REVA UNIVERSITY
Bengaluru, India

School of Computer Science and Engineering

Course Code: B22EF0608	Research based Mini Project Report – Review 2	Academic Year: 2024-2025
		Semester: 6th SEM E

Project Group Members Details

Group No:	Group Name: E6					
Sl. No	SRN	Full Name	Sec	Mobile No.	Email_ID	Sign
1	R22EF289	MICHAEL JOHN PAUL	E	+91 9789657583	22020131956@reva.edu.in	
2	R22EF291	MOHIT CHOUDHARY	E	+91 7877753056	22020135666@reva.edu.in	
3	R22EF300	PRAMUKH SATISH	E	+91 7676871817	22020437205@reva.edu.in	
4	R21EF016	C AKSHAI	E	+91 9686368064	2103881@reva.edu.in	

Task Distribution

Project Leader	MOHIT CHOUDHARY
Development Lead:	PRAMUKH SATISH
Document Lead:	MICHAEL JOHN PAUL, C AKSHAI

Project Details

Project Title:	"ENHANCING FILE SECURITY WITH BLOCKCHAIN-BASED ANOMALY DETECTION SYSTEMS"	
Type of Project	BLOCKCHAIN AND CYBERSECURITY	
Guide Details:	Name: DR. SUPREET Designation: ASSOCIATE PROFESSOR	Mobile No:
Place of Project Work:	REVA UNIVERSITY, BANGALORE	
Remarks by Guide:		Guide Signature Date:

Guide Signature with Date **HOD CSE Signature with Date** **Director Signature with Date**

1. CONTENT

Page No.

2. ABSTRACT

In the current digital age, ensuring the security and integrity of files is paramount. This mini-project introduces a novel system for enhancing file security using a blockchain-based anomaly detection approach. The system enables users to upload files in formats such as .txt, .pdf, and .exe, and employs a multi-tiered detection mechanism tailored to each file type. Upon file upload, a unique hash is generated and compared against a malware hash database. Files not immediately flagged undergo further scrutiny using rule-based or machine learning techniques. Text files are analyzed for structural and keyword anomalies, PDFs are examined using an ML model trained on specific malicious features, and executables are targeted for future development with behavior and opcode analysis. Final classification outcomes—safe, malicious, or suspicious—are recorded on a blockchain ledger, ensuring transparency, immutability, and traceability. This combination of blockchain with intelligent file-type-specific analysis sets the system apart, aiming to provide a more secure, accountable, and efficient method for threat detection. Ongoing work focuses on improving detection accuracy, enhancing the user interface, and expanding file format support.

Keywords: Cybersecurity, blockchain, anomaly detection, malware detection, machine learning, file integrity, threat classification, digital forensics, secure file upload.

3. INTRODUCTION

With the rapid increase in digital data exchange, securing files against malicious threats has become a critical concern. Traditional antivirus solutions often rely on signature-based detection, which may fail to identify newly emerging or obfuscated malware. Moreover, these systems generally lack transparency in how files are analyzed and classified. To address these limitations, our project proposes an intelligent file security system that integrates anomaly detection techniques with blockchain technology. This approach not only enhances detection capabilities but also provides an immutable audit trail of file analysis results.

The core idea is to analyze files based on their type—text, PDF, or executable—using specialized techniques. Lightweight rule-based methods are applied to text files, while more advanced machine learning models are used for PDFs. Future plans include extending support for executable file analysis through behavior and static pattern analysis. A distinguishing feature of this system is the use of blockchain to log scan results securely, ensuring tamper-proof record-keeping and traceability. By combining targeted anomaly detection with blockchain-based logging, the system offers a robust solution for improving trust and reliability in file security.

4. PROBLEM STATEMENT

As cyber threats become increasingly sophisticated, conventional file security systems struggle to keep pace, often producing inaccurate results such as false positives or failing to detect newly crafted malware. Additionally, most existing systems lack transparency and do not offer verifiable proof of how a file was classified, leaving room for data tampering and mistrust. There is a pressing need for a more intelligent and trustworthy solution that can accurately detect anomalies across different file types while ensuring the integrity and traceability of the results. This project aims to address

these challenges by combining tailored anomaly detection techniques with blockchain-based logging to create a secure and transparent file analysis framework.

