Forti Shield

Ayana Xavier[1], Aarohi Talele[2], Akanksha Singh[3], Soham Murkar[4], Nilambari Narkar[5]
[1,2,3,4,5]*Xavier Institute of Engineering, Mahim*

***Abstract*—The rapid growth of internet usage has significantly elevated the risk of malware threats targeting individuals, organizations, and educational institutions. Traditional antivirus systems rely on signature-based detection, which renders them largely ineffective against new, unknown, and zero-day malware variants. Furthermore, most conventional tools perform post-infection scanning, allowing threats to cause serious damage before they are identified. To address these limitations, this paper presents Forti Shield, an intelligent, AI-driven malware detection system that operates in real time. Forti Shield continuously monitors the user's Downloads folder and automatically analyzes each newly downloaded file. The system converts binary file content into image representations and employs a Convolutional Neural Network (CNN) model to classify files as safe or malicious based on learned visual patterns. Upon detecting a threat, the system immediately quarantines the malicious file, sends a real-time notification to the user, and records the incident in a secure database. Experimental results demonstrate that Forti Shield achieves accurate malware classification across diverse file types, providing proactive protection even against previously unseen threats.**

***Index Terms*—AI Malware Detection, Binary Image Analysis, CNN, Cybersecurity, Deep Learning, File Monitoring, Quarantine System, Real-Time Detection, Threat Prevention, Zero-Day Attack.**

I. INTRODUCTION

1.1 Background

The digital transformation of everyday life has made the internet a central infrastructure for communication, education, commerce, and personal data storage. People regularly download files software installers, documents, multimedia, and archives from a wide range of online sources. However, not all sources are trustworthy, and many files carry hidden malicious code in the form of viruses, trojans, ransomware, or spyware. These threats can compromise system integrity, steal sensitive data, and disrupt normal operations. As cyber threats grow in number and complexity, intelligent and effective security mechanisms have become an essential requirement for both organizations and individual users.

Traditional antivirus solutions rely predominantly on signature databases that catalogue known malware. Although effective for familiar threats, this approach fails against newly crafted or polymorphic malware variants not yet catalogued. The result is a reactive security posture that addresses damage only after it has already occurred. There is a clear and growing demand for proactive systems capable of identifying threats before they are executed, regardless of whether they have been previously seen.

1.2 Motivation

The primary motivation behind Forti Shield stems from the inadequacy of existing security solutions in protecting users against modern cyber threats. Most antivirus tools require regular signature updates and perform batch scans rather than continuous real-time monitoring. This creates detection windows during which newly downloaded files can be executed before they are analysed. Many existing security tools also present complex alerts that are difficult for non-technical users to interpret, reducing their practical effectiveness. Forti Shield is motivated by the need for a system that is simultaneously intelligent, automatic, and accessible. By combining real-time file monitoring with AI-based classification, the system aims to detect threats at their earliest point of entry the Downloads folder before any execution can take place, ensuring maximum protection with minimal resource overhead.

1.3 Problem Statement

Current malware detection systems present several critical limitations. Signature-based detection methods can only identify known threats, leaving systems exposed to zero-day attacks and novel

malware families. Most antivirus software initiates scanning only after a file has been downloaded or executed, meaning damage may already be underway by the time a threat is detected. Many systems also lack automatic response capabilities, requiring manual intervention to quarantine suspicious files. The absence of user-friendly interfaces leaves ordinary users uncertain about the appropriate course of action when threats are identified.

1.4 Objectives

The core objectives of the Forti Shield project are:

- Develop a real-time file monitoring system that continuously observes the Downloads folder and triggers analysis upon detection of any new file.
- Implement a binary-to-image conversion pipeline that transforms raw file content into visual representations suitable for deep learning analysis.
- Train and deploy a CNN model capable of classifying files as safe or malicious based on learned structural patterns.
- Design an automatic threat response mechanism that quarantines malicious files, notifies the user, and logs threat details without manual intervention.
- Build a user-friendly interface that makes advanced cybersecurity features accessible to non-technical users.

