International Journal of Engineering Research and Science & Technology

www.ijerst.org

ISSN : 2319-5991

Vol. 22 No. 2 (2026)



ijerst.editor@gmail.com

editor@ijerst.com

International Journal of Engineering Research and Science & Technology

ISSN 2319-5991 www.ijerst.com
Vol. 22, No. 2, 2026
© 2026 IJERST. All Rights Reserved
Peer Reviewed, Referred & Indexed Journal

AI-Driven Cyber security Threat Monitoring and Network Intelligence Dashboard Using Django

PASUPULETI SURYA RAMYA SUJITHA NAGA LAKSHMI

PG Scholar. Department of MCA, DNR College, Bhimavaram, Andhra Pradesh

K. Venkatesh

(Assistant Professor), Master of Computer Applications, DNR College, Bhimavaram, Andhra Pradesh

ABSTRACT

In the modern digital era, the rapid expansion of networked systems has significantly increased the risk of cyber threats, making cyber security a critical concern for organizations. Traditional security monitoring systems often fail to provide real-time insights and actionable intelligence required to mitigate sophisticated attacks. This project presents the design and implementation of an AI-driven cyber security threat monitoring and network intelligence dashboard using the Django web framework. The proposed system integrates real-time data visualization, event tracking, and network performance monitoring into a unified dashboard. It is designed to detect, classify, and manage security threats such as Distributed Denial of Service (DDoS) attacks, SQL injection attempts, and brute-force login activities. The system employs structured data models including Security Event, Network Metric, and Business Process to organize and analyze security-related information effectively.

The dashboard provides administrators with an intuitive interface to monitor recent security events, assess network health, and evaluate the operational status of critical business processes. Threat levels are dynamically computed based on unresolved high-severity events, enabling proactive decision-making. Additionally, simulated network metrics such as traffic flow, latency, and packet loss offer insights into network performance and potential anomalies. One of the key features of the system is its ability to categorize threats by severity levels (Low, Medium, High, and Critical) and allow administrators to resolve incidents efficiently. The integration of seed data generation ensures that the system remains functional and demonstrative even in the absence of real-time data sources. The use of Django provides a robust and scalable backend, while its built-in ORM simplifies database interactions. The system is designed with modularity and extensibility in mind, allowing future integration with machine learning models for predictive threat detection and anomaly analysis. This project demonstrates how web-based technologies can be leveraged to build intelligent cyber security solutions. It highlights the importance of real-time monitoring, centralized management, and data-driven decision-making in safeguarding digital infrastructure. The proposed system not

only enhances situational awareness but also reduces response time to potential threats, making it a valuable tool for modern cyber security operations.

KEYWORDS: Cyber security, Threat Detection, Django Framework, Network Monitoring, Intrusion Detection, Security Events, Real-time Analytics, Web Dashboard, Risk Assessment, Data Visualization

I. INTRODUCTION

With the exponential growth of internet-based services and digital transformation across industries, cyber security has emerged as one of the most critical challenges of the 21st century. Organizations today rely heavily on interconnected systems, cloud computing, and online platforms, which increases their exposure to cyber threats such as hacking, malware, phishing, and denial-of-service attacks. Traditional security systems often operate in isolation and lack real-time monitoring capabilities, making it difficult to identify and respond to threats promptly. Moreover, the increasing complexity of cyber attacks requires intelligent systems that can not only detect threats but also provide actionable insights. This has led to the development of advanced cyber security dashboards that combine monitoring, analytics, and visualization. This project focuses on developing a Cyber security Threat Monitoring and Network Intelligence Dashboard using the Django framework. Django, a high-level Python web framework, is known for its scalability, security features, and rapid development capabilities. The system aims to provide a centralized platform where administrators can monitor security events, analyze network metrics, and evaluate the health of business processes.

The core components of the system include:

- **Security Event Management**: Tracks various types of cyber threats and categorizes them based on severity.
- **Network Monitoring**: Provides insights into network performance through metrics such as traffic, latency, and packet loss.
- **Business Process Monitoring**: Evaluates the operational status and security posture of critical services.