5. OBJECTIVES

- To develop a secure file analysis system that detects malicious files using anomaly detection techniques tailored to specific file types (.txt, .pdf, .exe).
- To implement hash-based verification using SHA-256 for quick identification of known malicious files.
- To design lightweight rule-based detection for text files and apply machine learning models for analyzing PDFs.
- To plan and integrate static and behavioral analysis techniques for executable files in future iterations.
- To securely log file analysis results on a blockchain ledger, ensuring transparency, immutability, and traceability.
- To minimize false positives and false negatives by improving the accuracy and reliability of the detection mechanisms.
- To build a user-friendly interface that allows users to easily upload files and view analysis results.
- To expand the system's capability to support additional file types and enhance its applicability across different domains.

6. DESIGN & IMPLEMENTATION

6.1. SYSTEM ARCHITECTURE

The system architecture follows a modular, multi-layered approach to ensure scalability, flexibility, and efficiency in file security analysis. At the core, the system begins with the file upload module, which accepts files in various formats such as .txt, .pdf, and .exe. Once uploaded, the system generates a unique SHA-256 hash for each file, which is immediately compared to a database of known malicious hashes for quick identification of threats. Files that pass this initial check proceed to type-specific analysis. For text files, the system applies rule-based checks to detect anomalies in the content, such as suspicious keywords or unusual character patterns. PDF files undergo machine learning-based analysis, where features like JavaScript presence, auto-launch actions, and embedded content are extracted and used to classify the file. The architecture also integrates a blockchain layer for logging the analysis results and metadata, ensuring tamper-proof record-keeping and traceability. The system is designed to be extensible, with future plans to add support for executable files using behavior and static analysis. The modular design also allows for easy updates and the addition of new detection methods or file types as needed.

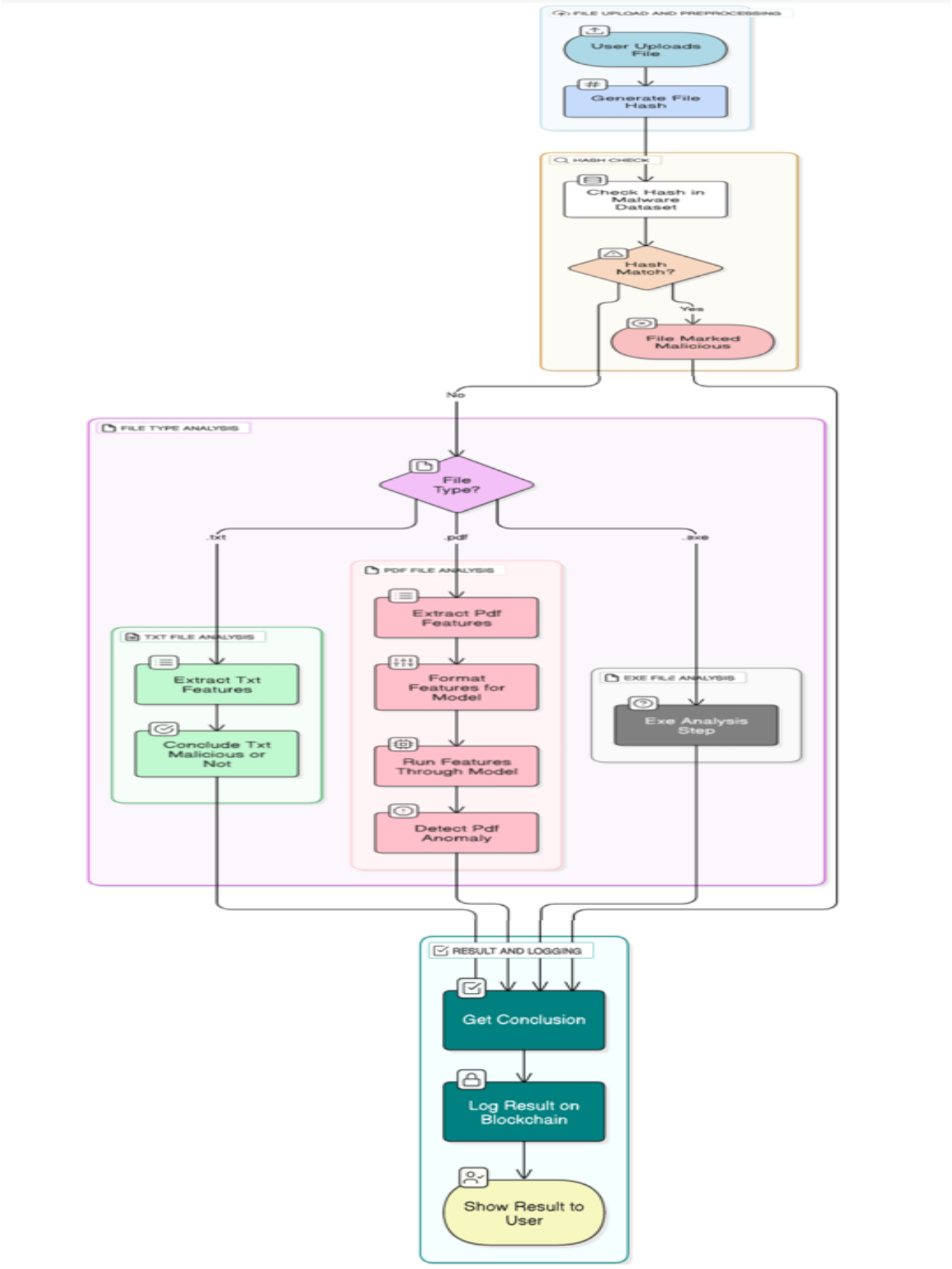
a) **Programming Language: Python**
   Python was chosen as the primary programming language due to its flexibility, vast library support, and ease of implementation for both anomaly detection and blockchain integration. Libraries such as hashlib for hash generation and comparison, pandas for data handling, and scikit-learn for machine learning models were extensively used.

