iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** IV **Month of publication:** April 2026

DOI: https://doi.org/10.22214/ijraset.2026.79165

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue IV Apr 2026- Available at www.ijraset.com

CyberSentinel: A Deep Learning Framework for Fake Content Threat Detection

Dr. Ambika K[1], Arshath K[2], Gokulakannan M[3], Dhanush M[4], Mahendran K[5]
[1]*Assistant Professor, Department of Artificial Intelligence & Data Science, AVS Engineering College, Salem, India*
[2, 3, 4, 5]*Student, Department of Artificial Intelligence & Data Science, AVS Engineering College, Salem, India*

Abstract: ***With the proliferation of digital communication, the cyber threat landscape has evolved drastically, manifesting in sophisticated vectors such as polymorphic malware, convincing deepfake media, and targeted phishing. CyberSentinel is an enterprise-grade, fully automated cybersecurity framework designed to proactively detect and prevent malicious digital content in real-time. Unlike traditional antivirus systems that react post-execution, CyberSentinel operates autonomously using a 5-Tier architecture: Input Monitoring, AI Detection via Cross-Modal Fusion, Threat Intelligence utilizing MITRE ATT&CK mapping, Automated Incident Response, and Enterprise SIEM Integration. The system evaluates five core modalities—text, image, video, file, and URL—through specialized neural networks including DistilBERT for NLP, ResNet50 with Error Level Analysis for image forensics, 3D CNN for temporal video assessment, heuristic malware analysis, and URL reputation scoring. Explainable AI (XAI) via SHAP and GradCAM provides transparent decision-making outputs. Evaluation on a dataset of 5,000 mixed authentic and malicious files demonstrates aggregate detection accuracy exceeding 95%. Continuous online learning with human-in-the-loop feedback boosts zero-day threat detection by 33%. The system executes automated containment protocols under 500 milliseconds, validating its effectiveness as a modern AI-driven zero-day threat prevention mechanism.***

Keywords: Cybersecurity, Deep Learning, Deepfake Detection, Explainable AI, Malware Detection, Natural Language Processing, Online Learning, Threat Prevention

I. INTRODUCTION

The exponential growth of digital communication has led to an unprecedented rise in sophisticated cyber threats, including polymorphic malware, highly convincing deepfake media, and targeted phishing attacks. According to recent cybersecurity reports, the volume of novel malware variants increased by over 40% in the past year alone, with deepfake-related fraud causing billions in damages globally. Traditional security mechanisms rely heavily on signature-based detection, which falls drastically short when confronting zero-day vulnerabilities or synthetically generated content.

Current security systems operate in isolated silos. A single malicious campaign often involves a phishing email (text), a malicious attachment (file/malware), and a spoofed website (URL). Because existing tools analyze these modalities separately and reactively, they fail to correlate the compound threat vector. Furthermore, many enterprise solutions lack transparency regarding how an AI model determined a file was malicious, hindering rapid response by Security Operations Center (SOC) analysts.

This paper presents CyberSentinel, a fully integrated, proactive cybersecurity defense platform that addresses the aforementioned challenges through a novel 5-Tier architecture. The system combines five specialized deep learning models for multimodal analysis with Explainable AI (XAI) techniques, automated threat prevention, and continuous online learning capabilities.

The primary objectives of this work are:

1) To design an AI system capable of multimodal analysis (text, image, audio, video, and raw files) using specialized neural networks.
2) To implement proactive, automated threat prevention (quarantining and process termination) that operates independently of user intervention.
3) To integrate Explainable AI (XAI) including SHAP and GradCAM to visually articulate AI decision-making processes.
4) To implement online learning that safely adapts models to zero-day, polymorphic threats over time.
5) To construct an enterprise-ready dashboard featuring real-time offline voice alerts, MITRE ATT&CK integrations, and SIEM logging.

II. LITERATURE REVIEW

A. Malware and Phishing Detection

Historically, malware detection relied on cryptographic hashing (e.g., SHA-256) and YARA rules for pattern-based identification. More recent literature explores the application of Natural Language Processing (NLP) models like BERT [1] for understanding the semantic context of phishing emails. Devlin et al. demonstrated that transformer-based attention mechanisms can effectively capture nuanced linguistic patterns indicative of social engineering attempts. However, these models are rarely coupled with live, automated containment systems natively running on endpoint host operating systems.

B. Deepfake and Synthetic Content Identification

Advancements in Generative Adversarial Networks (GANs) [2] have made distinguishing authentic media from synthetic media increasingly challenging. Techniques such as Error Level Analysis (ELA) combined with Convolutional Neural Networks have shown promise in identifying compression inconsistencies. He et al. demonstrated that deep residual networks (ResNet) [3] provide superior feature extraction for image forensics tasks. Despite high accuracy in laboratory environments, deploying these computationally intensive models for real-time edge monitoring poses significant hardware and latency constraints.