The system dynamically calculates threat levels based on active unresolved incidents, allowing organizations to prioritize responses. It also includes features for filtering, resolving, and managing security events efficiently. One of the key motivations behind this project is to simplify cyber security management through an intuitive and user-friendly interface. By presenting complex data in a structured and visual format, the system enhances situational awareness and supports faster decision-making. Furthermore, the modular design of the system allows for future enhancements such as integration with machine learning algorithms for predictive analytics, anomaly detection, and automated threat response. This makes the system not only relevant for current needs but also

adaptable to future cyber security challenges. In conclusion, this project addresses the growing need for intelligent cyber security solutions by combining web technologies, data modeling, and real-time analytics into a comprehensive dashboard.

II. LITERATURE SURVEY (WITH EXISTING METHODS)

Cyber security has been an active area of research, with numerous approaches developed to detect and prevent cyber threats. Traditional intrusion detection systems (IDS) are broadly classified into signature-based and anomaly-based systems. Signature-based systems rely on predefined patterns to detect known threats but fail to identify new or unknown attacks. Anomaly-based systems, on the other hand, use statistical and machine learning techniques to detect deviations from normal behavior. Recent studies have explored the use of machine learning algorithms such as Random Forest, Support Vector Machines (SVM), and Neural Networks for intrusion detection. These methods have shown improved accuracy in identifying complex attack patterns. However, they often require large datasets and high computational resources, making real-time implementation challenging. Web-based dashboards have also gained popularity as tools for cyber security monitoring. Systems like Security Information and Event Management (SIEM) platforms integrate data from multiple sources and provide real-time analysis. While effective, these systems are often expensive and complex to deploy. Research has also emphasized the importance of data visualization in cyber security. Visual dashboards help in understanding large volumes of data quickly and identifying patterns or anomalies. Tools like Kibana and Grafana are widely used for this purpose, but they require integration with external data pipelines. Another area of research focuses on network monitoring and traffic analysis. Techniques such as deep packet inspection and flow analysis are used to identify suspicious activities. However, these methods may raise privacy concerns and require specialized hardware.

The use of web frameworks like Django in cyber security applications has been explored due to their flexibility and security features. Django's built-in authentication system, ORM, and scalability make it suitable for developing secure web applications. Despite these advancements, there is still a need for lightweight, customizable, and cost-effective solutions that can provide real-time insights without requiring complex infrastructure. This project addresses this gap by developing a Django-based dashboard that integrates threat monitoring, network analysis, and business process evaluation into a single platform.

III. EXISTING SYSTEM

Existing cyber security systems primarily rely on standalone tools or enterprise-level platforms such as SIEM (Security Information and Event Management) systems. These systems collect and analyze data from various sources, including network devices, servers, and applications, to detect potential threats. While effective, these systems have several limitations. Firstly, they are often expensive and require significant infrastructure

and expertise to deploy and maintain. Small and medium-sized organizations may find it difficult to adopt such solutions due to cost constraints. Secondly, many traditional systems lack real-time visualization capabilities. Although they generate logs and reports, interpreting this data can be time-consuming and requires skilled personnel. This delays response times and increases the risk of damage. Thirdly, existing systems may not provide a unified view of security events, network performance, and business processes. This fragmentation makes it difficult to correlate data and identify the root cause of issues.Additionally, most systems rely heavily on manual intervention for threat resolution. Automated features are limited, and integrating advanced technologies such as machine learning often requires additional tools and configurations. Another major drawback is the lack of customization. Many commercial solutions are rigid and do not allow organizations to tailor the system according to their specific needs. In contrast, the proposed system offers a lightweight, cost-effective, and customizable solution. It provides real-time monitoring, intuitive visualization, and centralized management of security events, making it more accessible and efficient for modern cyber security requirements.

