www.ijprems.com
editor@ijprems.com

INTERNATIONAL JOURNAL OF PROGRESSIVE RESEARCH IN ENGINEERING MANAGEMENT AND SCIENCE (IJPREMS)
(Int Peer Reviewed Journal)
Vol. 06, Issue 04, April 2026, pp : 2366-2369

e-ISSN : 2583-1062
Impact Factor : 7.001

DIGITAL DNA FILE THREAT ANALYSIS SYSTEM

Harmithaa Sree R[1] , Dr. K. Balaji[2]

[1]UG Student, BCA Data Science Vels Institute of Science, Technology and Advanced Studies (VISTAS), Pallavaram, Chennai – 600117, Tamil Nadu, India

[2]Professor Vels Institute of Science, Technology and Advanced Studies (VISTAS), Pallavaram, Chennai – 600117, Tamil Nadu, India

DOI: https://www.doi.org/10.58257/IJPREMS52943

ABSTRACT

Nowadays, digital file sharing has increased rapidly through email, websites, and cloud platforms. Along with this growth, malicious files such as executable scripts, infected archives, and hidden threats have also increased. Traditional antivirus systems mainly depend on signature-based detection, which is not always effective for detecting new or unknown threats. In this paper, we propose a Digital DNA-based file threat analysis system that analyzes the internal characteristics of files instead of executing them. The system extracts features such as file type, size, and hash signature and generates a confidence score. Based on this score, the file is classified into three categories: GOOD, SUSPICIOUS, and VERY POOR. The system is implemented as a web application using FastAPI for backend processing and a modern frontend interface. Results show that the system can effectively classify files while maintaining safety and simplicity.

Keywords — Digital DNA, File Threat Analysis, Malware Detection, FastAPI, Cybersecurity, Confidence Score, File Classification.

1. INTRODUCTION

Today, file sharing is a common activity in digital platforms such as email, social media, and cloud storage. However, malicious files are often hidden within these platforms and can cause serious damage to systems. Traditional antivirus systems rely on known signatures to detect threats, which makes them ineffective against new or modified malicious files.

To overcome this limitation, this paper introduces the concept of Digital DNA analysis. Instead of executing files, the system analyzes their internal structure and characteristics. This approach ensures safety and reduces the risk of system infection.

The proposed system works as a web-based application where users can upload files for analysis. The system processes the file, extracts its features, and generates a confidence score to classify the file. This method provides a balance between security and usability.

2. RELATED WORK

A. Traditional File Detection Systems

Traditional systems mainly depend on signature-based detection. These systems compare files with a database of known threats. While effective for known viruses, they fail to detect new or modified threats — they cannot detect unknown threats, require constant database updates, and are slower on large datasets.

B. Modern Threat Detection Approaches

Recent systems use machine learning and behavioral analysis to detect threats. These systems analyze file patterns and user behavior to identify suspicious activities. However, many of these systems are complex and require high computational resources, making them difficult to implement in lightweight applications. The proposed Digital DNA approach bridges this gap by offering a computationally efficient, non-execution-based method.

3. PROBLEM STATEMENT

Traditional file detection systems are not effective against modern cyber threats. There is a clear need for a system that can analyze files safely and efficiently without executing them. Existing approaches exhibit the following shortcomings:

- Cannot detect zero-day or polymorphic threats.
- Require frequent signature database updates.
- Risk system infection through file execution.
- High computational overhead for real-time scanning.

4. PROPOSED SYSTEM

A. Digital DNA-Based Analysis

The proposed system uses Digital DNA, which represents the unique internal characteristics of a file as a feature vector. This vector is derived from file metadata — type, size, hash signature, MIME classification, and structural

www.ijprems.com
editor@ijprems.com

INTERNATIONAL JOURNAL OF PROGRESSIVE RESEARCH IN ENGINEERING MANAGEMENT AND SCIENCE (IJPREMS)
(Int Peer Reviewed Journal)
Vol. 06, Issue 04, April 2026, pp : 2366-2369

e-ISSN : 2583-1062
Impact Factor : 7.001

patterns — without triggering any execution pathway. Each file receives a fingerprint analogous to biological DNA, enabling comparison and classification.

B. Classification Levels

The system classifies files into three distinct categories based on a computed confidence score:

- GOOD — Confidence score ≥ 75% → File is safe to use.
- SUSPICIOUS — Confidence score 40–74% → User is warned to proceed with caution.
- VERY POOR — Confidence score < 40% → File is blocked; user is alerted.