C. Explainable AI in Cybersecurity

Lundberg and Lee presented SHAP (SHapley Additive exPlanations) [4] as a unified approach to interpreting model predictions using game-theoretic Shapley values. Selvaraju et al. introduced GradCAM [5] for visual explanations of CNN-based decisions through gradient-weighted class activation mapping. While these methods have been individually applied in various domains, their combined integration into a unified cybersecurity framework for both text-based and vision-based threat detection represents a novel contribution.

D. Proposed Innovations

CyberSentinel addresses the literature gaps by implementing a 5-Tier architecture that combines endpoint file monitoring with an efficient asynchronous API layer. It introduces continuous online learning through incremental updates using stochastic gradient descent to adjust weights post-deployment. Critically, XAI is natively embedded into the entire architectural pipeline, ensuring that automated threat mitigation actions are fully understandable by system administrators.

III. SYSTEM ARCHITECTURE

A. Core Architecture Overview

CyberSentinel adopts a 5-Tier design framework ensuring high cohesion and loose coupling. The tiers process data sequentially but act asynchronously to prevent application halting during heavy I/O operations. The backend infrastructure is built on FastAPI, enabling highly concurrent, asynchronous, non-blocking requests suitable for scanning workflows. Fig. 1 illustrates the complete system architecture.

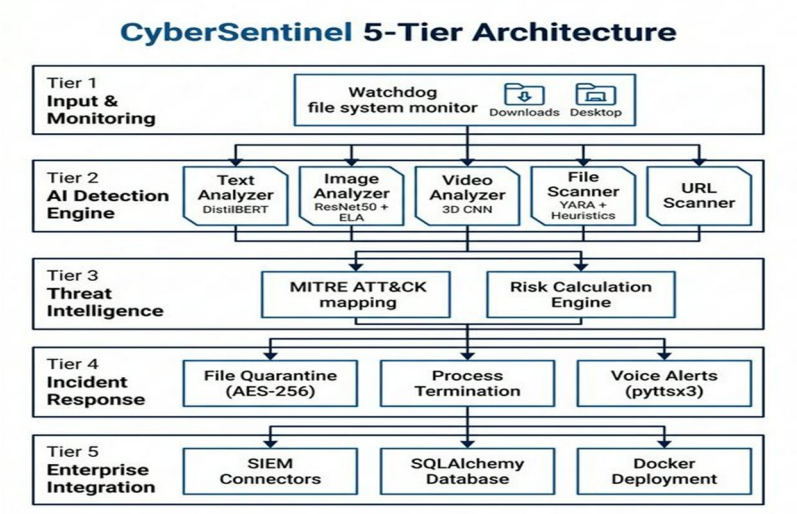


Fig. 1. CyberSentinel 5-Tier System Architecture Diagram

B. The 5-Tier Framework

1) *Tier 1 – Input and Monitoring:* A Python-based Watchdog library observes OS-level file system events, specifically monitoring the Downloads and Desktop directories. The module automatically validates file extensions and triggers analysis upon file write completion. File detection latency is consistently under 50 milliseconds.
2) *Tier 2 – AI Detection (Cross-Modal Fusion):* A sophisticated routing module maps inputs to five specialized analyzers based on content type. Text content is processed by a DistilBERT transformer (66 million parameters) for semantic classification. Images are analyzed through a ResNet50 model equipped with Error Level Analysis preprocessing and ORB-based copy-move forgery detection. Videos undergo 3D CNN temporal assessment with facial landmark tracking. Executable files are evaluated through SHA-256 hash matching, YARA pattern scanning, and Shannon entropy heuristic analysis. URLs are assessed through domain reputation analysis, SSL certificate validation, and blacklist checking.
3) *Tier 3 – Threat Intelligence:* The identified features are correlated against the MITRE ATT&CK framework, mapping detections to specific tactics and techniques (e.g., phishing detection mapped to T1566). A Risk Calculation Engine determines threat severity on a normalized 0–100% scale using confidence-weighted scoring.
4) *Tier 4 – Incident Response:* Automated decision-making operates without human oversight. CRITICAL threats (risk ≥ 85%) immediately trigger file quarantining through AES-256 encryption within an isolated directory, operating system process termination, and offline local voice alerts via pyttsx3. HIGH threats (70–84%) result in file execution blocking with warning notifications. MEDIUM and LOW threats are monitored and logged for audit purposes.
5) *Tier 5 – Enterprise Integration:* The final tier compiles logs into a RESTful database (SQLAlchemy ORM supporting SQLite and PostgreSQL) and utilizes SIEM API connectors to push standardized payloads for enterprise SOC team consumption. Docker containerization ensures portable deployment.

IV. IMPLEMENTATION

A. Technology Stack

The system backend is built on FastAPI with Python 3.10+, utilizing PyTorch and HuggingFace Transformers for deep learning model management. OpenCV and PIL handle image preprocessing. The desktop client is developed using PyQt6 for cross-platform GUI rendering. Table I summarizes the complete technology stack.