IV. PROPOSED METHOD

The proposed system is an AI-driven Cyber security Threat Monitoring and Network Intelligence Dashboard developed using the Django web framework. It is designed to provide real-time monitoring, analysis, and management of security events, network metrics, and business processes in a unified platform. The system introduces a centralized architecture where all security-related data is collected, processed, and visualized. It uses structured models such as Security Event, Network Metric, and Business Process to maintain organized and scalable data storage. The system continuously tracks incoming events such as DDoS attacks, SQL injections, and unauthorized access attempts, categorizing them into severity levels (Low, Medium, High, and Critical). A dynamic threat assessment mechanism is implemented, where the overall threat level is automatically calculated based on unresolved high-severity events. This allows administrators to quickly identify critical issues and prioritize responses. Additionally, the system provides filtering capabilities to view resolved, unresolved, or all threats, improving incident management efficiency.

The network monitoring module analyzes traffic metrics such as inbound and outbound traffic, latency, and packet loss. These metrics help detect anomalies and performance degradation in real time. Simulated geolocation-based traffic analysis is also included to identify suspicious sources. The proposed system emphasizes usability by offering an interactive dashboard with intuitive data visualization. It also supports administrative actions such as resolving incidents and monitoring system health. Furthermore, the architecture is designed to be extensible, allowing integration with machine learning algorithms for anomaly detection and predictive analytics. Recent research highlights the growing importance of AI-based intrusion detection systems in handling complex cyber

threats efficiently. Overall, the proposed system aims to deliver a cost-effective, scalable, and intelligent cyber security solution suitable for modern organizations.

V. IMPLEMENTATION

The implementation of the Cyber security Dashboard is carried out using the Django framework, which provides robust backend architecture and efficient database handling through its Object-Relational Mapping (ORM) system. The system is divided into multiple modules, each responsible for a specific functionality. The core modules include the dashboard module, threat management module, network monitoring module, and settings module. The dashboard module serves as the central interface, displaying recent security events, network metrics, and business process statuses. When the system is initialized, a seed_data() function populates the database with sample data to ensure functionality even in the absence of real-time inputs. This includes generating security events and network metrics using random values. The threat management module allows users to view, filter, and resolve security incidents. It supports multiple filters such as resolved, unresolved, and all events. Each event is stored with attributes like timestamp, severity, and source IP address. The resolve_event function enables administrators to mark threats as resolved, updating the system in real time. The network monitoring module tracks performance metrics such as inbound traffic, outbound traffic, latency, and packet loss. These metrics are displayed in a structured format, enabling administrators to identify anomalies. Additionally, simulated traffic sources with geolocation data provide insights into potential external threats.

The system uses Django models to define database structures. The Security Event model captures threat-related data, while the Network Metric model records performance metrics. The Business Process model tracks the status and security score of critical services. The admin panel is configured to allow easy management of data. Features such as filtering, searching, and inline editing improve usability and administrative efficiency. From a technical perspective, the system follows the Model-View-Template (MVT) architecture. Views handle business logic, models manage data, and templates render the user interface. This separation ensures maintainability and scalability. The implementation also considers future integration with machine learning models. Studies show that combining machine learning with intrusion detection improves detection accuracy and reduces false positives. Overall, the implementation demonstrates how modern web technologies can be effectively utilized to build a scalable cyber security monitoring system.

VI. ALGORITHMS

The proposed system utilizes a combination of rule-based logic and data-driven approaches for threat detection and analysis. While the current implementation focuses

on simulation and basic logic, it is designed to support integration with advanced machine learning algorithms. The primary algorithm used in the system is a rule-based threat classification mechanism. Security events are categorized based on predefined severity levels such as Low, Medium, High, and Critical. The system continuously checks for unresolved high-severity events to determine the overall threat level. Another important component is anomaly detection in network metrics. The system monitors parameters such as traffic volume, latency, and packet loss. Sudden deviations from normal ranges can indicate potential cyber threats such as DDoS attacks or network intrusions. In future enhancements, machine learning algorithms such as Random Forest, Support Vector Machines (SVM), and Neural Networks can be integrated. These algorithms analyze historical data to identify patterns and predict potential threats. Research indicates that ML-based intrusion detection systems significantly improve detection performance compared to traditional methods .

Additionally, advanced techniques such as Graph Neural Networks and Explainable AI are emerging as powerful tools for cyber security. These methods enable real-time detection and provide interpretable insights for decision-making. The system can also incorporate threshold-based algorithms for network monitoring, where predefined limits trigger alerts. For example, high latency or packet loss beyond a threshold can generate warnings. Overall, the algorithmic design ensures flexibility, allowing the system to evolve from basic rule-based detection to advanced AI-driven cyber security analytics.

