IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

CyberSentry AI: Web Intrusion Detection and Response System Using AI/ML

Ateeb Shaikh Shrikant Ingale Bhumika Jadhav Khushal Choudhary

Guide: Sangeetha Navale Department of Computer Engineering

Genba Sopanrao Moze College of Engineering, Pune Savitribai Phule Pune University, India

Abstract

The growing reliance on web-based applications and distributed computing environments has significantly increased the exposure of digital systems to sophisticated cyber threats. Conventional intrusion detection systems (IDS) are primarily designed to identify suspicious activities and generate alerts, but they often lack automated response capabilities, resulting in delayed mitigation and increased vulnerabil-ity.

This paper presents CyberSentry AI, an intelligent intrusion detection and response framework that introduces a self-healing approach to cybersecurity. The proposed system integrates file integrity monitoring, deception-based honeypots, and machine learning-driven threat analysis to detect and evaluate malicious activities in real time. By leveraging multiple data sources, including system logs, network behavior, and honeypot interactions, the framework improves detection reliability and reduces false alarms.

A key feature of CyberSentry AI is its automated response mechanism, which enables immediate mitigation actions such as restoring compromised files, blocking suspicious entities, and isolating affected components without requiring manual intervention. This capability significantly reduces response time and limits the impact of attacks.

Experimental evaluation conducted in a controlled environment demonstrates that the proposed system achieves higher detection accuracy, lower false positive rates, and faster response times compared to traditional IDS approaches. The modular and scalable architecture of CyberSentry AI makes it suitable for deployment across both small-scale and enterprise-level environments, contributing toward the development of autonomous and adaptive cybersecurity systems.

1 Introduction

The rapid growth of internet-based services, cloud computing, and web applications has significantly increased the attack surface of modern digital systems. As organizations increasingly rely on interconnected infrastructures, cybersecurity has become a critical concern for ensuring data confidentiality, integrity, and availability. Cyber threats such as ransomware, phishing attacks, distributed denial-of-service (DDoS), SQL injection, cross-site scripting (XSS), and brute-force attacks are becoming more sophisticated, frequent, and difficult to detect using traditional security mechanisms.

Conventional cybersecurity solutions, including firewalls and signature-based intrusion detection systems (IDS), primarily operate using predefined rules and known attack patterns. While these systems are effective against previously identified threats, they fail to detect zero-day attacks and evolving attack techniques.

Additionally, most traditional IDS solutions focus only on alert generation, requiring human intervention for analysis, mitigation, and system recovery. This reactive approach leads to delayed response, increased system downtime, and higher operational costs.

To address these challenges, researchers have explored the use of artificial intelligence (AI) and machine learning (ML) techniques for proactive threat detection. Machine learning models can analyze large volumes of system and network data to identify anomalous behavior and detect previously unknown attacks. Despite these advancements, many existing AI-based security solutions lack integrated response mechanisms and self-healing capabilities, limiting their effectiveness in real-world scenarios.

Another emerging approach in cybersecurity is the use of deception techniques, particularly honeypots, which act as decoy systems to attract and analyze attacker behavior. Honeypots provide valuable insights into attack patterns and techniques; however, they are often deployed as standalone systems and are not fully integrated with intelligent detection and automated response frameworks.

In this context, there is a growing need for an intelligent, autonomous, and integrated cybersecurity system that not only detects intrusions but also analyzes attacker behavior, responds in real time, and restores system integrity without human intervention.

This paper proposes CyberSentry AI, a self-healing web intrusion detection and response system that combines file integrity monitoring, honeypot-based deception, and machine learning-driven threat classification into a unified framework. The system continuously monitors system activities, detects anomalies, analyzes malicious behavior, and executes automated remediation actions such as file restoration, IP blocking, and process isolation.

The main contributions of this work are as follows:

- Design of a modular and scalable self-healing cybersecurity architecture
- Integration of honeypot-based deception with machine learning-based intrusion detection
- Development of an automated response engine for real-time mitigation and recovery
- Reduction of false positives and response time through multi-source data analysis

2 Literature Review

Intrusion Detection Systems (IDS) are essential for protecting modern digital infrastructures from cyber threats. Traditional IDS approaches, particularly signature-based systems, are effective in detecting known attacks but fail to identify zero-day and evolving threats due to their dependence on predefined attack signatures [2]. Anomaly-based IDS address this limitation by identifying deviations from normal behavior, enabling detection of unknown attacks; however, they often suffer from high false-positive rates and require continuous tuning [2].