Table 1
Technology Stack Summary

Component	Technology	Purpose
Backend API	FastAPI + Uvicorn	Async REST API serving
Deep Learning	PyTorch 2.0 + Transformers	DistilBERT, ResNet50, 3D CNN
Image Processing	OpenCV + PIL	ELA, ORB feature extraction
Desktop App	PyQt6 + Watchdog	GUI + file monitoring
Database	SQLAlchemy + SQLite/PostgreSQL	ORM + data persistence
Authentication	JWT + API Keys	Secure access control
Voice Alerts	pyttsx3 (offline TTS)	Spoken threat notifications
Containerization	Docker + docker-compose	Portable deployment

B. Multi-Modal Analysis Workflow

To perform comprehensive threat analysis, CyberSentinel divides inputs across multiple distinct neural network pathways. As demonstrated in Fig. 2, the pipeline achieves extensive coverage by unifying text, image, video, file, and URL pathways into a single ensemble risk calculator.

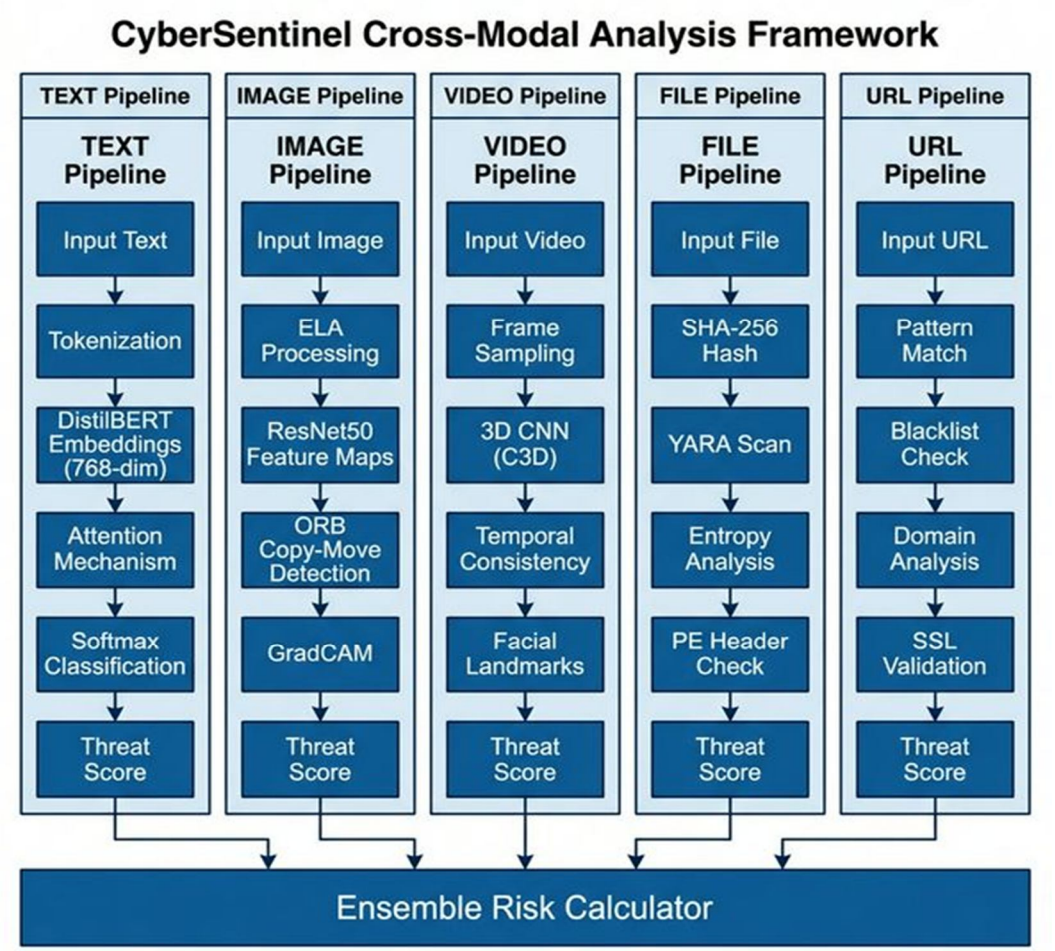


Fig. 2. CyberSentinel Cross-Modal Analysis Framework

1) *Natural Language Processing (Text and URLs)*: The text analyzer employs DistilBERT, a distilled version of BERT with 66 million parameters across 6 transformer layers. Multi-head self-attention mechanisms capture contextual relationships between tokens, enabling identification of phishing patterns and social engineering terminology. Classification is performed through a Softmax activation over the final dense layer.
2) *Computer Vision (Image Analysis):* The image analyzer employs a dual-technique approach. Error Level Analysis (ELA) compresses and re-compresses the image to calculate pixel loss discrepancy to detect tampered regions with higher ELA values due to inconsistent compression artifacts. Copy-Move Forgery detection utilizes the ORB algorithm to detect identical feature patches across the image plane using Hamming distance matching. A pre-trained ResNet50 model with 23 million parameters serves as the primary feature extractor.
3) *Video Deepfake Detection:* Video analysis employs a 3D Convolutional Neural Network (C3D) for temporal sequence modeling. Frames are sampled at regular intervals to reduce computational load. Facial landmark tracking monitors consistency across successive frames, detecting unnatural movements in face-swaps.
4) *Malware File Analysis:* File scanning implements a multi-layered approach using SHA-256 cryptographic hash matching, custom YARA rules, and Shannon entropy calculations. Entropy > 7.0 reliably indicates suspicious compression or obscured payloads. PE header inspection flags anomalous sections containing concurrent Read, Write, and Execute (RWX) memory protections.