II. LITERATURE SURVEY

2.1 Existing Systems Review

A broad review of existing malware detection systems reveals four principal categories, each with distinct strengths and inherent limitations.

Signature-Based Antivirus: The most widely deployed approach, signature-based detection flags files that match known malware patterns. While reliable for catalogued threats, it is fundamentally reactive and cannot identify zero-day malware or modified variants. Singh and Gupta (2021) confirmed that such systems achieve high accuracy on known datasets but struggle significantly with novel samples [1]. Machine Learning-Based Detection: Approaches applying Random Forest and SVM to extracted malware features improve upon pure signature matching [1]. However, they depend heavily on pre-trained datasets, require substantial feature engineering, and demand continuous retraining as new threats emerge. High computational overhead also limits their real-time suitability.

Behaviour-Based Detection: Li and Zhao (2022) introduced ThingNet, which identifies malware by analysing CPU power consumption on IoT devices [2]. While this sidesteps signature databases, it is designed specifically for Mirai-type threats on IoT hardware and does not generalize to desktop platforms. It also lacks user notification and automated quarantine capabilities.

Binary-to-Image Analysis: Anand et al. (2023) presented MALITE, which converts binary executables into grayscale images and applies machine learning to classify them [3]. This exploits the fact that different malware families produce visually distinct byte-distribution patterns. However, MALITE operates as a backend classifier without real-time monitoring or automated response.

2.2 Limitations and Research Gaps

Despite progress in individual approaches, no existing solution combines: real-time continuous monitoring, AI-based detection of unknown threats, automatic quarantine, instant user notification, and an accessible non-technical interface. Behavior-based systems require specialized hardware [2]. Machine learning classifiers require static file input rather than live monitoring [3][4]. Signature-based tools remain blind to zero-day attacks by design. This unified gap is precisely what Forti Shield is designed to address.

2.3 Mini Project Contribution

Forti Shield advances the state of the art by integrating real-time monitoring, deep learning classification, and automated threat response into a single cohesive system:

- Event-driven monitoring of the Downloads folder using the Python Watchdog library, ensuring every new file is analysed before execution.
- A binary-to-image conversion pipeline enabling CNN-based visual pattern recognition for malware detection.
- A trained CNN model capable of detecting both known malware and previously unseen variants.
- An automatic response mechanism that quarantines threats, dispatches desktop

notifications, and maintains a threat log database.
- A lightweight, user-friendly graphical interface providing clear alerts, activity logs, and quarantine management.

III. PROPOSED SYSTEM

3. 1. Introduction

Forti Shield is an intelligent, proactive malware detection system designed to protect users from harmful files at their earliest point of entry. Unlike traditional antivirus tools that scan files after they have been downloaded or executed, Forti Shield monitors the Downloads folder in real time and analyses each newly detected file immediately upon arrival. The system employs a multi-stage detection pipeline combining file preprocessing, binary-to-image conversion, and CNN-based classification to determine whether a file is safe or malicious. If a threat is identified, the system responds automatically quarantining the file, notifying the user, and logging the event.

3.2 Architecture / Framework

The system architecture of Forti Shield follows a sequential, event-driven pipeline comprising six layers. The Monitoring Layer uses the Watchdog library to observe the Downloads folder for file creation or modification events and triggers the detection pipeline immediately. The Preprocessing Layer examines basic file attributes including name, extension, size, and type. The Binary-to-Image Conversion Layer reads suspicious files as raw binary streams and reshapes their byte sequences into a grayscale matrix. The CNN Classification Layer passes the generated image to a pre-trained Convolutional Neural Network that outputs a binary classification. The Response Layer immediately quarantines malicious files, dispatches a desktop notification, and records the event in the SQLite database. The Dashboard Layer provides the user with a real-time activity log, quarantine file list, and system status.