VII. SYSTEM DESIGN

The system design follows a modular and layered architecture to ensure scalability, flexibility, and maintainability. It is based on the Model-View-Template (MVT) architecture of Django.

1. Architecture Overview

The system consists of three main layers:

- **Presentation Layer (Frontend)**: Displays data through dashboards and user interfaces.
- **Application Layer (Backend)**: Handles business logic and processing.
- **Data Layer (Database)**: Stores security events, network metrics, and business process data.

2. Data Flow

Data enters the system through simulated inputs or real-time integrations. It is processed by the backend logic and stored in the database. The frontend retrieves this data and presents it in an organized format.

3. Module Design

- **Dashboard Module**: Provides an overview of system status, recent threats, and metrics.
- **Threat Module**: Manages security events and allows filtering and resolution.
- **Network Module**: Monitors traffic and performance metrics.
- **Business Process Module**: Tracks operational status and security scores.

4. Database Design

The database uses relational models:

- Security Event: Stores threat details.
- Network Metric: Stores performance metrics.
- Business Process: Stores service status.

Relationships between these entities are minimal, ensuring simplicity and fast query performance.

5. Security Design

The system leverages Django's built-in security features such as authentication, CSRF protection, and secure ORM queries. Future enhancements may include role-based access control and encryption.

6. Scalability and Extensibility

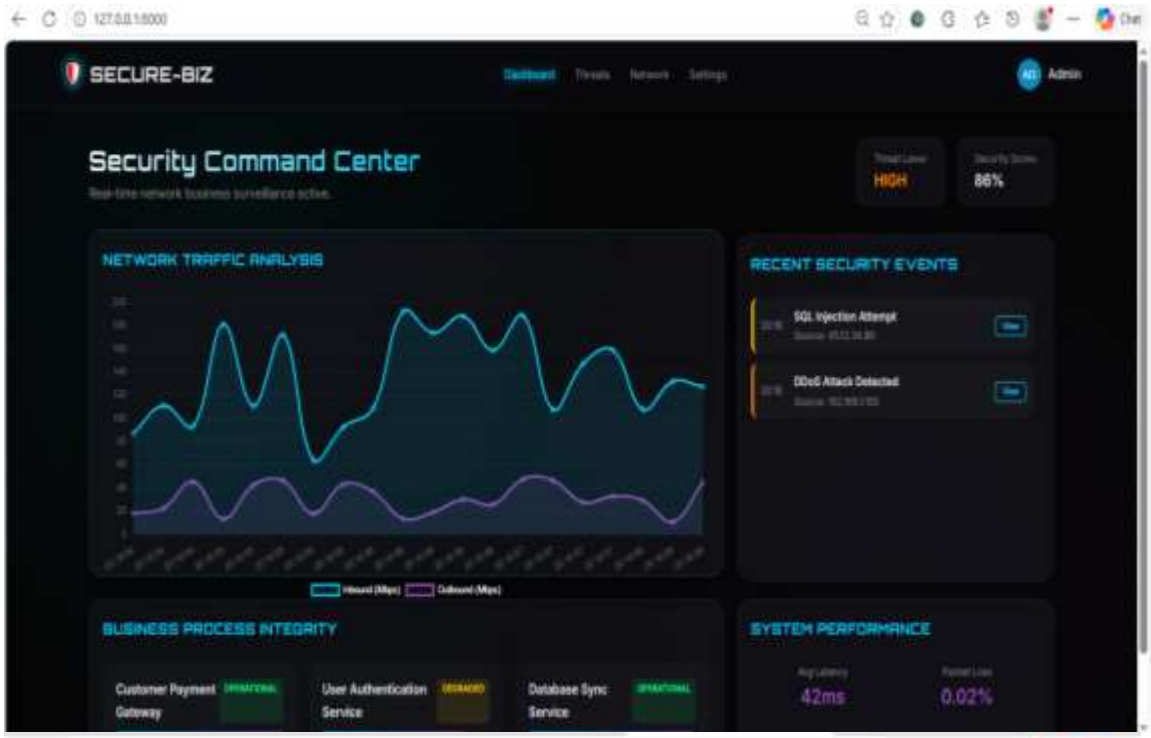
The system is designed to scale horizontally by integrating APIs, external monitoring tools, and machine learning services. Research shows that modern intrusion detection systems require scalable architectures to handle large-scale data processing .

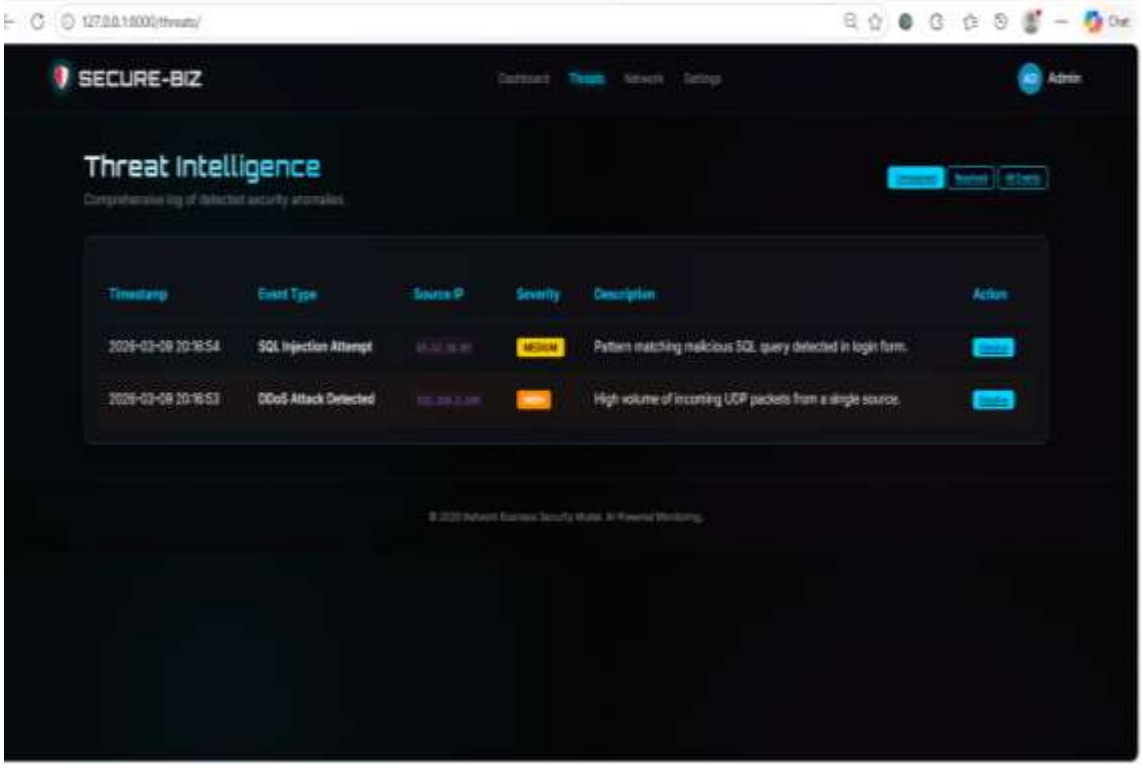
7. Visualization Design

The dashboard uses charts, tables, and indicators to present data effectively. This improves decision-making and situational awareness.

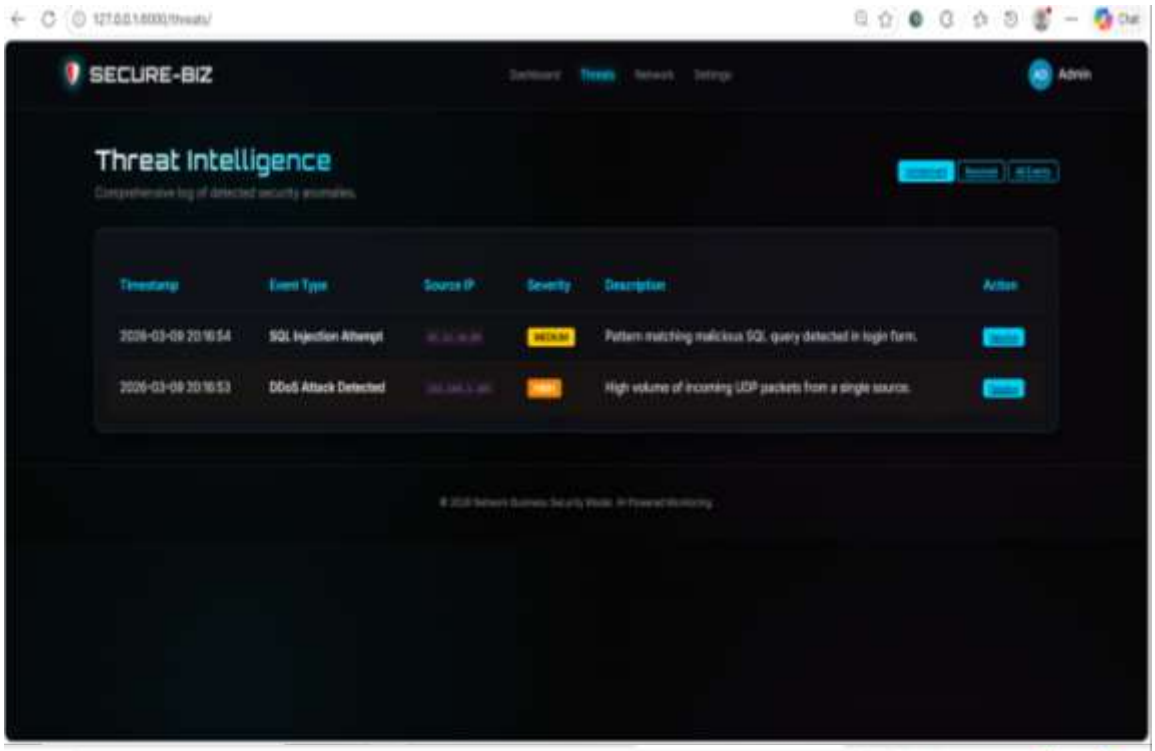
Overall, the system design ensures a balance between performance, usability, and scalability, making it suitable for real-world deployment.

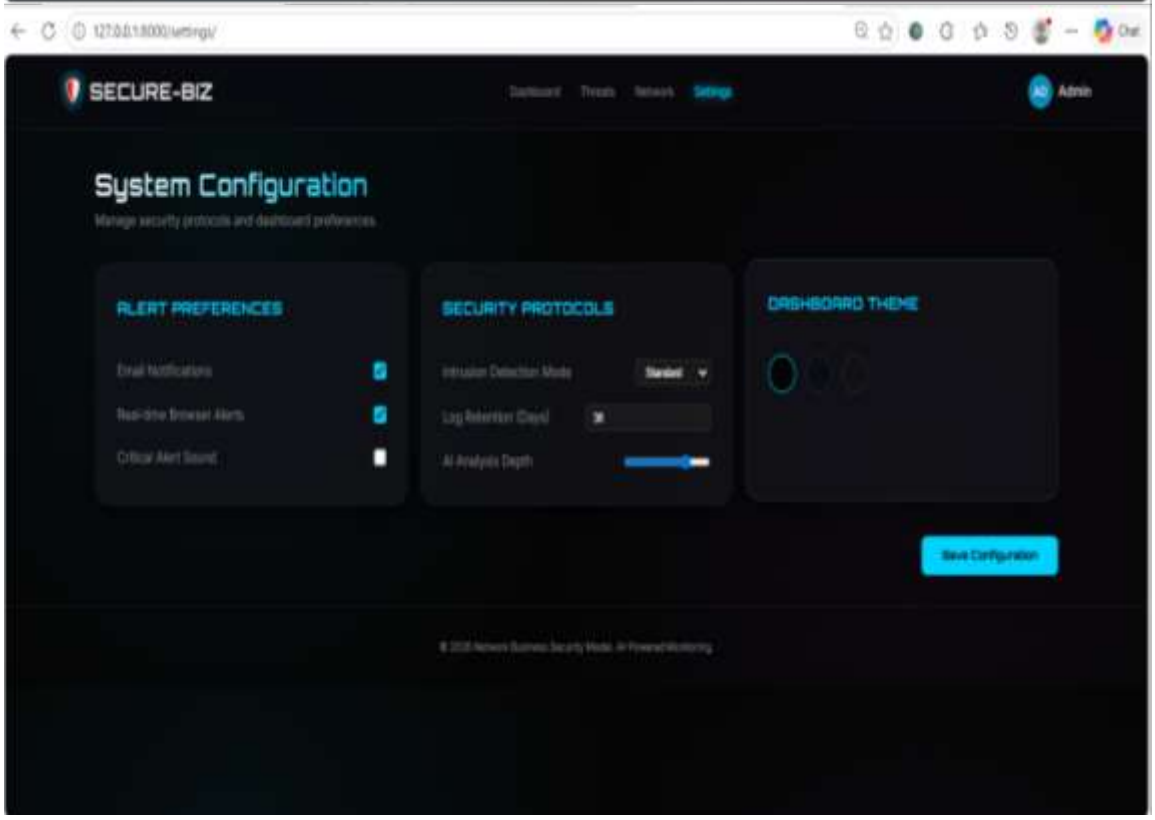
SYSTEM DESIGN IMAGES





International Journal of Engineering Research and Science & Technology





VIII. CONCLUSION

The Cyber security Threat Monitoring and Network Intelligence Dashboard presented in this project provides a comprehensive solution for managing and analyzing security events in real time. By integrating multiple components such as threat detection, network monitoring, and business process evaluation, the system offers a unified platform for cyber security management. One of the key strengths of the system is its simplicity and flexibility. Unlike traditional enterprise-level solutions, which are often complex and expensive, this system provides a lightweight and cost-effective alternative. The use of Django ensures scalability, security, and ease of development. The system enhances situational awareness by presenting critical information through an intuitive dashboard. Administrators can quickly identify threats, assess network performance, and take necessary actions.

The ability to categorize and resolve events improves incident management efficiency. Furthermore, the system is designed with future advancements in mind. The integration of machine learning and artificial intelligence can significantly enhance threat detection capabilities. Research indicates that AI-driven intrusion detection systems are becoming essential for modern cyber security due to their ability to handle complex and evolving threats. In conclusion, the proposed system demonstrates how web technologies and data analytics can be combined to build an effective cyber security solution. It addresses the limitations of existing systems and provides a foundation for future enhancements.

REFERENCES

1. Hozouri, A., et al. (2025). *A comprehensive survey on intrusion detection systems with AI*. Springer.
2. Pinto, A., et al. (2023). *ML-based IDS for critical infrastructure*. Sensors Journal.
3. Pawlicki, M., et al. (2024). *Explainable AI in intrusion detection*. Artificial Intelligence Review.
4. Sivagaminathan, V., et al. (2023). *IDS for wireless networks*. Cyber security Journal.
5. Janabi, A.H., et al. (2024). *IDS in Software Defined Networks*. IEEE Access.
6. Comparative study of IDS in healthcare (2024). ScienceDirect.
7. Shakir, V., et al. (2024). *Classification-based IDS analysis*. JASTT.

8. Panchal, R., et al. (2024). *Network-based IDS survey*. IEEE Conference.
9. Ngueajio, M., et al. (2022). *SVM-based IDS survey*. arXiv.
10. Hernandez-Ramos, J., et al. (2023). *Federated learning IDS*. arXiv.
11. Farrukh, Y.A., et al. (2024). *GNN-based IDS (XG-NID)*. arXiv.
12. Ali, T., et al. (2023). *HuntGPT intrusion detection*. arXiv.
13. Kim, G., et al. (2023). *Deep learning for cyber security*. IEEE.
14. Vinayakumar, R., et al. (2023). *Deep IDS frameworks*. IEEE Access.
15. Alazzam, H., et al. (2024). *Hybrid IDS using ML techniques*. Elsevier.