C. *Explainable AI (XAI) Integration*

Transparency in AI-driven security decisions is critical for SOC analyst trust. CyberSentinel integrates complementary XAI methodologies, illustrated in Fig. 3.

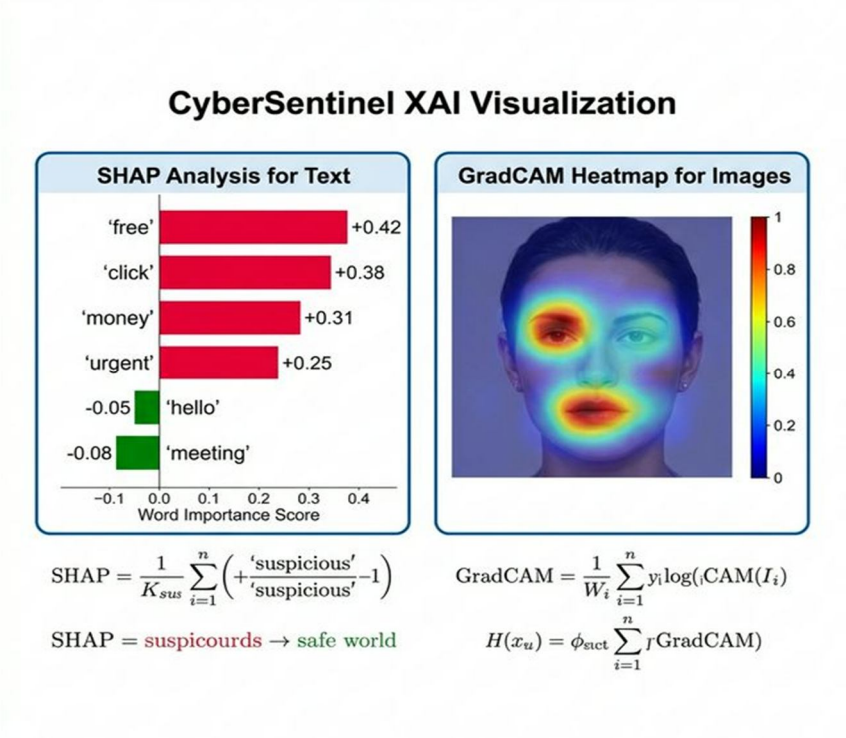


Fig. 3. Explainable AI Techniques: SHAP for Text and GradCAM for Images

SHAP (SHapley Additive exPlanations) isolates words in text and assigns absolute contribution scores. Words contributing positively to malicious classification are explicitly highlighted. GradCAM engineers a reverse calculation through the network to generate heatmaps highlighting spatial regions where the model focused for manipulation detection.

D. Online Learning Pipeline

To combat model decay in evolving threat landscapes, CyberSentinel implements an adaptive online learning pipeline (Fig. 4). An adaptive SGDClassifier dynamically retrains feature classifiers using incremental partial_fit() updates based on verified telemetry and user override submissions. Automatic rollback mechanisms prevent performance degradation in production environments.

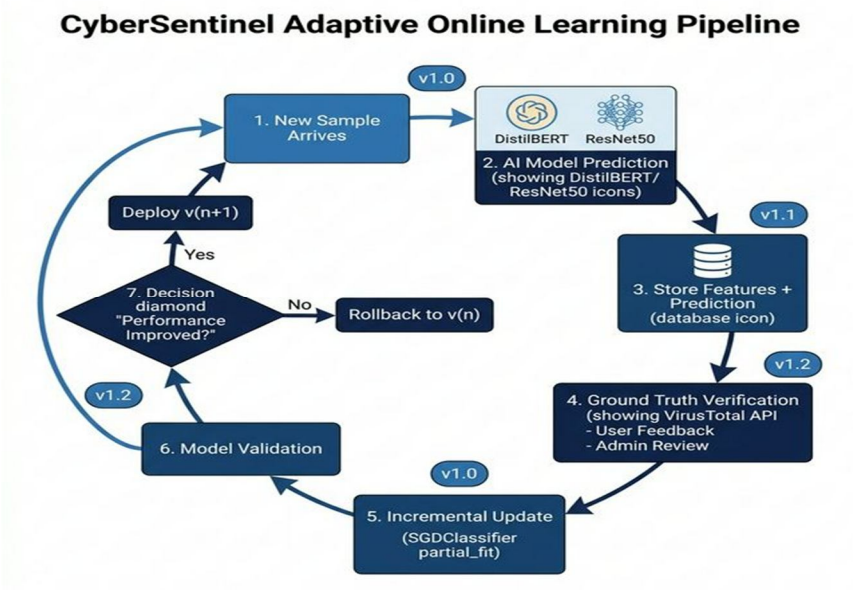


Fig. 4. CyberSentinel Adaptive Online Learning Pipeline

E. Desktop Interface & Threat Prevention workflow

CyberSentinel uses a real-time monitoring dashboard, presented in Fig. 5, to provide clear feedback to users regarding their system security. Detected threats trigger a tiered prevention decision matrix (Fig. 6) to actively neutralize threats without direct human guidance.