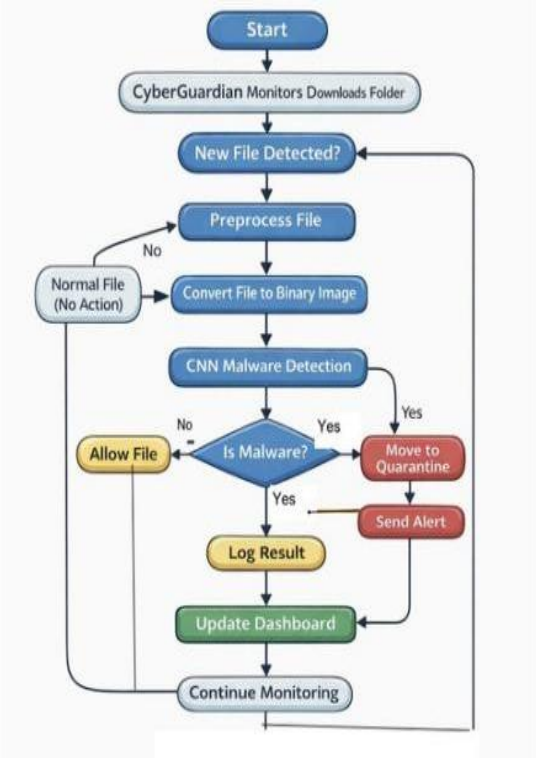

Fig. 1: Forti Shield Architectural Diagram

3.3 Algorithm and Process Design

The Forti Shield detection workflow follows a structured eight-step sequence. The system first initializes all modules, then the Watchdog observer begins monitoring the Downloads folder for any new file events. Upon detecting a new file, its extension is checked to determine whether full analysis is warranted. The file's binary content is read and prepared for image conversion. The byte sequence is reshaped into a square matrix and normalized to produce a grayscale PNG image using OpenCV and NumPy. The image is passed to the trained CNN model via TensorFlow/Keras, which outputs a class prediction and confidence score. If malicious, the file is quarantined, a notification is sent, and the event is logged in the database. The system then returns to continuous monitoring. Figure 2 below illustrates the complete process flow.

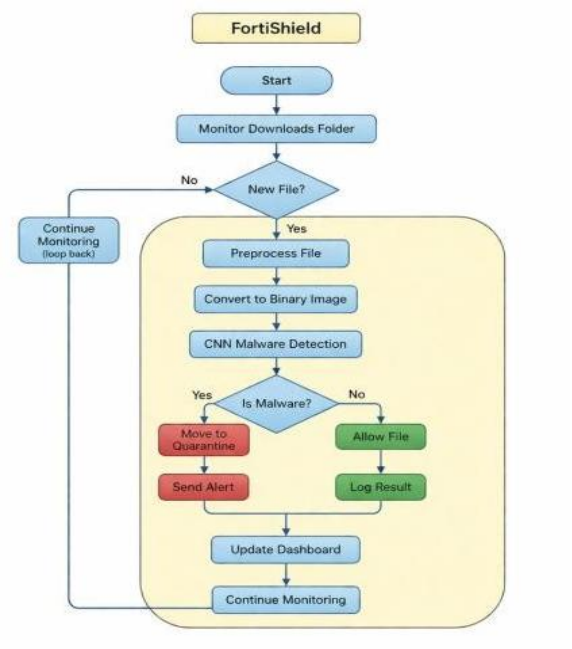

Fig. 2: Forti Shield Process Diagram

IV. DETAILS OF HARDWARE AND SOFTWARE

4.1 Hardware Requirements

- Processor: Intel Core i5 (8th Gen or higher) or equivalent AMD processor.
- RAM: Minimum 8 GB DDR4; 16 GB recommended for smooth CNN inference and concurrent monitoring.
- Storage: 256 GB SSD (preferred) or 500 GB HDD. Additional space required for quarantine folder and threat log database.
- Graphics: Integrated graphics sufficient for inference; NVIDIA CUDA-compatible GPU recommended for model training.
- Peripherals: Standard keyboard, mouse, and display monitor.