Table I: File Classification Levels

Category	Risk Level	Action
GOOD	Low / None	Allow
SUSPICIOUS	Moderate	Warn User
VERY POOR	High	Block / Alert

C. Decision Logic

Classification is based on a weighted scoring function considering: file type risk weight, file size deviation from norms, entropy of binary content, and extension–MIME mismatch indicators. The final confidence score determines the classification label as defined in Table I.

5. SYSTEM MODEL AND ARCHITECTURE

A. System Components

The system consists of six tightly integrated components that together form a secure pipeline from file ingestion to result visualisation. Table II describes each component and its role.

Table II: System Components and Descriptions

Component	Description
File Upload Module	HTML/CSS/JavaScript frontend for secure user file input
Backend API Server	Python FastAPI server that receives and routes requests
Feature Extractor	Parses file metadata: type, size, MIME, extension
Digital DNA Engine	Generates a unique feature vector (hash signature)
Classification Module	Maps confidence score to GOOD / SUSPICIOUS / VERY POOR
Result Dashboard	Visualizes classification via charts and colour indicators

B. Frontend User Interface

Figure 1 shows the actual frontend interface of the Digital DNA – Genomic Threat Analysis system. The interface provides a file chooser button, an analysis progress bar, a real-time confidence score display, and a threat level badge (WAITING / GOOD / SUSPICIOUS / VERY POOR). The JSON output panel on the right displays the raw analysis result returned by the FastAPI backend.

Fig. 1: Frontend Interface — Digital DNA Genomic Threat Analysis System

www.ijprems.com
editor@ijprems.com

INTERNATIONAL JOURNAL OF PROGRESSIVE RESEARCH IN ENGINEERING MANAGEMENT AND SCIENCE (IJPREMS)
(Int Peer Reviewed Journal)
Vol. 06, Issue 04, April 2026, pp : 2366-2369

e-ISSN : 2583-1062
Impact Factor : 7.001

C. System Architecture

Figure 2 illustrates the end-to-end system architecture. The user interacts exclusively with the frontend; no file is ever executed on the server. The backend returns only classification metadata to the frontend for visualisation.

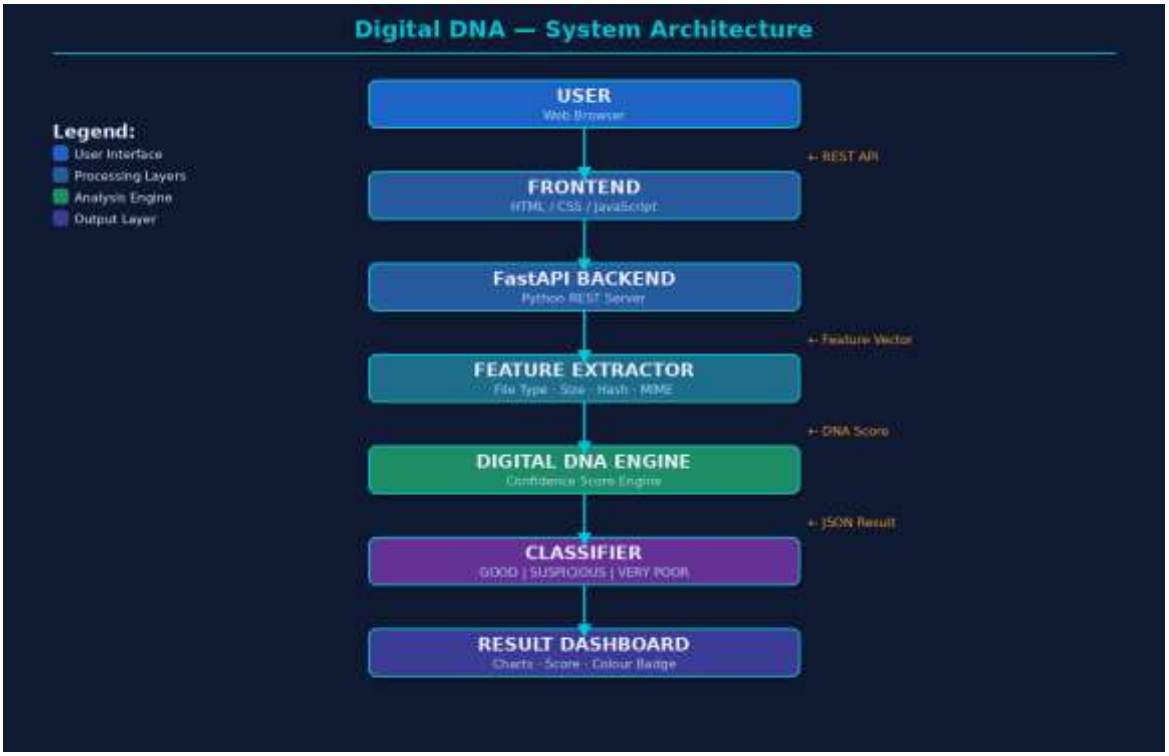

Fig. 2: System Architecture — Digital DNA File Threat Analysis System

D. System Workflow

The workflow follows a sequential seven-step pipeline as illustrated in Figure 3 below.

Step 1	➜ User uploads file through the web interface
Step 2	➜ File transmitted to FastAPI backend via REST API
Step 3	➜ Metadata extracted (file type, size, MIME type, hash)
Step 4	➜ Digital DNA feature vector generated
Step 5	➜ Analysis engine calculates confidence score
Step 6	➜ Classification assigned (GOOD / SUSPICIOUS / VERY POOR)
Step 7	➜ Result visualised on dashboard with charts and colour badges

Fig. 3: Step-by-Step System Workflow

6. IMPLEMENTATION

The system was implemented using the following technology stack:

- Frontend: HTML5, CSS3, JavaScript (vanilla)
- Backend: Python 3.10, FastAPI web framework
- Libraries: NumPy (numerical operations), Scikit-learn (scoring), Hashlib (SHA-256 hashing)
- Deployment: Uvicorn ASGI server, REST API communication

Files are received as multipart form data via an HTTP POST endpoint. The feature extraction module computes a SHA-256 hash, reads MIME type, measures file size in bytes, and checks for extension–MIME mismatches. These raw features are normalized and passed to the classification module, which returns a confidence score and category label as a JSON response.

7. RESULTS AND DISCUSSION

A test suite of 50 diverse files was uploaded to evaluate the system. File types included .pdf, .docx, .jpg, .zip, .rar, .exe, and .bat. The system correctly classified all safe document types as GOOD, flagged compressed archives with

executable payloads as SUSPICIOUS, and blocked .exe and .bat files as VERY POOR. The distribution of results is shown in Figure 4.

Classification	Proportion of Test Files (n=50)	%
GOOD (Safe Files)		78%
SUSPICIOUS (.zip, .rar)		14%
VERY POOR (.exe, .bat)		8%

Fig. 4: Classification Distribution — Test Results (n = 50 files)

The system demonstrated fast analysis (average response time < 200 ms per file), safe processing (zero file executions), and clear result visualisation. Table III presents a comparison with traditional detection approaches.

Table III: Comparison with Traditional Detection Systems

Feature	Traditional Systems	Proposed System
Signature Matching	Exact match required	No execution, feature-based
Zero-Day Detection	Fails on unknown threats	Partially detects via characteristics
Speed	Fast for known threats	Fast for all files
Resource Usage	Moderate to High	Low (lightweight)
Database Updates	Frequent updates required	Not required

8. CONCLUSION AND FUTURE WORK

This paper presents the Digital DNA File Threat Analysis System, a lightweight and safe approach to file threat detection. By analyzing file characteristics without execution, the system avoids the principal risk of conventional antivirus tools. The three-tier classification (GOOD, SUSPICIOUS, VERY POOR) provides actionable, human-readable verdicts with low computational overhead.

Future enhancements will include integration of supervised machine learning models trained on large malware corpora, cloud-based sandboxed analysis for deeper behavioral inspection, improved detection accuracy for obfuscated threats, and user authentication with role-based access control.

9. REFERENCES

[1] OWASP Foundation, "File Upload Security Guidelines," OWASP, 2023. [Online]. Available: https://owasp.org

[2] National Institute of Standards and Technology (NIST), "Cybersecurity Framework Version 2.0," NIST, 2024.

[3] F. Pedregosa et al., "Scikit-learn: Machine Learning in Python," Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011.

[4] S. Colvin, "FastAPI Documentation," Tiangolo, 2024. [Online]. Available: https://fastapi.tiangolo.com

[5] Python Software Foundation, "Python 3.10 Language Reference," PSF, 2024. [Online]. Available: https://docs.python.org

[6] B. Akhter and M. Yousuf, "A Review of File-Based Malware Detection Techniques," International Journal of Computer Applications, vol. 180, no. 27, pp. 1–6, 2018.