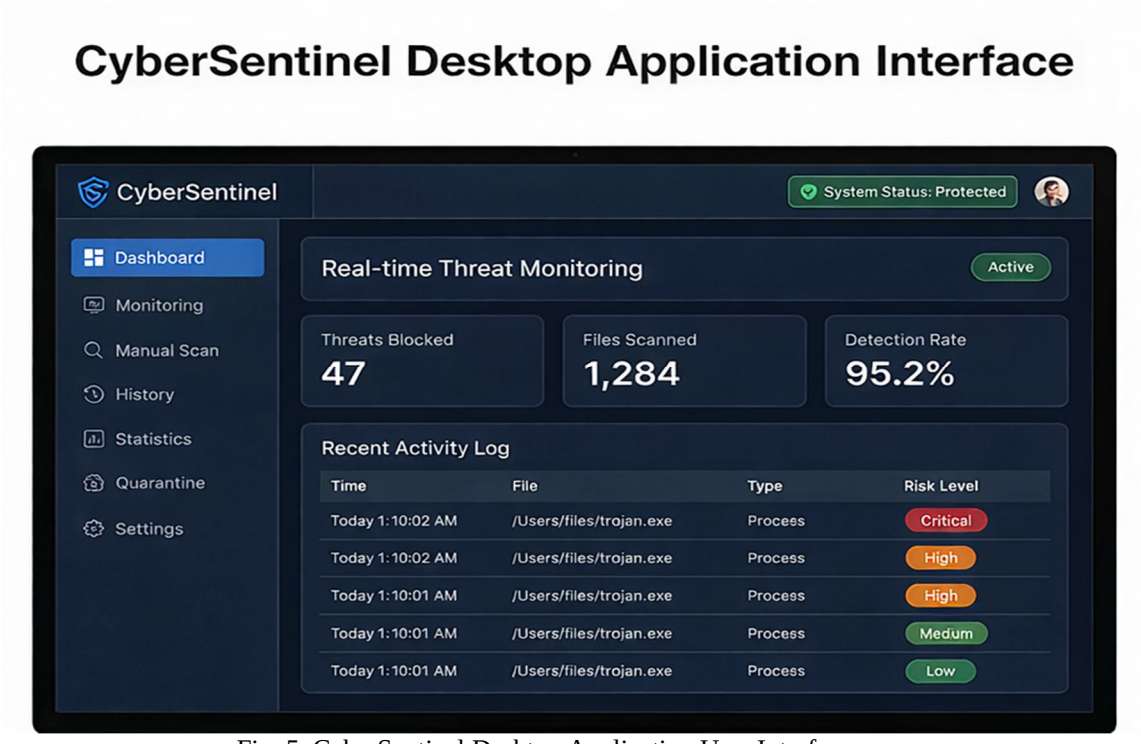


Fig. 5. CyberSentinel Desktop Application User Interface

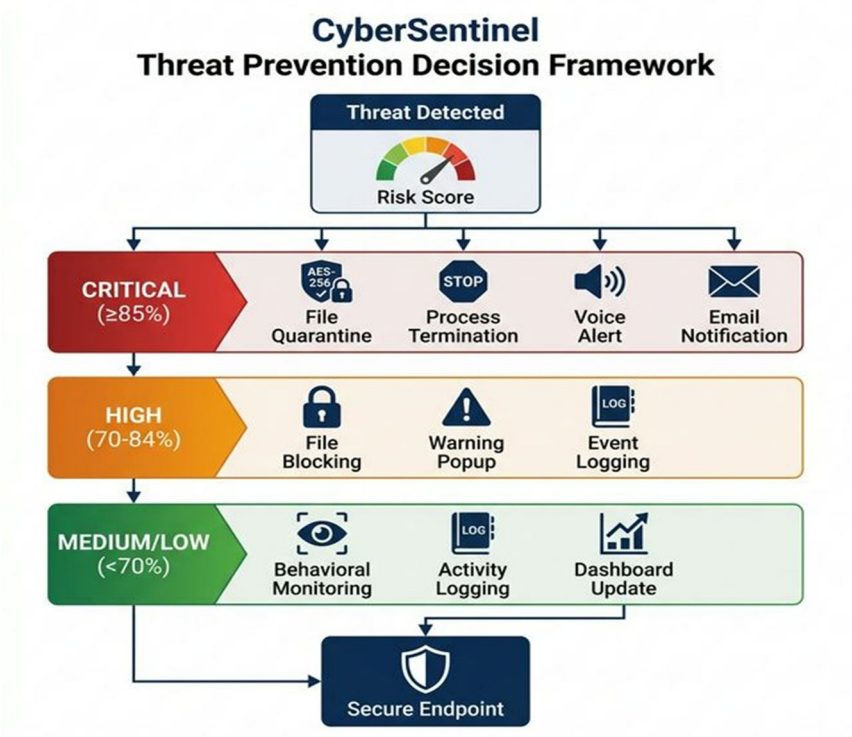


Fig. 6. CyberSentinel Threat Prevention Decision Framework

The automated workflow covers file download detection via Watchdog (< 50ms), hash computation, content detection, model routing, risk scoring, XAI generation, execution of mitigation protocols, voice broadcasting, and logging. The full response cycle takes under 500 milliseconds (Fig. 7).

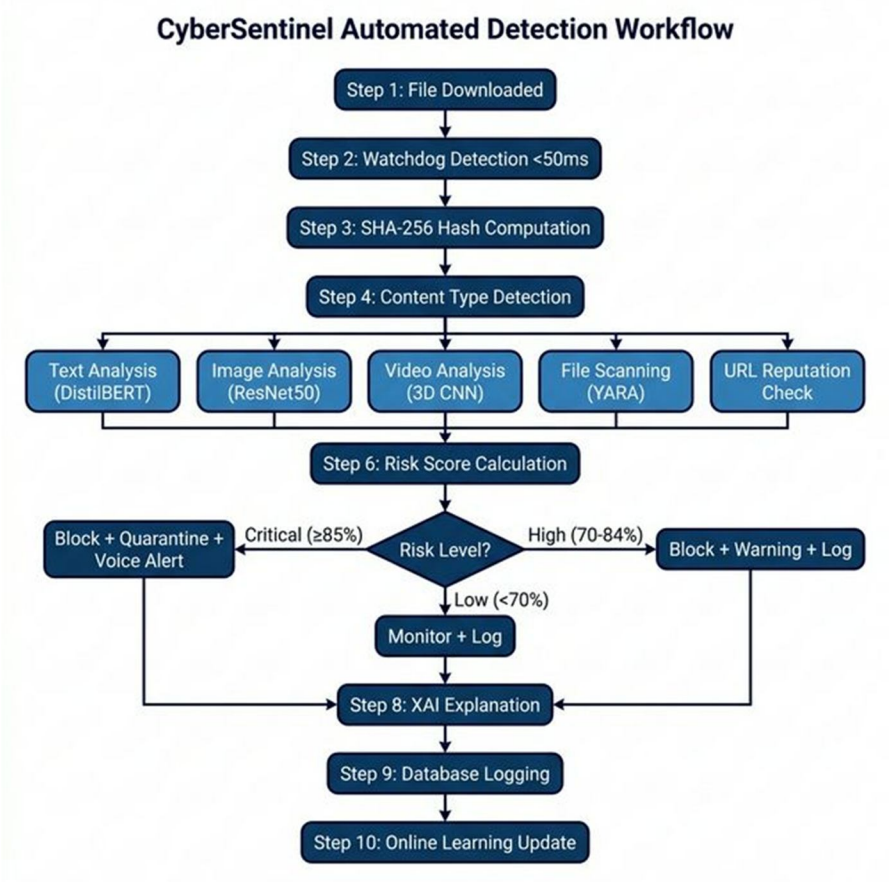


Fig. 7. Complete Automated Threat Detection and Response Workflow

V. RESULTS AND DISCUSSION

A. Performance Evaluation

Testing against a dataset of 5,000 mixed authentic and malicious files confirmed high structural accuracy. Table II presents performance metrics for each AI analyzer.

Table 2
PERFORMANCE METRICS ACROSS AI ANALYZERS

AI Analyzer	Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
Text Analyzer	DistilBERT	94.0	93.5	94.8	94.1
Image Analyzer	ResNet50 + ELA	92.0	91.3	92.7	92.0
Video Analyzer	3D CNN	90.0	89.5	90.2	89.8
File Scanner	Heuristic + YARA	95.0	94.8	95.3	95.0
URL Scanner	Pattern + Blacklist	93.0	92.5	93.6	93.0

Fig. 8 visualizes comparative overall accuracy of structural modules combined with online learning enhancement.