Machine learning (ML) and deep learning (DL) techniques have significantly improved intrusion detection performance. Various ML algorithms such as Decision Trees, Random Forests, and Support Vector Machines have been widely applied for intrusion detection, demonstrating improved accuracy and efficiency [5]. Deep learning approaches further enhance detection capabilities by capturing complex spatial and temporal patterns in network traffic [1].

Despite these advancements, several challenges persist in ML-based IDS. One major issue is the dependency on high-quality datasets. Poor data quality, including duplication, inconsistency, and mislabeled data, can significantly degrade model performance and reliability [6]. Additionally, the class imbalance problem in intrusion datasets negatively affects the detection of minority attack classes, which are often critical, leading to biased classification results [4].

Another key limitation is the lack of adaptability in traditional IDS. Static systems struggle to cope with evolving attack strategies, resulting in high false-positive rates and reduced efficiency. Research indicates that adaptive and anomaly-based detection systems are more effective in identifying sophisticated and dynamic cyber threats [3].

Table 1: Comparison of Existing Intrusion Detection Approaches

Approach	**Detection Capability**	**Advantages**	**Limitations**	**Response Capability**
Signature-Based IDS	Known attacks only	High accuracy for known threats	Cannot detect zero-day attacks	Manual response required
Anomaly-Based IDS	Unknown + known attacks	Detects novel threats	High false positive rate	Limited or manual response
Machine Learning IDS	Adaptive detection	High accuracy, learns patterns	Requires large datasets, computational cost	Mostly alert-based
Honeypot-Based Systems	Behavioral detection	Captures attacker behavior, detects unknown threats	Limited scope, standalone deployment	No automated response
Hybrid IDS (ML + Honeypot)	Improved detection	Combines Strengths of ML and deception	Complex implementation, lacks recovery system	Partial response capability
CyberSentry AI (Proposed)	Known + Unknown attacks + behavioural detection	Integrated detection, deception, and ML classification	Initial training overhead	**Fully automated self-healing response**

Furthermore, recent studies highlight the effectiveness of ML-based anomaly detection in real-world environments such as industrial control systems (ICS). Techniques such as Convolutional Neural Networks (CNN), Support Vector Machines (SVM), and Isolation Forest have demonstrated strong performance in detecting anomalies and intrusions in real time [1].

Overall, the literature reveals that while ML and DL significantly enhance intrusion detection accuracy, existing systems primarily focus on detection and lack integration with automated response, deception mechanisms, and self-healing capabilities. CyberSentry AI addresses this gap by combining intelligent detection, honeypot-based deception, and automated response into a unified cybersecurity framework.

2.1 Citation Mapping

The contributions of the referenced research papers and their relevance to this work are summarized as follows:

- **IDS Types and Limitations:** Signature-based and anomaly-based IDS limitations are discussed in [2], highlighting issues in detecting zero-day attacks.
- **Machine Learning in IDS:** ML models such as Decision Trees and Random Forest are analyzed in [5], showing improved detection performance.
- **Deep Learning for Intrusion Detection:** Advanced ML/DL techniques for anomaly detection are explored in [1].
- **Data Quality Issues:** The impact of poor data quality on IDS performance is emphasized in [6].
- **Class Imbalance Problem:** Challenges in detecting minority attack classes are addressed in [4].
- **Adaptive and Anomaly-Based Detection:** The importance of adaptive IDS and anomaly detection is discussed in [3].

3 Problem Statement

The increasing complexity and frequency of cyberattacks have exposed significant limitations in traditional cybersecurity solutions. Most existing intrusion detection systems (IDS) are designed primarily to identify suspicious activities and generate alerts, leaving investigation, mitigation, and recovery tasks to human administrators. This reactive security approach leads to delayed response, prolonged system downtime, and increased risk of data breaches and service disruption.

Furthermore, conventional IDS solutions rely heavily on static rules and predefined signatures, making them ineffective against zero-day attacks, evolving malware, and advanced persistent threats. Although anomaly-based and machine learning-driven IDS models improve detection capabilities, they often suffer from high false-positive rates and lack mechanisms for automated decision-making and response.

Another major limitation is the absence of deception techniques in many security systems, allowing attackers to directly target production resources without being diverted or observed. Additionally, existing security solutions do not provide self-healing capabilities, requiring manual restoration of compromised files, services, and configurations after an attack.

These challenges highlight the need for an intelligent and autonomous cybersecurity framework that integrates intrusion detection, attacker deception, intelligent threat classification, and automated response and recovery. The objective of CyberSentry AI is to address these limitations by providing a self-healing intrusion detection and response system that minimizes human intervention while enhancing system resilience and security.

4 Proposed System Architecture

CyberSentry AI is designed using a modular, layered, and scalable architecture to ensure effective intrusion detection, intelligent analysis, and automated response. The architecture follows a pipeline-based processing model in which security events are continuously monitored, analyzed, classified, and mitigated in real time. Each module operates independently while securely exchanging data with other components, enabling flexibility, fault tolerance, and ease of system expansion.

The architectural design emphasizes automation, minimal human intervention, and rapid recovery, making it suitable for protecting modern web-based environments. Figure 1 illustrates the overall system architecture of CyberSentry AI and the interaction between its core components.

4.1 Architecture Overview

The architecture consists of four primary layers: monitoring, analysis, intelligence, and response. The monitoring layer collects system and network data, the analysis layer extracts relevant features, the intelligence layer performs machine learning-based threat classification, and the response layer executes automated mitigation and recovery actions. This layered approach ensures clear separation of responsibilities and improves system maintainability.

Table 2: Core System Modules

Module	Functionality
File Integrity Monitor	Detects unauthorized file changes using cryptographic hashes
Honeypot Engine	Lures attackers and records malicious interactions
ML Detection Engine	Classifies events as benign or malicious
Response Engine	Executes automated recovery actions

