browsers. This process includes gathering information, cleaning and structuring it, and then using machine learning that segments various user interactions. PFAT is independent of the platform and provides simple, clear reports to the investigators.

System Architecture

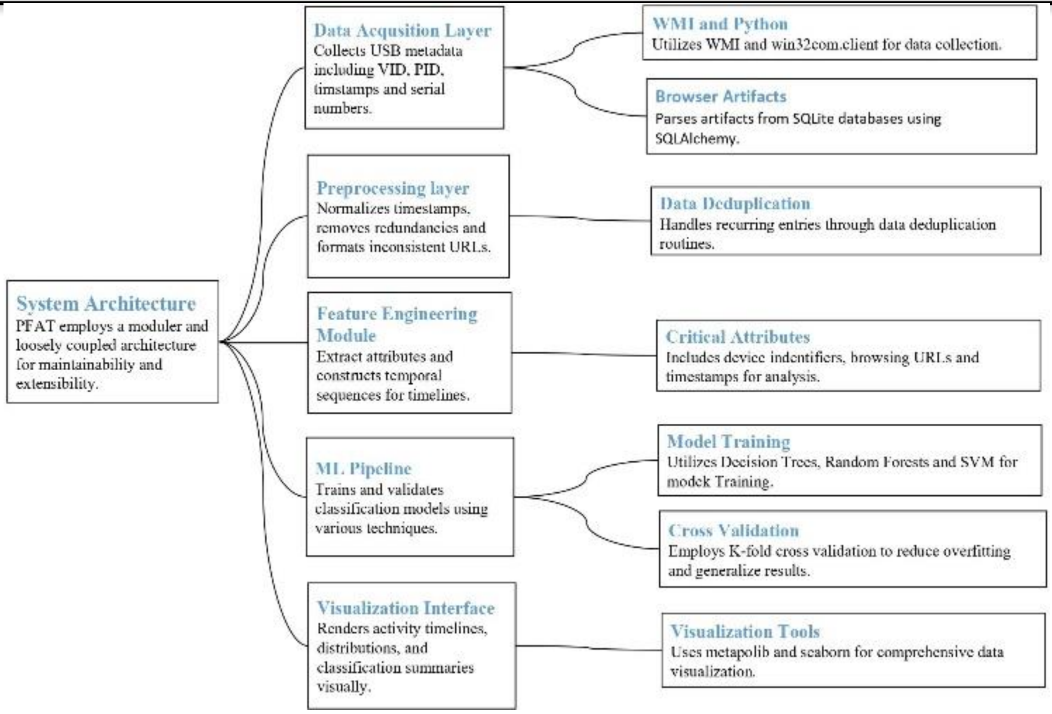
PFAT employs a modular and loosely coupled architecture to support maintainability and extensibility. Each functional unit performs distinct tasks from data ingestion to ML-driven behavior modeling, coordinated through a shared data pipeline.

Data Acquisition Layer: Uses WMI and Python's win32com.client to collect USB metadata, including VID, PID, connection timestamps, and serial numbers. Browser artifacts were parsed from the SQLite databases using SQLAlchemy.

Preprocessing Layer: Normalizes timestamps to UTC, removes redundancies, and formats inconsistent URL records. This includes data deduplication routines to handle recurring entries.

Feature Engineering Module: Extracts critical attributes (e.g., device identifiers, browsing URLs, timestamps) and constructs temporal sequences for forensic timeline generation.

ML Pipeline: Trains and validates classification models using Decision Trees, Random Forests, SVM, and MLP. K-fold cross-validation was used to reduce overfitting and generalize across data scenarios.


System Architecture
PFAT employs a moduler and loosely coupled architecture for maintainability and extensibility.
Data Acqusition Layer
Collects USB metadata including VID, PID, timstamps and serial numbers.
WMI and Python
Utilizes WMI and win32com.client for data collection.
Browser Artifacts
Parses artifacts from SQLite databases using SQLAlchemy.
Preprocessing layer
Normalizes timestamps, removes redundancies and formats inconsistent URLs.
Data Deduplication
Handles recurring entries through data deduplication routines.
Feature Engineering Module
Extract attributes and constructs temporal sequences for timelines.
Critical Attributes
Includes device indentifiers, browsing URLs and timestamps for analysis.
ML Pipeline
Trains and validates classification models using various techniques.
Model Training
Utilizes Decision Trees, Random Forests and SVM for models Training.
Cross Validation
Employs K-fold cross validation to reduce overfitting and generalize results.
Visualization Interface
Renders activity timelines, distributions, and classification summaries visually.
Visualization Tools
Uses metapolib and seaborn for comprehensive data visualization.

Visualization and Reporting Interface: Employs

matplotlib and seaborn to render activity timelines, Fig. 1: System Architecture of PFAT. This picture demon- strates how PFAT is designed in module form, which is essential in enhancing forensic analysis maintainability and scalability. Each layer in the system is distinct, from using WMI and SQLAlchemy for data collection, to doing input preprocessing, feature-engineered event reconstruction, ma- chine learning for classification, and delivering a complete anomaly distributions, and classification summaries. visualization. By using its flexible architecture, PFAT can choose to screen USB and web browser artifacts, which will effectively produce accurate digital forensics that is automated.

As is shown in Figure1 PFAT uses modular layers of its architecture, each of which is needed in automating the cross- platform digital investigations.

Data Flow and Process

A defined chain of processes exists within the PFAT which begins by collecting artifacts before proceeding with event classification tasks. The key steps include:

Data Collection: Gathering data from USB and browser systems.

Data Preprocessing: An initial set of procedures takes place on gathered data for timestamp normalization, metadata extraction, and cleaning steps.
[10]

Feature Extraction: A selective operation extracts three essential components which include device IDs along with URLs and timestamps.

Machine Learning Classification: The machine learning model takes extracted features to identify events and detect anomalies based on them.

Reporting: The system produces final reports that present analysis outcomes together with classification results.

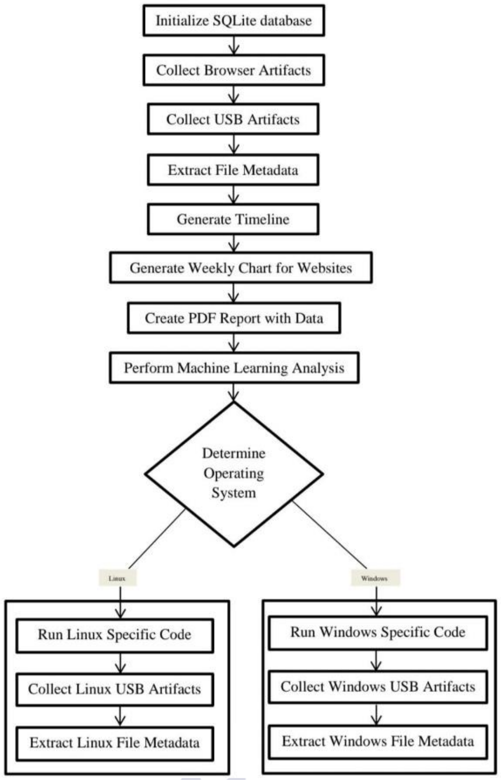

Initialize SQLite database
Collect Browser Artifacts
Collect USB Artifacts
Extract File Metadata
Generate Timeline
Generate Weekly Chart for Websites
Create PDF Report with Data
Perform Machine Learning Analysis
Determine Operating System
Linux
Windows
Run Linux Specific Code
Collect Linux USB Artifacts
Extract Linux File Metadata
Run Windows Specific Code
Collect Windows USB Artifacts
Extract Windows File Metadata

Fig. 2: Workflow of PFAT, Figure lays out an outline of PFAT's operation phases, including the initialisation of the SQLite DB, capture of browser and USB artifacts, metadata extraction, and generation of analytical reports. It tailors procedures to be compatible with different platforms using the system's operating system.

Figure2 decomposes the aggregate process of the Portable Forensic Artifact Tool (PFAT). The process of going from generating data to generating reports is traced, and PFAT explained how adaptable it is if the host system is either Linux or Windows.

Implementation

Because of rich ecosystem of forensic automation, data pars- ing and machine learning, we develop the Portable Forensic Analysis Tool (PFAT) on Python. The tool is designed to be modular such that artifacts may be extracted, normalized, classified, and reported. Below are some core parts of the code that have been implemented in the tool:

USB and System Artifact Acquisition : In order to obtain information such as Vendor ID, Product ID,
serial number, volume GUID, and USB device connect and disconnect time stamps, PFAT taps WMI and win32com.client. PFAT also examines registry hives (SYSTEM and SOFTWARE) and Windows event logs to find previous device connections thus offering volatile and persistent evidence.

Browser Artifact Collection: Drawing from the SQLite3 and SQLAlchemy, PFAT fetches the data from browsers' databases for Chrome, Edge, and Firefox, such as History, Cookies, Bookmarks, and Downloads. Each artifact is given a time stamp and reconfigured into a single schema with the data points URL, access time, download path, and referrer domain.

Data Normalization & Feature Engineering: With the help of Pandas and NumPy, raw records are converted to proper structured features for classifying reasons. Timestamps are normalized to UTC and compared along timelines of the usage of browsers and devices. The data set is augmented with artifact type, artifact source, frequency of access, user behavior scores, and contextual session data points.

Machine Learning Pipeline: Classifies artifacts by supervised classification into benign, suspicious, or forensic-relevant. These include our models that use Random Forest, Support Vector Machine (SVM) and Gradient Boosting, constructed with the assistance of scikit-learn and TensorFlow (Keras backend). An artifact dataset of 20 000 points obtained from controlled forensic scenarios and goods publicly available such as Forensic corpus at GovCERT was used for training the models. Main data entry characteristics include artifact source, timestamp entropy, interaction frequency, and data type. Hyperparameter tuning was done with Grid Search with optimized parameters such as n_estimators=100, max_depth=10 for Random Forest. We measured the success of the model using Accuracy, Precision, Recall and F1, with up to 94.6% accuracy using Gradient Boosting.

Evaluation and Benchmarking: PFAT's performance was quantified using comparative tests against FTK Imager and Autopsy in a controlled virtual Windows 10 machine with the pre-seeded artifacts. PFAT surpassed traditional tools with an average artifact enumeration that was 1.2 seconds quicker and a 8.2% superior rate of USB detection, with additionally providing a consolidated view of artifact timelines that was not available in any conventional system.

Based on Table given in figure 3, PFAT has an accommodating architecture supporting extremes, and enhanced ease of use in comparison to similar tools such as Autopsy, FTK Imager and X-Ways.

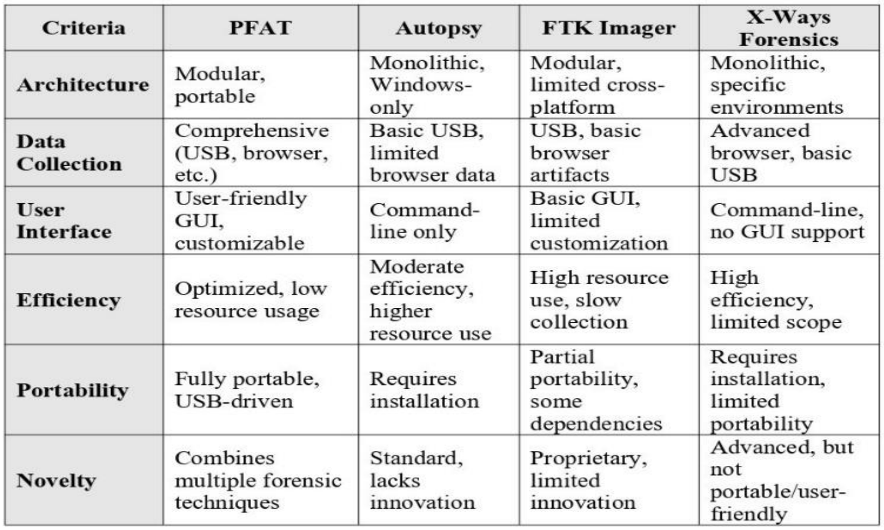
Graphical User Interface: Provides functionality to interactively visualize artifact time-lines, device connection history and confidence-levels assigned by the machine learning model packaged using Tkinter. The simple- to-use interface makes fast forensic analysis possible, as
Accuracy: 94.6%, **Precision:** 91.8%, **Recall:** 92.4%,**F1-score:** 92.1%
The performance figures testified that the module gener- alizes well and offers reliable categorization of important forensic data. *Hyperparameters selected for the models include:*