b) **Machine Learning Frameworks: scikit-learn, TensorFlow**
   Machine learning models for PDF file analysis were built using scikit-learn for basic classification algorithms and TensorFlow for deeper models if required. These frameworks allowed for feature extraction and classification based on previously trained data sets.

c) **File Analysis Tools: PyPDF2, pefile**
   Libraries like PyPDF2 were used for parsing and extracting data from PDF files to check for embedded content like JavaScript or auto-launch actions. The pefile library was intended for future use in analyzing executable files, specifically for parsing Portable Executable (PE) file headers and extracting relevant static features.

d) **Frontend: Flask**
   Flask, a lightweight Python web framework, was used to build the frontend of the system. It manages file uploads, triggers the analysis process, and displays results. Flask's simplicity and flexibility allow for seamless integration with the backend, providing a smooth user experience for interacting with the file security system.

6.2. DESIGN DIAGRAMS (FLOWCHART)



6.3. IMPLEMENTATION DETAILS

The implementation of this file security system involves the integration of anomaly detection techniques with blockchain technology. It provides a multi-layered approach for analyzing uploaded files based on their type, such as .txt, .pdf, and .exe. The system uses a combination of Python-based programming, machine learning, and blockchain to ensure accurate, efficient, and tamper-proof file analysis.

Tools Used

1. Python – Primary language used for developing the file analysis system, leveraging libraries for machine learning and blockchain integration.
2. scikit-learn – Machine learning library for building and training classification models, used for analyzing PDF files.
3. TensorFlow – Deep learning framework (optional) for more advanced PDF analysis models in future implementations.
4. web3.py – Python library to interact with the Ethereum blockchain for logging analysis results.
5. PyPDF2 – Used for parsing PDF files to extract features like JavaScript and auto-launch actions.
6. pefile – Library for parsing and analyzing executable files (future integration for .exe files).
7. Git – Version control system for managing codebase changes and collaboration.

6.3.1. Key Features Implemented:

1. **File Upload Support** – Allows users to upload .txt, .pdf, and .exe files for analysis. Ensures broad applicability by supporting multiple file formats commonly used for document sharing and software distribution.
2. **SHA-256 Hash Generation** – Automatically generates and compares file hashes to identify known malicious files. Provides a quick and efficient way to verify files against a database of known malware, enhancing detection speed.
3. **File-Specific Detection** – Implements tailored detection methods for text files, PDFs, and planned executable files. Optimizes the detection process by using different techniques suited to the unique characteristics of each file type.