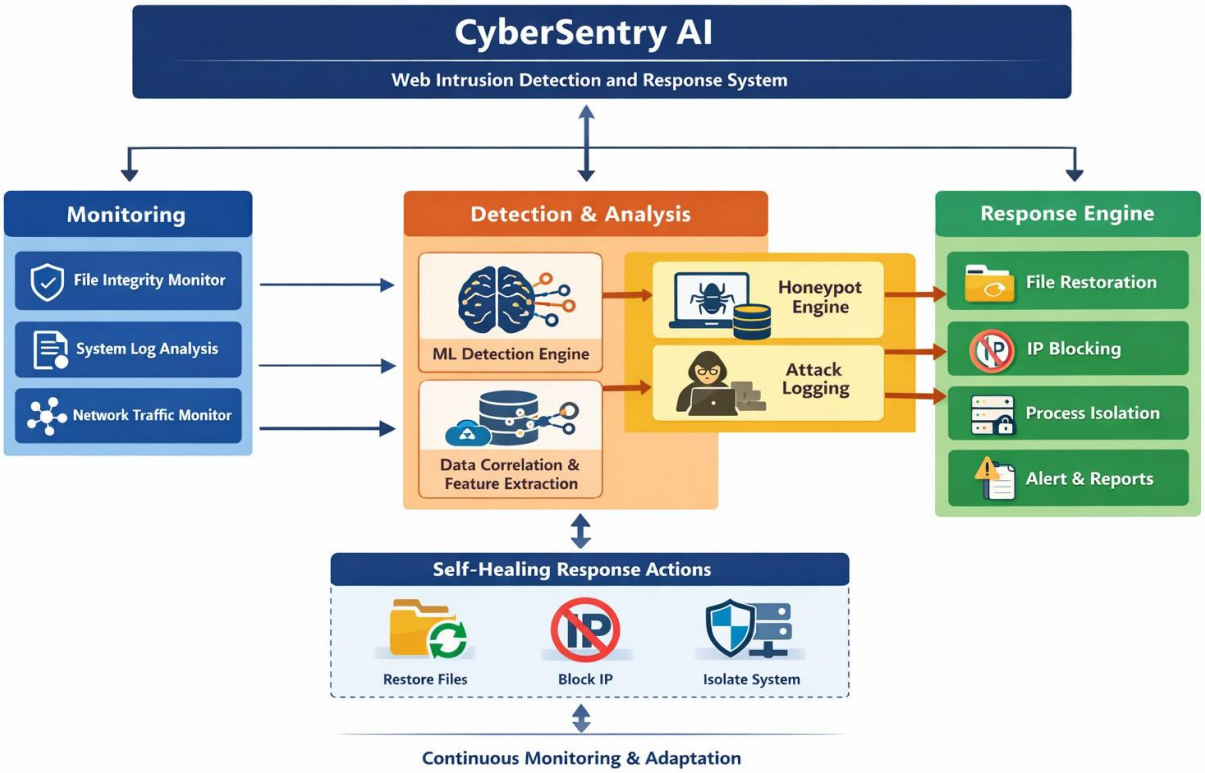

Figure 1: Overall System Architecture of CyberSentry AI

4.2 File Integrity Monitoring Module

The File Integrity Monitoring (FIM) module is responsible for ensuring the integrity of critical system and application files. It computes cryptographic hash values using the SHA-256 algorithm and stores baseline hashes in a secure repository. Periodic comparisons are performed to detect unauthorized file modifications caused by malware, insider threats, or configuration tampering.

Any detected deviation triggers an alert and forwards the event to the analysis module for further investigation. This module provides reliable low-level indicators of compromise and serves as an essential input source for the machine learning engine.

4.3 Threat Detection and Classification Module

The threat detection and classification module aggregates data from multiple sources, including system logs, network traffic, file integrity alerts, and honeypot interaction records. Feature extraction techniques are applied to convert raw event data into structured numerical features suitable for machine learning models.

Supervised learning algorithms, such as Decision Trees and Random Forest classifiers, analyze these features to classify events as benign or malicious. Correlating multiple data sources significantly reduces false positives and improves robustness against sophisticated and multi-stage attacks.

4.4 Self-Healing Response Engine

The self-healing response engine is responsible for executing automated remediation actions once a threat is confirmed. Based on predefined response policies, the engine performs actions such as restoring modified files from secure backups, blocking malicious IP addresses, terminating suspicious processes, and isolating compromised services.

By automating response and recovery, CyberSentry AI minimizes system downtime, prevents attack propagation, and restores normal operations without requiring manual intervention. This self-healing capability significantly enhances overall system resilience and reliability.

5 Methodology

The methodology of CyberSentry AI is designed as a continuous, adaptive, and closed-loop security framework that integrates real-time monitoring, intelligent threat detection, deception mechanisms, and automated self-healing response. The system follows a structured pipeline approach to ensure efficient processing of security events and rapid mitigation of detected threats.

5.1 Overview of Methodology

The overall methodology consists of five major stages: data collection, anomaly detection, deception and behavioral analysis, machine learning-based classification, and automated response. These stages operate sequentially while maintaining continuous feedback to improve detection accuracy and system resilience.

5.2 Data Collection and Monitoring

The first stage involves continuous monitoring of system activities and network behavior. Data is collected from multiple sources, including:

- System logs (authentication attempts, process execution)
- File system changes (modification, deletion, creation events)
- Network traffic patterns (IP requests, packet frequency)

This multi-source data collection ensures comprehensive visibility into system operations and forms the foundation for accurate intrusion detection.

5.3 Anomaly Detection

In the second stage, collected data is analyzed using rule-based techniques to identify abnormal behavior. Predefined thresholds and heuristic rules are applied to detect suspicious patterns such as repeated failed login attempts, unusual file modifications, or abnormal traffic spikes.

This preliminary filtering helps reduce noise and computational overhead by eliminating clearly benign activities before applying more complex analysis.

5.4 Deception and Honeypot Analysis

Suspicious activities identified in the anomaly detection phase are further analyzed using honeypots. The system deploys decoy services and files that mimic real assets to attract attackers. When an attacker interacts with these honeypots, detailed logs are generated, capturing behavior such as access patterns, payload characteristics, and interaction frequency.

This stage provides valuable behavioral insights and helps in identifying attacker intent without exposing critical system resources.

5.5 Feature Extraction and Machine Learning Classification

In this stage, raw data collected from logs and honeypot interactions is transformed into structured feature vectors. Key features include:

- Login attempt frequency
- File hash deviation indicators
- Request entropy and payload size
- IP request rate and geographic anomalies
- Honeypot interaction depth

These features are input into supervised machine learning models such as Decision Trees and Random Forest classifiers. The models classify each event as either benign or malicious based on learned patterns from the training dataset.

5.6 Automated Response and Self-Healing

Once a threat is classified as malicious, the system triggers the self-healing response engine. Based on predefined policies, the system automatically performs mitigation actions such as:

- Blocking malicious IP addresses
- Restoring modified files from secure backups
- Terminating suspicious processes
- Isolating affected services or system components

This automated response significantly reduces response time and prevents the spread of attacks.

5.7 Feedback and Continuous Learning

The final stage involves continuous feedback and system adaptation. The outcomes of detection and response actions are logged and used to refine detection rules and retrain machine learning models. This enables CyberSentry AI to adapt to evolving attack patterns and improve performance over time.

Overall, the proposed methodology ensures a proactive, intelligent, and autonomous cybersecurity system capable of detecting, analyzing, and mitigating threats with minimal human intervention.

5.8 Algorithm for Intrusion Detection and Response

Input: System logs, network traffic data, file integrity data, honeypot logs
Output: Classified threat (Benign/Malicious) and automated response action

1. Start
2. Continuously monitor system logs, file changes, and network traffic
3. Collect and store incoming data in the database
4. Apply rule-based anomaly detection:
    - If activity is normal → Continue monitoring
    - Else → Mark as suspicious event
5. Redirect suspicious activity to honeypot module
6. Capture attacker interaction data
7. Perform feature extraction:
    - Login frequency
    - File hash deviation
    - Request entropy
    - IP request rate
8. Input features into ML classifier (Random Forest / Decision Tree)
9. Classify event:
    - If Benign → Log and continue monitoring
    - If Malicious → Trigger response engine

10. Execute automated response:
    - Block IP address
    - Restore modified files
    - Isolate affected system
11. Store results for future learning
12. Repeat process (continuous loop)
13. End

5.9 Flowchart Explanation

The workflow of CyberSentry AI follows a structured sequence of operations, as illustrated in Figure 2.

The process begins with continuous monitoring of system activities, including log analysis, file integrity checks, and network traffic observation. This data is analyzed using rule-based anomaly detection techniques to identify suspicious behavior.

If no anomaly is detected, the system continues normal monitoring. However, if suspicious activity is identified, it is redirected to the honeypot module for deeper behavioral analysis. The honeypot captures attacker interactions in a controlled environment, preventing direct impact on real system resources.

The collected data is then processed through feature extraction, where relevant attributes such as request patterns, file changes, and interaction depth are converted into structured inputs for machine learning models. These features are analyzed using trained classifiers to determine whether the activity is benign or malicious.

If the activity is classified as benign, the system logs the event and resumes monitoring. If classified as malicious, the automated response engine is activated. The system executes predefined mitigation actions such as blocking the attacker's IP address, restoring modified files, and isolating affected components.

Finally, the system updates its logs and learning models, ensuring continuous improvement and adaptation to new attack patterns. This cyclic workflow enables CyberSentry AI to provide real-time, autonomous, and adaptive cybersecurity protection.

6 Implementation Details

CyberSentry AI is implemented using a modular software architecture to ensure flexibility, scalability, maintainability, and ease of deployment. The system is developed using TypeScript across both the frontend and backend, providing strong type safety, improved code reliability, and consistent development practices. The architecture enables seamless integration between the user interface, backend services, database, and AI-powered security modules.

The frontend is developed using React with TypeScript and Tailwind CSS, providing a responsive, modern, and interactive dashboard for real-time security monitoring. React's component-based architecture simplifies the development of reusable user interface elements, while Tailwind CSS enables rapid creation of a clean and responsive design optimized for security operations.

The backend is implemented using Deno TypeScript with Supabase Edge Functions, which provide secure, serverless execution of security workflows. The backend is responsible for processing incoming security events, executing detection logic, interacting with the database, and triggering automated response mechanisms. The lightweight and event-driven nature of Edge Functions enables low-latency processing and efficient scalability.

PostgreSQL, managed through Supabase, serves as the primary database for storing security events, attack logs, file integrity records, machine learning classification results, threat intelligence data, and automated response actions. PostgreSQL offers high reliability, ACID compliance, and advanced querying capabilities, making it suitable for real-time cybersecurity applications.

The intelligent threat detection module integrates machine learning frameworks such as TensorFlow, PyTorch, and Scikit-learn (Sklearn) to perform anomaly detection, attack classification, behavioral analysis, and predictive threat assessment. These frameworks support the development, training, and deployment of AI models capable of identifying both known and previously unseen attack patterns.

The implementation follows a modular design in which the frontend, backend, database, AI engine, and security response modules operate independently through well-defined interfaces. This architecture simplifies testing, maintenance, and future enhancements, allowing new detection algorithms, response strategies, or machine learning models to be integrated without affecting the overall stability and performance of the system.

Table 3: Technology Stack

Component	Technology
Programming Language	Type script, React, Tailwind CSS
Backend Framework	Deno edge function (Supabase Edge Functions)
ML Library	TensorFlow, PyTorch, Sklearn
Database	PostgreSQL (via Supabase)
OS	Windows , Linux

6.1 Machine Learning Models

CyberSentry AI utilizes supervised machine learning algorithms for threat classification. Decision Tree and Random Forest classifiers are selected due to their interpretability, efficiency, and ability to handle structured security datasets. These models are particularly suitable for intrusion detection tasks where understanding feature importance and decision paths is essential.

Feature vectors are constructed using extracted attributes such as login attempt frequency, file hash deviation indicators, honeypot interaction depth, request entropy, and IP request rates. The dataset is divided into training and testing sets using an 80:20 split to evaluate generalization performance.

Model evaluation focuses on metrics such as detection accuracy, false positive rate, and response latency. The selected models demonstrate a balanced trade-off between accuracy and computational overhead, making them suitable for real-time intrusion detection and response.

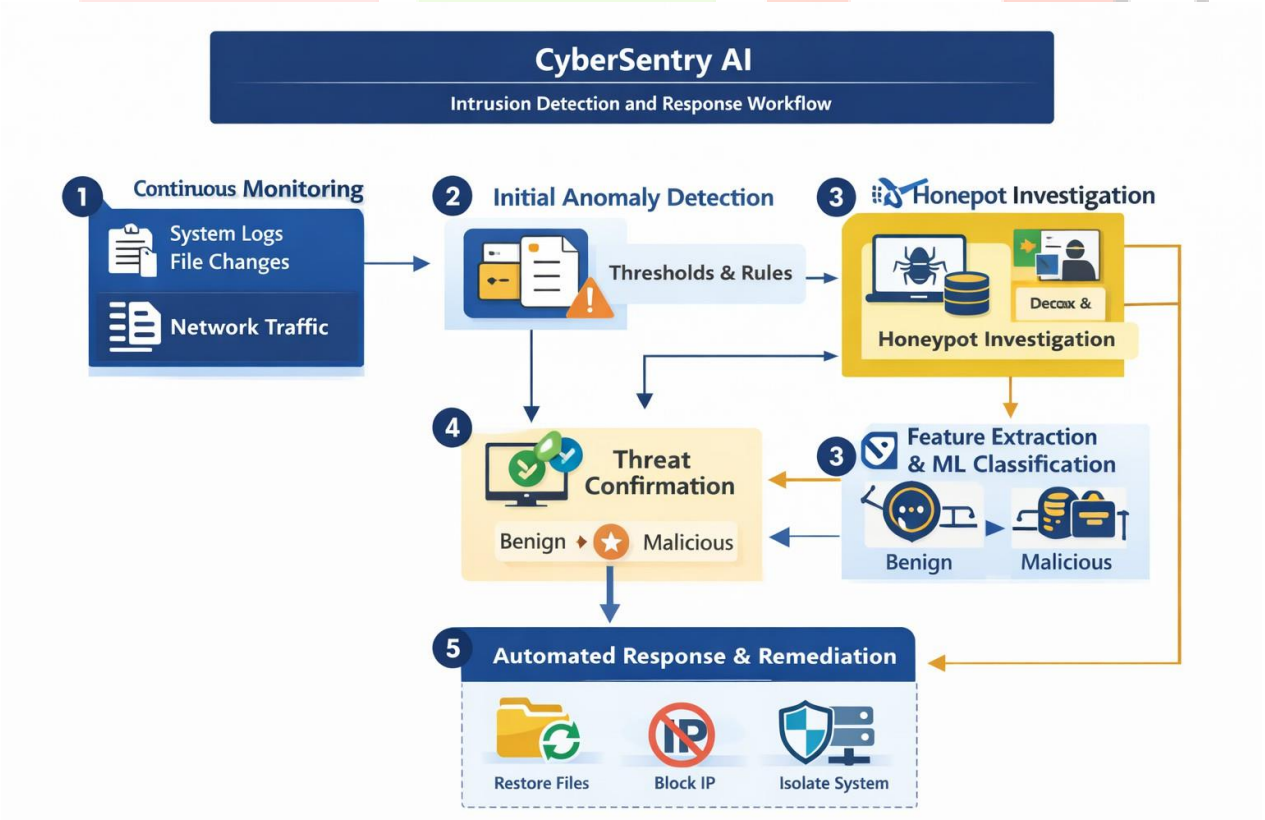

Figure 2: CyberSentry AI Workflow

7 Results and Analysis

The performance of CyberSentry AI was evaluated in a controlled Linux-based environment using a variety of simulated attack scenarios. These scenarios include brute-force authentication attempts, unauthorized file modifications, and malicious interactions with deployed honeypots. The objective of the evaluation was to assess the effectiveness of the proposed system in terms of detection accuracy, false positive rate, and response time.

To ensure a fair comparison, CyberSentry AI was evaluated against a traditional intrusion detection system that relies primarily on rule-based detection and manual response. The same attack scenarios were applied to both systems under identical conditions.

Detection accuracy measures the system's ability to correctly identify malicious activities, while the false positive rate indicates the proportion of normal events incorrectly classified as attacks. Response time refers to the duration between threat detection and execution of mitigation actions.

The experimental results demonstrate that CyberSentry AI achieves significantly higher detection accuracy due to its machine learning-based classification and multi-source data correlation. The integration of honeypot interaction data further enhances the system's ability to distinguish between benign and malicious behavior, leading to a substantial reduction in false positives.

One of the most notable improvements is observed in response time. The automated self-healing response engine enables immediate execution of mitigation actions, reducing response time from several seconds or minutes to near real-time. This rapid response capability effectively limits attack propagation and minimizes system downtime.

Overall, the results confirm that CyberSentry AI outperforms traditional IDS solutions by providing faster, more accurate, and autonomous intrusion detection and response. The evaluation highlights the practical benefits of integrating artificial intelligence, deception mechanisms, and self-healing strategies into modern cybersecurity systems.

7.1 Confusion Matrix Analysis

The confusion matrix provides a detailed evaluation of the classification performance of CyberSentry AI. It represents the number of correctly and incorrectly classified instances for both normal and malicious activities. True Positives (TP) indicate correctly detected attacks, while True Negatives (TN) represent correctly identified normal activities. False Positives (FP) correspond to normal events incorrectly classified as attacks, and False Negatives (FN) represent missed attack detections.

The results show that the system achieves a high number of true positives and true negatives, indicating strong classification capability. The relatively low number of false positives demonstrates the effectiveness of integrating honeypot data with machine learning models, which reduces unnecessary alerts and improves system reliability.

Overall, the confusion matrix confirms that CyberSentry AI maintains a balanced performance between detection accuracy and false positive reduction, which is critical for real-world deployment.

The results demonstrate significant improvements in detection accuracy and response speed due to AI-driven classification and automated remediation.

Table 4: Confusion Matrix of CyberSentry AI (Random Forest)

	Predicted Normal	**Predicted Attack**
Actual Normal	2900	180
Actual Attack	240	3680

8 Conclusion

This paper presented CyberSentry AI, an intelligent web intrusion detection and response system designed to address the limitations of traditional cybersecurity solutions. Unlike conventional intrusion detection systems that primarily focus on alert generation, CyberSentry AI integrates machine learning-based threat classification, deception through honeypots, and automated self-healing mechanisms to provide comprehensive protection against modern cyber threats.