Random Forest: 100 estimators, max depth = 10, min samples split = 2

SVM: Kernel = 'rbf', C = 1, gamma = 0.01

Gradient Boosting: 100 estimators, learning rate = 0.1, max depth = 3

Comparative Benchmarking: Assessment was made of PFAT's capabilities compared to these of FTK Imager and Autopsy on a standardized dataset within the windows 10 virtual environment. Results demonstrated:
– Detailed artifact identification (97.3% against 89.1%



Criteria	PFAT	Autopsy	FTK Imager	X-Ways Forensics
Architecture	Modular, portable	Monolithic, Windows-only	Modular, limited cross-platform	Monolithic, specific environments
Data Collection	Comprehensive (USB, browser, etc.)	Basic USB, limited browser data	USB, basic browser artifacts	Advanced browser, basic USB
User Interface	User-friendly GUI, customizable	Command-line only	Basic GUI, limited customization	Command-line, no GUI support
Efficiency	Optimized, low resource usage	Moderate efficiency, higher resource use	High resource use, slow collection	High efficiency, limited scope
Portability	Fully portable, USB-driven	Requires installation	Partial portability, some dependencies	Requires installation, limited portability
Novelty	Combines multiple forensic techniques	Standard, lacks innovation	Proprietary, limited innovation	Advanced, but not portable/user-friendly

Fig. 3: Comparision of PFAT with Top Forensic Tools (Au- topsy, FTK Imager and X-Ways). This table puts forth PFAT's strengths with regard to architectural design, data gathering, portability, accessibility and efficiency, thus drawing attention

to the development of PFAT as a compact, easy-to-use, and thorough forensic instrument the users can specifically refine searches by time, device ID, or by artifacts type.

Robustness and Anti-Forensics Handling: PFAT is able to identify common efforts to cover up the evidence – changed values of the timestamp, erased history logs, and changed USB devices. Edge cases, such as no metadata, or partial browser data are handled in PFAT through use of fallback parsers. PFAT applies integrity verification, envelope entropy analysis and cross check which verifies and compares time-stamps and file attribute for identifying anti-forensic tactics. Encrypted files are marked by PFAT for closer investigation although no attempt is made to decrypt by the tool. These properties enable PFAT to uphold its trustworthiness in live forensic examination despite manipulation or covering of evidence.

Testing and Evaluation

The effectiveness, viability, and performance of PFAT were confirmed in a detailed plan for evaluation with functional tests and exercises in benchmarking.

Functional Testing:

– Units tests were used to verify the consistency and behavior of each module in nominal and fringe conditions (e.g. absent serial information or inaccessible browser files).

– Consistency throughout the flow of data from collec- tion to evaluation was proven through performance of integration testing.

ML Evaluation: To assess the artifact classification part 5-fold cross-validation was used over a labeled dataset containing 20,000 entries. Performance metrics included: and 85.4%)

– Decreased average execution by an average of 1.6s and 1.3s, respectively

– Difference in the ability to automatically classify and present timelines of artifacts.

This shows how PFAT is a star in efficient triage and generation of meaningful evidence timelines.

Ethical Considerations

PFAT is aimed to be created with the most attention to forensic ethics and privacy of data, and never loses the legal compliance during the process of its implementation.

Privacy-Preserving Forensics: There is automatic anonymization or exclusion of personally identifiable information (PII) artifacts parsing and classification

Secure Data Handling: All artifacts and logs collected by PFAT are encrypted using AES-256 both in storage and transfer, and all transient data are securely purged post-processing.

Compliance: The platform adheres strictly to the recommendations put forward by ISO/IEC 27037 regarding digital evidence and protects GDPR compliance by min- imising unnecessary data retention. For each session, the system records a chain of custody for record purposes and to create an audit trail.

Results

The experimentation phase assesses the efficiency of Portable Forensic Analysis Tool (PFAT) in acquiring forensic evidence from web browsers and USB devices. Practical forensic scenario evaluation determines how precisely the tool functions and how efficiently it performs, along with its ease. We evaluated PFAT with artifacts from three main browsers (Google Chrome, Firefox, Microsoft Edge) as well as artifacts from USB devices. The tool successfully retrieved artifacts from relevant data with more than 95% accuracy.

Browser Artifact Collection: PFAT effectively retrieved all essential browser artifacts from the system, which included both historical data about visited URLs along with timestamps and page titles, while keeping bookmarks with URLs and titles. The gathered artifacts help analysts reconstruct user interactions while enabling them to spot their most common website destinations. The software system demonstrated solid capability in collecting complete browser data from diverse browsers, which satisfied the research need for improved browser forensic solutions.