4. **Rule-Based Analysis for Text Files** – Extracts and analyzes basic features such as file size, string count, and suspicious keywords. Enables fast, resource-efficient checks for potential threats without relying on complex algorithms for simpler text-based files.
5. **Machine Learning for PDF Files** – Uses ML models to classify PDFs based on features like JavaScript presence, encryption flags, and auto-launch actions. Adapts to evolving threats in PDF files by learning from a diverse dataset of features and continuously improving its accuracy.
6. **User Interface (UI) –** Provides an easy-to-use interface for uploading files, viewing results, and interacting with the system. Simplifies user interaction with intuitive design, allowing users to easily understand the results and take appropriate actions.
7. **Error Handling** – Implements error detection for missing files, incorrect formats, and unsupported file types. Improves user experience by proactively addressing potential issues, reducing downtime and manual troubleshooting.
8. **Result Classification** – Classifies files as "Safe," "Malicious," or "Suspicious" based on analysis, with options for manual review. Offers clarity on file status, helping users make informed decisions or escalate for further investigation if needed.
9. **Extensible Design** – Modular architecture for future support of additional file types and enhanced detection techniques. Provides flexibility to expand the system's capabilities over time, accommodating new threats and evolving file formats.

6.3.2. TOOLS USED:

7. INTERMEDIATE PROGRESS & FUNCTIONALITY

7.1. COMPLETED MODULES

The following modules were successfully designed, developed, and tested.

1. File Upload and Hash Generation:

- Allows Implemented file upload support for .txt, .pdf, and .exe formats.
- Automatically generates SHA-256 hash for each uploaded file.
- Compares generated hash with a preloaded dataset of known malicious hashes.
- Immediately flags and blocks files with matching malicious hashes.
- Ensures the integrity of files before further processing.

2. Text File (.txt) Analysis Module:

- Extracts basic features like file size, character count, and keyword presence.
- Identifies patterns using rule-based logic to detect potentially harmful content.
- Lightweight checks for suspicious strings or command patterns.
- Efficient execution with minimal computational resources.
- Suitable for early-stage filtering of benign or clearly malicious files.

3. PDF File (.pdf) Analysis Module

- Extracts complex features such as JavaScript presence, auto-launch actions, and encryption.
- Applies a pre-trained machine learning model for classification.
- Supports feature preprocessing to match training data format.
- Classifies PDFs as safe or malicious based on learned patterns.
- Continually refined using new training data during testing.

4. Executable File (.exe) Module (In Development)

- Planned static analysis to detect suspicious opcode sequences.
- Behavior-based analysis to simulate execution in a controlled environment.
- Features under development include PE header parsing and string extraction.
- Focus on identifying ransomware, trojans, and other malware types.
- Will support modular integration for future scalability.

5. Blockchain Logging and Security Layer

- Logs each file’s hash, analysis result, and metadata on a blockchain.
- Ensures immutability and tamper-proof record-keeping of scan outcomes.
- Helps in creating an auditable and transparent trail for future references.

- Blockchain structure designed to be lightweight and efficient.
- Adds trust and integrity to the system’s classification process.

6. User Interface (UI) and Experience

- Basic UI implemented for file uploads and result viewing.
- Displays clear verdicts: Safe, Malicious, or Suspicious.
- Currently improving layout and interactivity for better usability.
- Future updates will include visual indicators and scan history.
- Responsive design planned for broader accessibility.

7. Error Handling and Debugging

- Implemented checks for file not found, unsupported formats, and empty files.
- Actively debugging issues related to false positives and false negatives.
- Included fallback mechanisms for missing data or unexpected inputs.
- Logging error instances for easier identification and testing.
- Continuously refining detection logic based on real-world test results.

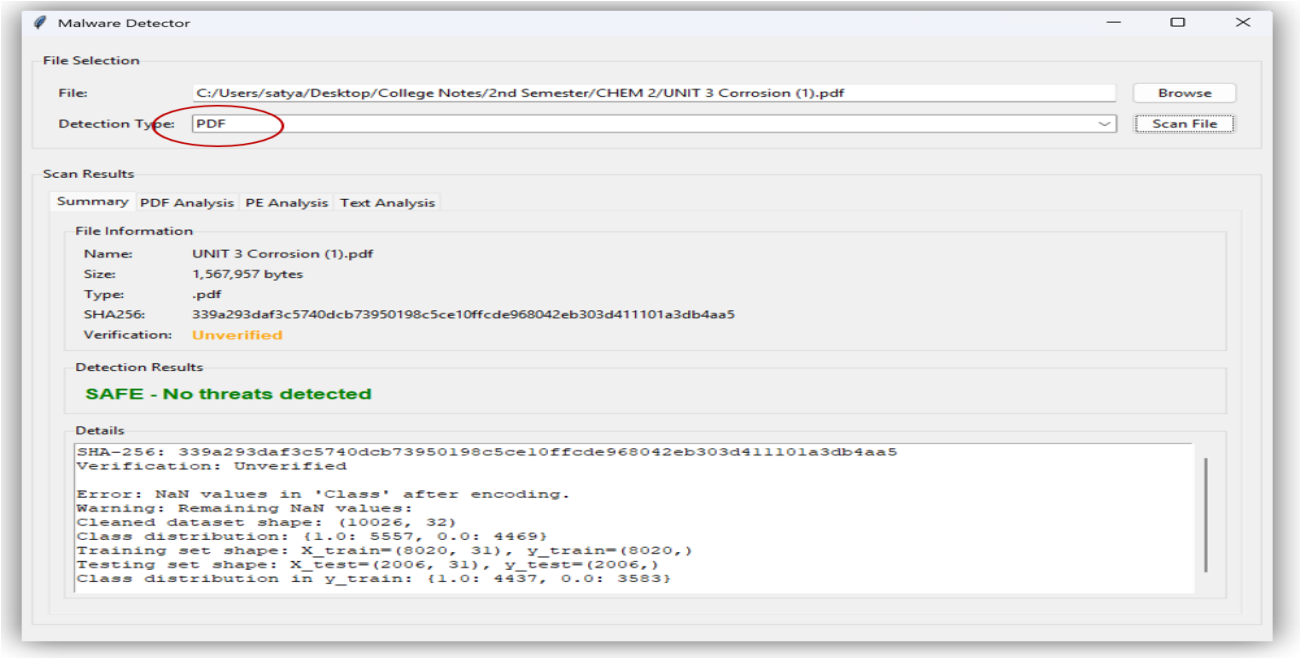
7.2. TESTING AND VALIDATION

The testing phase of the system involved uploading a diverse set of files, including known benign and malicious .txt and .pdf files, to evaluate the accuracy and reliability of the detection mechanisms. For text files, rule-based detection was tested using a variety of samples with both harmless and suspicious content to assess how well the system identifies anomalies based on predefined rules. PDF files were tested with a dataset containing examples of malicious documents (with embedded JavaScript, auto-launch code, etc.) and clean files. The machine learning model's performance was evaluated based on its classification accuracy, with attention given to false positives and false negatives. These tests helped fine-tune the feature extraction and preprocessing pipelines for improved model output.

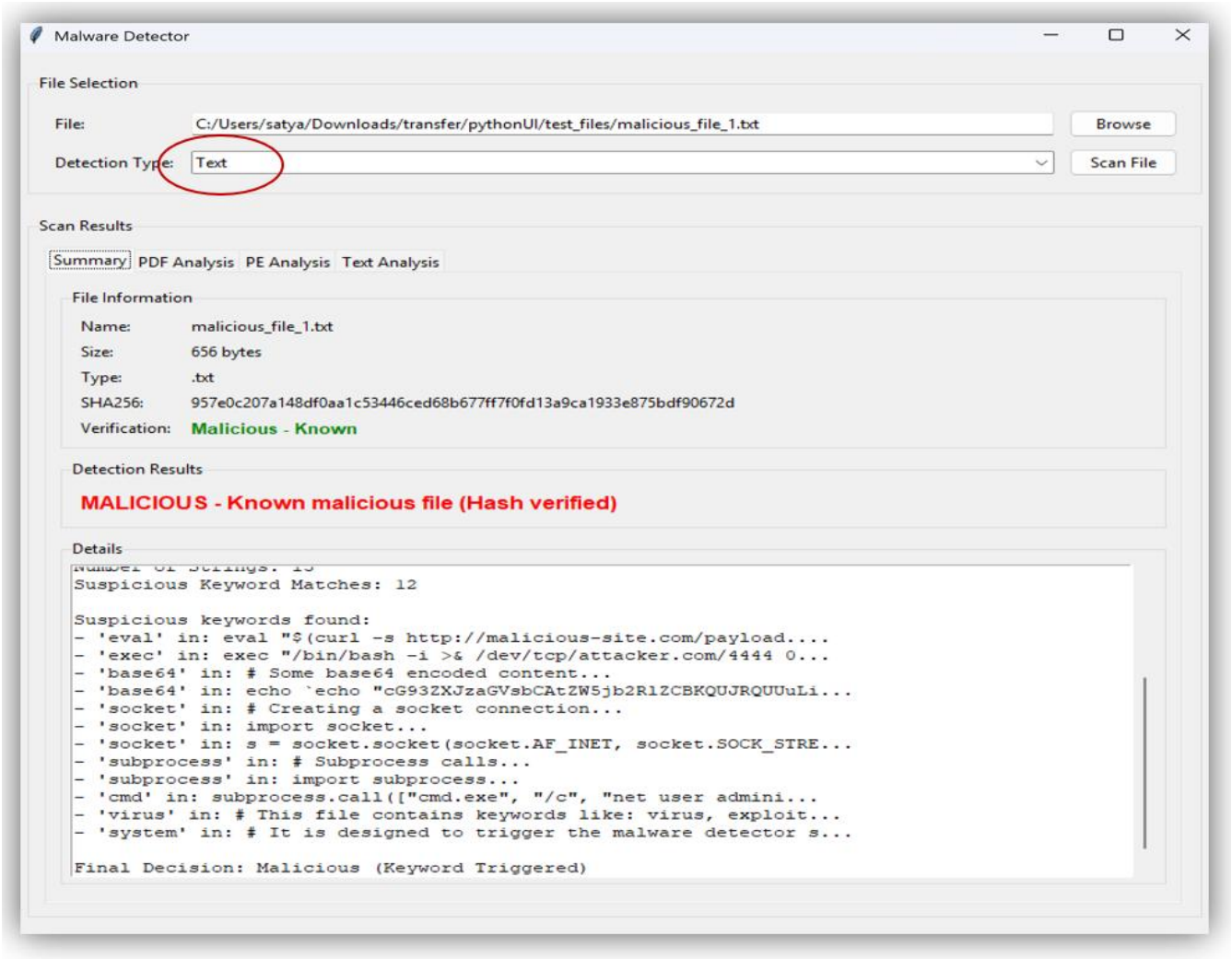
Validation was carried out by comparing system results with known outcomes from antivirus databases and benchmark datasets. Files flagged as malicious were cross-verified using online tools and malware analysis platforms to ensure consistency. Edge cases, such as obfuscated content and borderline suspicious patterns, were marked for manual review to better understand limitations in detection logic. During testing, issues such as inconsistent behavior in edge cases and occasional false alarms were identified and addressed with enhanced rule conditions and improved model retraining. The results demonstrate that while the system shows promising accuracy for .txt and .pdf files, ongoing refinements are required—particularly in preparing the system for handling executable files and in reducing misclassifications.

8. RESULTS & DISCUSSION

The system demonstrated effective performance in identifying known malicious files through hash matching and successfully classified most .txt and .pdf files using their respective detection modules. Text file analysis, based on rule-driven checks, yielded accurate results for straightforward cases, although some edge cases required refinement of keyword lists and anomaly thresholds. The machine learning model for PDFs showed good classification accuracy during testing, correctly identifying embedded threats like JavaScript or auto-launch actions. However, instances of false positives and false negatives were observed, highlighting the need for additional training data and fine-tuning.