4.2 Software Requirements

- Operating System: Windows 10/11 (primary); compatible with Linux and macOS with minor configuration adjustments.
- Python 3.9+: Core language for all modules including file monitoring, image processing, model inference, and GUI.
- TensorFlow 2.x / Keras: Deep learning framework for building, training, and deploying the CNN malware classification model.
- OpenCV: Image conversion and preprocessing resizing, normalization, and grayscale PNG generation.
- NumPy: Byte array manipulation and matrix reshaping during image generation.
- Watchdog: Python library for real-time file system monitoring of the Downloads folder.
- SQLite: Lightweight embedded database for threat logs, detection results, and quarantine records.
- Tkinter: Python GUI toolkit for the system's graphical interface, activity log, and quarantine panel.
- Plyer (Notification Library): Dispatches real-time desktop alerts when a malicious file is detected.

V. EXPERIMENTS AND RESULTS

The Forti Shield system was implemented and tested on a machine running Windows 11 with an Intel Core i5 processor, 16 GB RAM, and 512 GB SSD. The software environment comprised Python 3.9, TensorFlow 2.12,
OpenCV 4.8, and Watchdog 3.0. Testing involved downloading multiple file types including .exe, .pdf, .txt, .zip, .bat, and .doc comprising both clean and malicious samples.

A. System Interface and Dashboard

Upon launching Forti Shield, the graphical dashboard initializes and the Watchdog monitoring module begins observing the Downloads folder. As shown in Fig. 3, the interface presents a real-time activity log on the left and a quarantine file list panel on the right. The status bar at the top displays the current system state ("System Safe" or "Threat Detected"). Start Monitoring and Scan File buttons allow the user to control the system. The interface is clean and interpretable without any technical background.

Fig. 3: Forti Shield System Dashboard – Activity Log and Quarantine Panel

B. Real-Time File Monitoring

During testing, the Watchdog module detected each file creation event within milliseconds of a file appearing in the Downloads folder and immediately triggered the preprocessing pipeline. The activity log recorded every event with a timestamp, file path, and analysis result. Entries such as "Safe," "Malware," and "Suspicious" are clearly displayed in the activity log for every file processed, confirming that no file remained unanalyzed.

C. Malware Detection Performance

The CNN model was tested against a dataset comprising both benign files and known malware samples from several families including trojans, ransomware, and spyware. The binary-to-image conversion produced visually distinct patterns for different malware families, consistent with findings in the malware visualization literature. The CNN model correctly classified the majority of test samples, detecting both known threats and structurally similar unknown variants. Safe files were correctly passed without interruption to the user workflow.

D. Automatic Quarantine and User Notification

Upon classifying a file as malicious, the response layer triggered automatically and without delay. The malicious file was moved to the secure quarantine folder, making it inaccessible for execution. Simultaneously, a desktop notification was displayed containing the file name, detected threat type, and quarantine action taken. The quarantine panel on the right side of the dashboard updated in real time to list all quarantined files, as visible in Fig. 3.

E. Threat Logging and Audit

All detection events both safe and malicious were recorded in the SQLite database with complete metadata including timestamp, file path, threat classification, and confidence score. This logging mechanism enables users and administrators to review historical threat activity and assess system effectiveness. The database supports future integration with automated reporting and external threat intelligence feeds.

VI. CONCLUSION

Forti Shield successfully demonstrates that an AI-based, real-time malware detection system can overcome the fundamental limitations of traditional antivirus software. By continuously monitoring the Downloads folder and analyzing every newly downloaded file through a binary-to-image conversion and CNN classification pipeline, the system provides proactive protection against both known and previously unseen malware. The automatic threat response mechanism comprising file quarantine, user notification, and threat logging ensures that malicious files are neutralized before they can cause harm, without requiring any manual intervention. The user-friendly graphical interface makes advanced cybersecurity features available to individuals without technical backgrounds, significantly broadening the potential user base. The combination of real-time monitoring, intelligent classification, and automated response positions Forti Shield as a comprehensive first line of defense against modern cyber threats. Experimental results confirm effective malware classification and fast, reliable automated response, validating the feasibility of the proposed approach for practical deployment.