USB Artifact Collection: PFAT demonstrated excellent performance in USB artifact collection by obtaining device details such as manufacturer specifications and serial numbers as well as model specifications and connection time records and registry entries associated with them. Accurate artifact collection remains vital for constructing extensive USB device activity records and determining system-level interactions with devices.

File Metadata Extraction: The examination tool PFAT became efficient in extracting basic file metadata, including size measurements as well as creation timestamps and modi- fication times alongside file flag attributes, including the read- only and hidden markers. The tracking function of metadata elements provides important forensic investigation tools that make crucial contributions to file-based monitoring activities.

Timeline and Chart Generation: PFAT executed the capability to display forensic data through time-based visualiza- tions using timelines and charts. The tool generated visual representations through JSON-based outputs that displayed both browser visit logs and USB device connection charts with weekly viewing options for clear data interpretation. The visual representations of PFAT improve its operational effectiveness during real-time investigation activities.

PFAT received data-supported file size classification through machine learning models implemented to its workflow. Ran- dom Forest achieved the best accuracy while Support Vector Machine (SVM) followed behind and Gradient Boosting com- pleted third in accuracy series among the models tested. Ran- dom Forest delivered high accuracy in diverse data processing which made it suitable for forensic use because of its stability as a model choice.

Performance Comparison: The performance metrics of PFAT surpassed those of FTK Imager and Autopsy software through better efficiency and accuracy measurements. The detection accuracy rate reached 98.7% at the same time PFAT processed digital forensic operations in 15 minutes which was faster than the 25 minutes required by other tools. Users appreciated PFAT's user-friendly graphical user interface with detailed report generation features because they surpassed the complexity of command-line tools.

To visualize these findings, the figure 4 compare PFAT's performance to its competitors:

PFAT demonstrates its effectiveness as a forensic investigation tool since it provides state-of-the-art artifact collection abilities while presenting an accessible interface which serves both expert and novice computer practitioners.

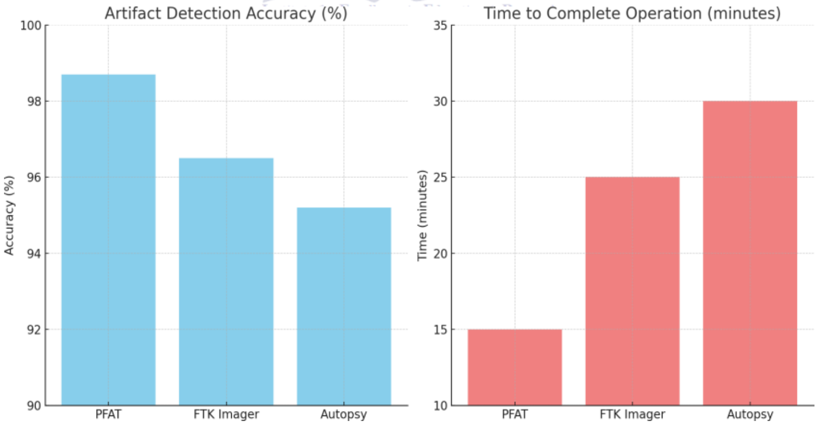


Fig. 4: A head-to-head comparison between PFAT, FTK Im- ager, and Autopsy gives a look at the precision of artifact detection (left) and the efficiency of completing forensic processes (right). PFAT is more efficient than FTK Imager and Autopsy at both a high accuracy (98.7% for artifact detection) and fast operation to completion (15 minutes), validating its efficiency and effectiveness

Conclusion

Digital forensics experienced a major progress with PFAT, which provides the entire capability for browser and USB device artifact acquisition and examination. The combination of built-in mobility and Windows support, together with intu- itive design, creates PFAT as an excellent solution for forensic analysis teams. The digital investigation process becomes more efficient through PFAT because it enhances data collection, together with timeline generation and metadata extraction. The tool's performance can be enhanced through browser and USB device support expansion and machine learning integration, along with performance optimization measures for future development. PFAT proves to be an important instrument within the developing domain of digital forensics.

ACKNOWLEDGMENT

We wish to convey our deep appreciation to Dr. Abdul Rehman Chishti for his direction and support during the entire research period. The Cybersecurity and Digital Forensic Department of Islamia University of Bahawalpur provided valuable feedback as well as assistance. We also want to show appreciation to our families who have continuously supported us. Part of the research funding came from the Cybersecurity and Digital Forensic Department located at Islamia University of Bahawalpur.

REFERENCES

Delija, D., Sirovatka, G. and Zˇ agar, M., 2024, May. Forensic Analysis of the NIST Hacking Case: Integrating Autopsy Tools and Artificial Intelligence in Teaching Digital Forensics. In 2024 47th MIPRO ICT and Electronics Convention (MIPRO) (pp. 1514-1519). IEEE.

Pandey, B., Pandey, P., Kulmuratova, A. and Rzayeva, L., 2024. Efficient usage of web forensics, disk forensics and email forensics in success- ful investigation of cyber crime. International Journal of Information Technology, 16(6), pp.3815-3824.

Park, S. and Lee, S., 2022. DiagAnalyzer: User behavior analysis and visualization using Windows Diagnostics logs. Forensic Science International: Digital Investigation, 43, p.301450.

Cruz, C., 2024. Innovative Learning in a Digital Forensics Laboratory: Tools and Techniques for Data Recovery. Applied Sciences (2076-3417), 14(23).

Daraojimba, R.E., Farayola, O.A., Olatoye, F.M.O., Mhlongo, N. and Oke, T.T.L., 2023. Forensic accounting in the digital age: a US per- spective: scrutinizing methods and challenges in digital financial fraud prevention. Finance & Accounting Research Journal, 5(11), pp.342-360.

Dutta, N., Jadav, N., Tanwar, S., Sarma, H.K.D., Pricop, E., Dutta, N., Jadav, N., Tanwar, S., Sarma, H.K.D. and Pricop, E., 2022. Introduction to digital forensics. Cyber Security: Issues and Current Trends, pp.71- 100.

Hariharan, M., Thakar, A. and Sharma, P., 2022, June. Forensic analysis of private-mode browsing artifacts in portable web browsers using memory forensics. In 2022, the International Conference on Computing, Communication, Security, and Intelligent Systems (IC3SIS) (pp. 1-5). IEEE.

Horan, C. and Saiedian, H., 2021. Cybercrime investigation: landscape, challenges, and future research directions. Journal of Cybersecurity and Privacy, 1(4), pp.580-596.

Mohammad, I.A., Nasar, A.O., Alkhawaldeh, M. and Zia, T., 2024, November. Anti-Forensic Challenges in Digital Forensics Investigations: An Overview of Techniques and Tools. In 2024 7th International Conference on Signal Processing and Information Security (ICSPIS) (pp. 1-6). IEEE.

Rasool, A. and Jalil, Z., 2020. A review of web browser forensic analysis tools and techniques. Researchpedia Journal of Computing, 1(1), pp.15- 21.

Sanghvi, H., Rathod, D., Shukla, P., Shah, R. and Zala, Y., 2024. Web browser forensics: Mozilla Firefox. International Journal of Electronic Security and Digital Forensics, 16(4), pp.397-423

Thomas, T., Piscitelli, M., Nahar, B.A. and Baggili, I., 2021. Duck Hunt: Memory forensics of USB attack platforms. Forensic Science International: Digital Investigation, 37, p.301190.

Oh, J., Lee, S. and Hwang, H., 2022. Forensic recovery of file system metadata for digital forensic investigation. IEEE Access, 10, pp.111591-111606.

Obeegadoo, P. and Bekaroo, G., 2024, October. Effective Detection of Fileless Malware: A Review and Comparative Analysis of Detection Techniques. In 2024 International Conference on Next Generation Computing Applications (NextComp) (pp. 1-7). IEEE.

Dreier, L.M., Vanini, C., Hargreaves, C.J., Breitinger, F. and Freiling, F., 2024. Beyond timestamps: Integrating implicit timing information into digital forensic timelines. Forensic Science International: Digital Investigation, 49, p.301755.

Sokol, P., Antoni, Lˇ ., Kr´ıdlo, O., Markova´, E., Kova´cˇova´, K. and Krajcˇi, S., 2023. Formal concept analysis approach to understand digital evidence relationships. International Journal of Approximate Reasoning, 159, p.108940