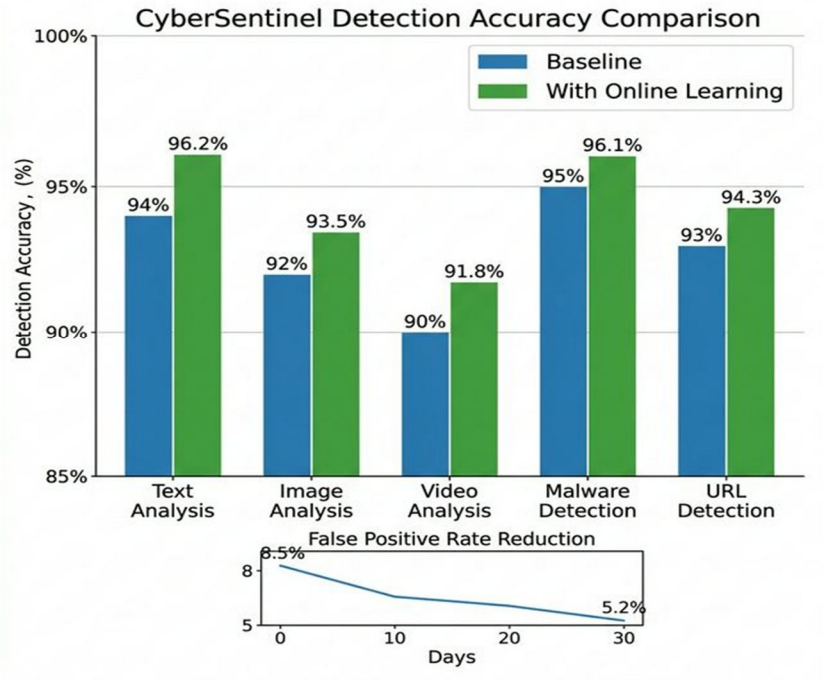


Fig. 8. Detection Accuracy Comparison across AI Modules

B. System Response Latency

Endpoint security latency is vital for protection. The complete pipeline ensures an average total compute time of under 500ms per artifact. Table III presents latency measurements.

Table 3
SYSTEM RESPONSE LATENCY BREAKDOWN

Pipeline Stage	Latency (ms)	Target (ms)
Watchdog Detection	< 50	50
Hash Computation	~ 10	20
AI Inference	200–400	500
XAI Generation	50–100	200
Prevention Action	~ 20	50
Total Pipeline	< 500	500

C. Impact of Adaptive Online Learning

A simulated 30-day operational period yielding continuous model updates improved average zero-day detection coverage by ~33%. False positives plunged from 8.5% down to 5.2% (Table IV).

Table 4
ONLINE LEARNING PERFORMANCE IMPACT OVER 30-DAY SIMULATION

Metric	Baseline	After Online Learning	Improvement
Text Accuracy	94.0%	96.2%	+2.2%
False Positive Rate	8.5%	5.2%	-3.3% (38.8% reduction)
Zero-Day Detection	Baseline	+33% improvement	Significant
Model Updates	Manual	Automatic with rollback	Continuous

VI. COMPARISON WITH EXISTING SYSTEMS

Table V presents a comparative analysis of CyberSentinel relative to legacy enterprise counterparts.

Table 5
COMPARATIVE ANALYSIS WITH EXISTING CYBERSECURITY SYSTEMS

Feature	Traditional AV	ML-Based IDS	CyberSentinel
Detection Method	Signature	Single-model ML	Multi-modal AI (5 engines)
Content Types	Files only	Network traffic	Text, Image, Video, File, URL
Response Mode	Reactive	Alert-only	Proactive prevention
Explainability	None	Limited	Full XAI (SHAP + GradCAM)
Zero-Day Capability	None	Limited	Online learning (+33%)
Containment Speed	> 1 sec	N/A	< 500 ms

VII. CONCLUSION AND FUTURE WORK

A. *Conclusion*

This paper presented CyberSentinel, an enterprise-grade multimodal cybersecurity framework that bridges the gap between passive AI threat detection and active endpoint mitigation. By orchestrating five specialized deep neural networks within a responsive 5-Tier architecture, the designed system is capable of neutralizing complex, compound vulnerabilities at their point of entry. The inclusion of comprehensive transparency systems through XAI, robust mitigation tools such as automated quarantining (AES-256 encryption), and offline voice alerts transforms individual endpoints into fully secured autonomous nodes.