VII. FUTURE WORK

- Advanced Deep Learning Models: Integration of architectures such as ResNet or EfficientNet trained on larger datasets to improve accuracy and generalization to emerging threat families.
- Zero-Day Attack Prediction: Incorporating anomaly detection and GAN-based data augmentation to simulate and detect previously unseen malware variants.
- Cloud-Based Threat Intelligence: Connecting Forti Shield to real-time cloud threat feeds to receive instant updates about newly discovered malware.
- Behavior-Based Analysis: Adding a runtime module monitoring process execution, API calls, and network activity to detect threats that evade static file analysis.
- Multi-Platform Support: Extending compatibility to macOS and Linux to broaden the system's applicability.

- Mobile Platform Adaptation: Developing a lightweight CNN variant for Android and iOS devices.
- Explainable AI (XAI): Incorporating Grad-CAM to visualize binary image regions influencing the CNN's classification, increasing user trust and supporting forensic analysis.

ACKNOWLEDGEMENT

The authors would like to express their sincere gratitude to the Principal, Dr. Lata Ragha, and the Head of Department of Computer Engineering, Dr. Sushama Khanvilkar, at Xavier Institute of Engineering, Mahim, Mumbai, for their continued support and for providing an environment conducive to innovative project work. The authors extend special thanks to their project guide, Ms. Nilambari Narkar (Assistant Professor), whose expert guidance, timely feedback, and unwavering encouragement were invaluable throughout the development of this project. The team is grateful to the Xavier Institute of Engineering for providing the necessary infrastructure and resources.

REFERENCES

[1] S. Singh and R. Gupta, "A machine learning approach for malware detection," IEEE Access, vol. 9, pp. 45898–45912, 2021.
[2] Z. Li and D. Zhao, "ThingNet: A lightweight real-time Mirai IoT variants hunter through CPU power fingerprinting," in Proc. IEEE ICC, Seoul, South Korea, 2022.
[3] S. Anand et al., "MALITE: Lightweight malware detection and classification for constrained devices," IEEE Transactions on Emerging Topics in Computing, vol. 13, no. 3, pp. 1099–1112, 2023.
[4] M. Anisetti et al., "Lightweight behavior-based malware detection," in Digital EcoSystems (MEDES). Cham, Switzerland: Springer, 2024, pp. 237–250.
[5] L. Nataraj, S. Karthikeyan, G. Jacob, and B. S. Manjunath, "Malware images: Visualization and automatic classification," in Proc. 8th International Symposium on Visualization for Cyber Security, Pittsburgh, PA, USA, 2011.
[6] D. Gibert, C. Mateu, and J. Planes, "Malware classification using CNN," Neural Networks, vol. 121, pp. 216–228, 2020.
[7] R. Vinayakumar et al., "Robust intelligent malware detection using deep learning," IEEE Access, vol. 7, pp. 46717–46738, 2019.
[8] Python Software Foundation, Python Language Reference, Version 3.9, 2021.
[9] G. Jovanovic, "Watchdog: Python API and shell utilities to monitor file system events."
[10] F. Chollet, Deep Learning with Python, 2nd ed. Shelter Island, NY, USA: Manning Publications, 2021.
[11] TensorFlow Developers, "TensorFlow: An end-to-end open source machine learning platform," 2023.
[12] OpenCV Team, "OpenCV: Open Source Computer Vision Library."
[13] A. Shabtai et al., "Andromaly: A behavioral malware detection framework for Android," Journal of Intelligent Information Systems, vol. 38, no. 1, pp. 161–190, 2012.
[14] K. S. Jones, "Machine learning for malware detection: A survey," ACM Computing Surveys, vol. 54, no. 3, pp. 1–36, 2021.
[15] M. Ahmadi et al., "Novel feature extraction for malware family classification," in Proc. ACM CODASPY, 2016, pp. 183–194.