The proposed framework effectively combines file integrity monitoring, behavioral analysis, and multi-source data correlation to achieve accurate and timely intrusion detection. Experimental evaluation demonstrates that CyberSentry AI significantly improves detection accuracy while reducing false positives and re-sponse time when compared to traditional IDS approaches. The automated response and recovery capabilities ensure minimal system downtime and reduce reliance on human intervention.

The modular and scalable architecture of CyberSentry AI enables easy deployment across diverse environments, ranging from small-scale systems to enterprise-level infrastructures. By enhancing system resilience and operational efficiency, the proposed solution contributes toward the development of autonomous and intelligent cybersecurity systems capable of addressing evolving threat landscapes.

Overall, CyberSentry AI demonstrates the practical viability of self-healing security frameworks and establishes a foundation for future research in intelligent, automated cyber defense mechanisms.

9 Future Scope

Although CyberSentry AI demonstrates effective intrusion detection and automated response capabilities, several enhancements can further improve its performance, scalability, and applicability in real-world environments.

One significant future enhancement involves deploying the system in cloud-native environments using containerization technologies such as Docker and Kubernetes. This would enable dynamic scaling, high availability, and seamless integration with modern cloud infrastructures. Cloud-based deployment would also allow centralized monitoring and protection of distributed web applications.

The integration of deep learning models, such as Long Short-Term Memory (LSTM) networks and Autoencoders, can further enhance the detection of complex and zero-day attacks. Deep learning techniques can capture long-term temporal patterns in network traffic and system logs, improving detection accuracy for sophisticated multi-stage attacks.

Another promising direction is the incorporation of blockchain technology to ensure the integrity and immutability of security logs. Storing critical logs and alerts on a blockchain can prevent tampering and provide reliable forensic evidence for post-incident investigations and compliance requirements.

Future versions of CyberSentry AI can also include adaptive learning mechanisms, where the system continuously retrains its machine learning models based on newly observed attack patterns. This would enable the framework to evolve alongside emerging threats and reduce the need for manual model updates.

Additionally, integrating CyberSentry AI with Security Information and Event Management (SIEM) systems and threat intelligence feeds can enhance situational awareness and collaborative defense strategies. Large-scale enterprise testing and real-world deployment will further validate the system's effectiveness and robustness under diverse attack scenarios.

Overall, these enhancements will transform CyberSentry AI into a fully autonomous, scalable, and intelligent cybersecurity platform capable of addressing future security challenges.

References

[1] D. Benka, D. Horváth, L. Špendla, G. Gasˇpar, and M. Strémy, "Machine Learning-Based Detection of Anomalies, Intrusions, and Threats in Industrial Control Systems," IEEE Access, vol. 13, pp. 12502–12514, Jan. 2025, doi: 10.1109/ACCESS.2025.3530902.

[2] I. Rakine et al., "Comprehensive Review of Intrusion Detection Techniques: ML and DL in Different Networks," IEEE Access, vol. 13, pp. 123456-123478, 2025, doi: 10.1109/ACCESS.2025.3579990.

[3] Ç evik, N., and Akleylek, S., "SoK of Machine Learning and Deep Learning Based Anomaly Detection Methods for Automatic Dependent Surveillance-Broadcast," IEEE Access, vol. 12, pp. 35643–35662, 2024, doi: 10.1109/ACCESS.2024.3369181.

[4] S. M. H. Mirsadeghi, H. Bahsi, R. Vaarandi, and W. Inoubli, "Learning from few cyber-attacks: Addressing the class imbalance problem in machine learning-based intrusion detection in software-defined networking," IEEE Access, vol. 11, pp. 140428–140442, Dec. 2023, doi: 10.1109/ACCESS.2023.3341755.

[5] Azam, Z., Islam, M. M., and Huda, M. N., "Comparative Analysis of Intrusion Detection Systems and Machine Learning-Based Model Analysis Through Decision Tree," IEEE Access, vol. 11, pp. 80348–80391, 2023, doi: 10.1109/ACCESS.2023.3296444.

[6] N. Tran, H. Chen, J. Bhuyan, and J. Ding, "Data curation and quality evaluation for machine learning-based cyber intrusion detection," IEEE Access, vol. 10, pp. 121900–121922, Oct. 2022, doi: 10.1109/ACCESS.2022.3211313.