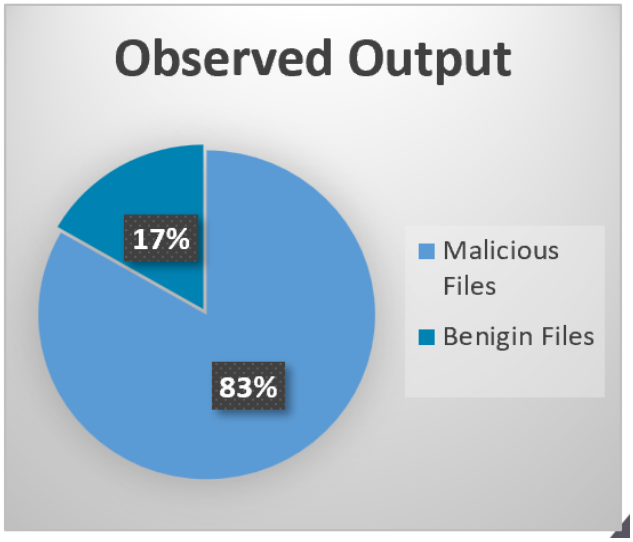


Output Preview (Safe .pdf file)



Output Preview (Safe .pdf file)

File Name	File Info	Expected Status	Observed Status
Hello.exe	Size: 1,446 bytes Type: .exe	Malicious ❌	Malicious ❌
Malicious_file_1.txt	Size: 656 bytes Type: .txt	Malicious ❌	Malicious ❌
Safe_file.txt	Size: 517 bytes Type: .txt	Safe ✅	Malicious ❌
Malicious_pdf_test.pdf	Size: 713 bytes Type: .pdf	Malicious ❌	Malicious ❌
Project_test_file.pdf	Size: 68,387 bytes Type: .pdf	Safe ✅	Safe ✅



Project Accuracy

Total Predictions = 5
Correct Predictions = 4
Accuracy Formula:

$$\text{Accuracy} = \frac{4}{5} = 0.80 = \mathbf{80\%}$$

Project Accuracy: 80%

9. INNOVATION & CREATIVITY

This project stands out by uniquely combining anomaly detection techniques with blockchain technology to enhance file security. Instead of relying solely on traditional antivirus methods, the system introduces a file-type-specific detection strategy, applying tailored rule-based logic for text files, machine learning for PDFs, and planned advanced analysis for executables. The creative integration of blockchain ensures that all scan results are securely logged, providing transparency, traceability, and tamper-proof verification. This innovative approach not only strengthens security but also builds trust in the system's classifications—an aspect rarely addressed in conventional detection systems.

9.1. NOVEL ASPECTS

- **Blockchain Integration for File Security** – Uses blockchain to log file analysis results, ensuring transparency, immutability, and verifiability.
- **File-Type-Specific Detection** – Applies different analysis methods based on file type, improving both efficiency and accuracy.
- **Hybrid Detection Techniques** – Combines rule-based logic, lightweight anomaly detection, and machine learning models to enhance detection performance.

- **Multi-Stage Threat Analysis** – Implements both quick hash-based checks and deeper content analysis for comprehensive file evaluation.

- **Modular Design for Scalability** – Designed to easily incorporate new file types and detection methods, ensuring adaptability to future needs.

9.2. UNIQUE FEATURES INTRODUCED

- **Adaptive Detection Workflow** – Tailors the analysis process to the specific structure and behavior of .txt, .pdf, and planned .exe files.

- **Blockchain-Based Result Logging** – Securely records all scan outcomes and file metadata on a blockchain ledger to prevent tampering.

- **Verdict Classification System** – Clearly labels files as safe, malicious, or suspicious, supporting manual review for uncertain cases.

- **Lightweight Rule System for Fast Screening** – Efficiently processes text files using low-resource rule-based checks for faster threat identification.

- **Planned Executable File Analysis Framework** – Lays the groundwork for future integration of static and behavioral analysis for .exe files.

10. CONCLUSION & FUTURE WORK

This project successfully demonstrates a novel approach to enhancing file security by combining anomaly detection with blockchain technology. By tailoring detection methods to specific file types, the system achieves improved accuracy and efficiency in identifying threats, particularly in .txt and .pdf files. The incorporation of blockchain for logging analysis results adds a critical layer of trust, transparency, and immutability, which traditional file security solutions often lack. Initial testing shows that the system is capable of detecting known and potentially malicious files with a reasonable degree of reliability, although continued refinement is required to handle complex or borderline cases effectively.

Looking ahead, several enhancements are planned to strengthen and expand the system's capabilities. Full implementation of the executable file analysis module will involve advanced static and behavioral analysis techniques. The blockchain component will be further developed to include smart contracts and decentralized access control for greater security and automation. Additionally, the system will be extended to support a wider range of file formats such as .csv, .xlsx, and .bin, making it more versatile for real-world applications. Improvements to the user interface are also prioritized to ensure better usability and a more intuitive experience. These future developments aim to position the system as a robust and scalable solution for modern file security challenges.

11. REFERENCES

[1] İbrahim Üzüm, Özgü Can "An anomaly detection approach for enterprise file integration" 2018 IEEE

[2] Xiaobin Wang; Yonglin Sun; Yongjun Wang "An abnormal file access behavior detection approach based on file path diversity" 2014 IEEE

[3] İbrahim Üzüm, Özgü Can "An anomaly detection system proposal to ensure information security for file integrations " 2018 IEEE

[4] S.D. Wolthusen "Security policy enforcement at the file system level in the Windows NT operating system family" 2002 IEEE

[5] Jiarong Wang; Lijun Cai; Aimin Yu; Min Zhu; Dan Meng "TempatMDS: A Masquerade Detection System Based on Temporal and Spatial Analysis of File Access Records"2018 IEEE

[6] Shiva Mehta; Sumeet Singh Sarpal "Real-Time Anomaly Detection and Data Integrity in Hospitals Using Blockchain and AI" 2025 IEEE

[7] Muskan Pandita; Dipali Himmatrao Patil; Saajan Raina; Monika Malve; Kajal Kashid; Archana J. Jadhav "A Survey on Security Attacks on Blockchain Using Deep Learning" 2023 IEEE

[8] Muskan Pandita; Dipali Himmatrao Patil; Saajan Raina; Monika Malve; Kajal Kashid; Archana J. Jadhav " A Survey on Security Attacks on Blockchain Using Deep Learning" 2023 IEEE

[9] Shiva Mehta; Ravi Kumar "Towards Secure Hospital Data Systems: Real-Time Anomaly Detection and Integrity Management with AI and Blockchain" 2025 IEEE

[10] M. G. Jabir, Muhammad Saad, Muhammad Khurram Khan "Blockchain-based Decentralized Digital Forensics Case Management System using IPFS" 2020 IEEE

REVA UNIVERSITY
Bengaluru, India

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

B.Tech in CSE , AI&DS and IoT

Research based Mini Project INTERNAL – Review 2

Examiner Evaluation Sheet

SEMESTER VI Year: 2024-25 Section:

Examiner Name:

Designation:

Date:

Time:

Parameters for Distribution of Marks		Marks	Course Outcomes
a	Design & Implementation	5	CO2
b	Intermediate Progress & Functionality	5	CO4
c	Innovation & Creativity	5	CO5
d	Publication Progress	5	CO3
e	Presentation & Review Feedback Incorporation	5	CO10
Total		25	Marks

Project Title:

Sl. No	Group No.	SRN	Student Name	a(5)	b(5)	c(5)	d(5)	e(5)	Total (25)
1									
2									
3									
4									

Recommendation by the Examiner: (Kindly √ on one recommendation) Patent | Publication | Product Development

Examiner Signature with Date

HOD CSE Signature with Date

Director Signature with Date

REVA UNIVERSITY
Bengaluru, India

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

B.Tech in CSE , AI&DS and IoT

Research based Mini Project INTERNAL – Review 2

Guide Evaluation Sheet

SEMESTER VI Year: 2024-25 Section:

Guide Name:

Designation:

Date:

Time:

Parameters for Distribution of Marks		Marks	Course Outcomes
a	Design & Implementation	5	CO2
b	Intermediate Progress & Functionality	5	CO4
c	Innovation & Creativity	5	CO5
d	Publication Progress	5	CO3
e	Presentation & Review Feedback Incorporation	5	CO10
Total		25	Marks

Project Title:

Sl. No	Group No.	SRN	Student Name	a(5)	b(5)	c(5)	d(5)	e(5)	Total (25)
1									
2									
3									
4									

Recommendation by the Examiner: (Kindly √ on one recommendation) Patent Publication Product Development

Guide Signature with Date HOD CSE Signature with Date Director Signature with Date