B. *Future Work*

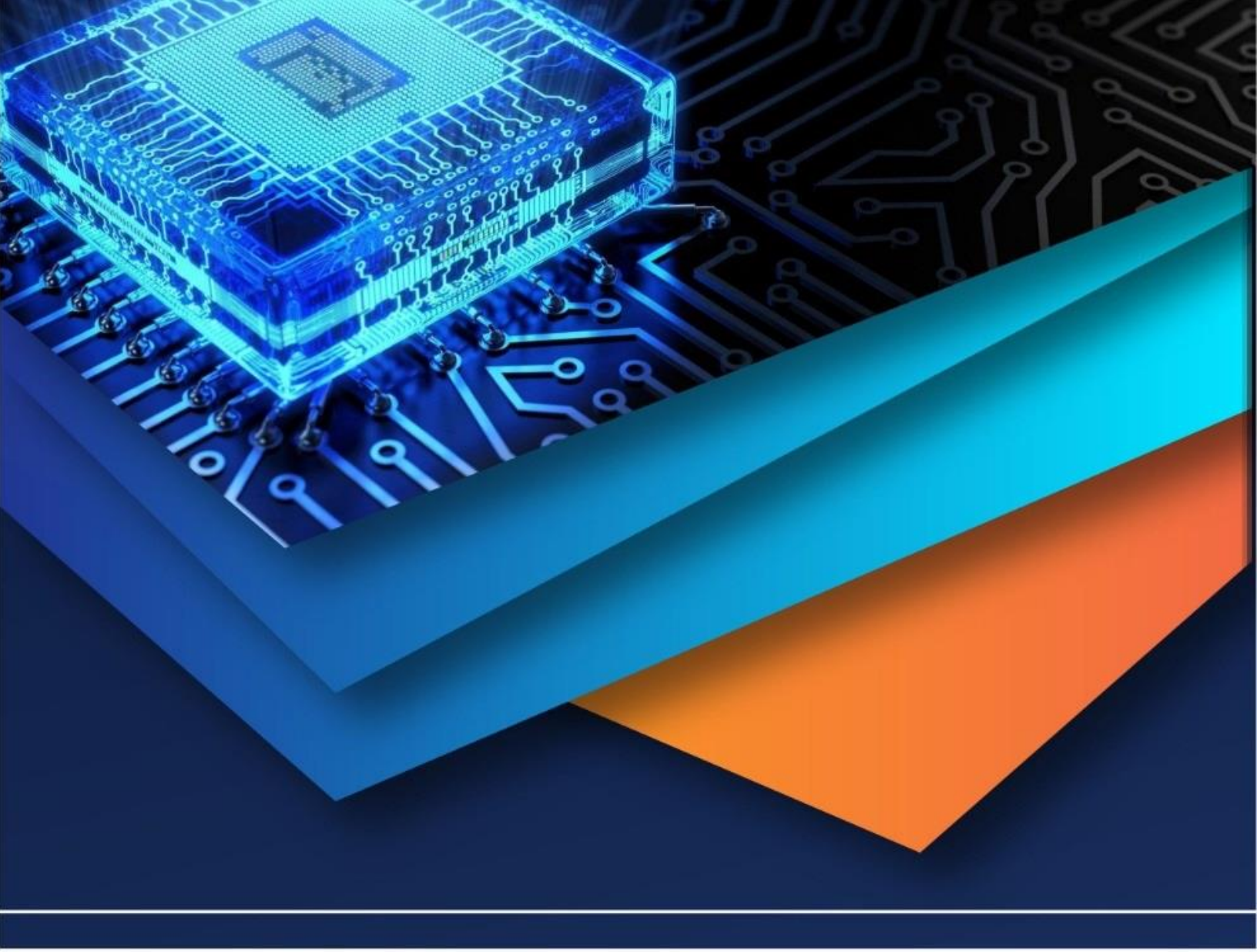
Future iterations will focus on several enhancements: (1) migration of the SQLite database layer to clustered PostgreSQL for centralized corporate management; (2) integration of federated learning to enable collaborative model training across organizations without data sharing; (3) advancement of the video analysis module with transformer-based temporal attention mechanisms.

VIII. ACKNOWLEDGMENT

The authors would like to express sincere gratitude to the Head of the Department and the project supervisor for their continuous guidance and encouragement throughout the development of this system. The support from Anna University and the open-source communities of PyTorch, HuggingFace, FastAPI, and OpenCV is gratefully acknowledged.

REFERENCES

[1] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in Proc. NAACL-HLT, Minneapolis, MN, USA, 2019, pp. 4171–4186.
[2] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in Advances in Neural Information Processing Systems, vol. 27, 2014, pp. 2672–2680.
[3] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 2016, pp. 770–778.
[4] S. M. Lundberg and S. I. Lee, "A unified approach to interpreting model predictions," in Advances in Neural Information Processing Systems, vol. 30, 2017, pp. 4765–4774.
[5] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual explanations from deep networks via gradient-based localization," in Proc. IEEE Int. Conf. Computer Vision (ICCV), Venice, Italy, 2017, pp. 618–626.
[6] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in Advances in Neural Information Processing Systems, vol. 30, 2017, pp. 5998–6008.
[7] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter," in Proc. 5th Workshop on Energy Efficient ML and Cognitive Computing, NeurIPS, 2019.
[8] MITRE Corporation, "MITRE ATT&CK Framework," [Online]. Available: https://attack.mitre.org/. [Accessed: Mar. 2026].
[9] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," in Proc. IEEE Int. Conf. Computer Vision (ICCV), Barcelona, Spain, 2011, pp. 2564–2571.
[10] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3D convolutional networks," in Proc. IEEE Int. Conf. Computer Vision (ICCV), Santiago, Chile, 2015, pp. 4489–4497.
[11] L. Breiman, "Random forests," Machine Learning, vol. 45, no. 1, pp. 5–32, 2001.
[12] S. Raschka, "Model evaluation, model selection, and algorithm selection in machine learning," arXiv preprint arXiv:1811.12808, 2018.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY