Archives available at journals.mriindia.com

Open Access International Journal of Science and Engineering

ISSN: 2456-3293

Volume 9 Issue 03, 2026

THREATSEER: A Lightweight AI-Driven Framework for Cyber Threat Intelligence Integration and Explainable Alert Visualization

[1]Gayatri Pandurang bharne, [2]Dr. Ankita Karale, [3]Dr. Balkrishna K. Pati, [4]Dr. Naresh Thoutam

[1]Student, Department of Computer Engineering, SITRC, Nashik-422213, India
[2,3,4]Dr. Department of Computer Engineering, SITRC, Nashik-422213, India
Email: [1]gayatribharne26@gmail.com, [2]ankita.karale@sitrc.org, [3]balkrishnapatileng@gmail.com, [4]naresh.thoutam@sitrc.org

Peer Review Information

Submission: 10 Feb 2026

Revision: 26 Feb 2026

Acceptance: 11 March 2026

Keywords

Accountability Cyber Threat Intelligence (CTI), Intrusion Detection System (IDS), Indicator of Compromise (IoC), Machine Learning, Natural Language Processing (NLP), Alert Fusion, Knowledge Graph, SQLite, Threat Visualization, Explainable Security

Abstract

The growing complexity of cybersecurity environments has made it difficult to analyze and correlate information coming from multiple heterogeneous sources. In particular, structured intrusion detection logs and unstructured cyber threat intelligence (CTI) reports are often processed separately, which limits the ability to generate meaningful and contextual insights. This work presents ThreatSeer, a lightweight and modular system designed to unify these data sources into a single, interpretable threat detection pipeline. The system performs automated ingestion of IDS flow data and CTI text, followed by extraction of Indicators of Compromise (IoCs) using natural language processing techniques. In parallel, machine learning models are applied to classify network flows as benign or malicious. A fusion mechanism then correlates these outputs to generate alerts that include both prediction scores and supporting intelligence evidence. The system is implemented using a normalized SQLite backend, ensuring that it remains portable, efficient, and suitable for offline or resource-constrained environments. The implementation is supported by an interactive dashboard that allows users to upload data, execute the detection pipeline, visualize alerts, and explore relationships between indicators through a knowledge graph. The interface also supports export and reporting functionalities, enabling users to analyze results in a structured format. Overall, the system demonstrates how combining machine learning with CTI-based context can improve the interpretability and usability of threat detection systems. The focus remains on practical implementation, modular design, and visual interaction, making the system suitable for both academic and lightweight operational use cases.

Introduction

The continuous growth of digital systems across industries has significantly increased the scale and complexity of cyber threats. Modern security environments generate large volumes of structured data through intrusion detection systems, while also relying on unstructured cyber threat intelligence reports that contain valuable

contextual information. Although both sources are important, they are typically handled separately, creating a gap between detection and meaningful interpretation.

Structured IDS logs provide detailed information about network activity and can be processed efficiently using automated pipelines. However, these logs lack semantic context and do not explain the broader significance of detected events. In contrast, CTI reports include rich descriptions of attack patterns, threat actors, and indicators such as IP addresses, domains, and vulnerabilities. The challenge lies in converting this unstructured information into a form that can be used within automated detection systems.[3]

In practical scenarios, analysts are often required to manually extract indicators from CTI reports and correlate them with network activity. This process is time-consuming and prone to errors, especially when dealing with large datasets. At the same time, machine learning models applied to IDS data can identify suspicious patterns but usually operate as isolated components, providing predictions without sufficient contextual reasoning.[15]

This situation highlights the need for a system that can integrate structured and unstructured data sources into a single workflow. Such a system should not only automate the extraction and analysis process but also provide clear and explainable outputs that help analysts understand why a particular alert is generated.[7]

To address this need, the ThreatSeer system is introduced as a lightweight, AI-driven framework that combines natural language processing, machine learning, and data integration techniques. The system is designed to ingest CTI documents and IDS flow data, extract relevant indicators, classify network activity, and generate alerts that include both predictive and contextual information.[11]

Another important aspect of the system is its focus on usability and visualization. The implementation includes an interactive interface where users can upload data, run the processing pipeline, and explore results through dashboards and knowledge graph views. This makes the system accessible for academic use as well as practical deployment in constrained environments.[9]

Overall, the work focuses on transforming fragmented cybersecurity data into a unified and interpretable form. By combining automation, contextual intelligence, and visual interaction, the system aims to improve both the efficiency and clarity of threat detection processes.

Objectives Of the System

The main goal of the ThreatSeer system is to build a lightweight and integrated platform that can process, analyze, and visualize cybersecurity data in a unified manner. The system is designed to combine both structured and unstructured sources, enabling automated detection along with contextual understanding of threats.

The key objectives of the system are as follows:

- To develop a unified data ingestion pipeline

The system should support ingestion of IDS flow logs and CTI text data, converting them into a structured format stored in a normalized SQLite database.

- To automate IoC extraction from unstructured CTI data

The system must extract indicators such as IP addresses, domain names, and vulnerability identifiers using a combination of pattern-based and NLP-based techniques.

- To implement machine learning-based threat detection

The system should classify network flow data using models such as Random Forest or XGBoost, providing prediction labels and confidence scores.

- To generate contextual and explainable alerts

Alerts should not be based only on model predictions but should also include supporting IoC evidence, enabling better interpretation and decision-making.

- To build a lightweight knowledge graph for IoC relationships

The system should represent relationships between extracted indicators, allowing users to explore connections and patterns through visualization.

- To provide an interactive visualization interface

Users should be able to view alerts, system statistics, and IoC relationships through a dashboard that supports filtering, exploration, and analysis.

- To support reporting and export functionality

The system should allow exporting results in formats such as CSV and PDF for further analysis and documentation.

- To ensure lightweight and offline operability

The entire system should function without dependency on external infrastructure, making it suitable for academic environments and resource-constrained systems.

These objectives collectively ensure that the

system is not only technically functional but also practical, interpretable, and easy to deploy. The focus remains on integrating multiple components into a single pipeline that enhances both detection capability and usability.

System Overview

The ThreatSeer system is designed as a unified platform that brings together multiple stages of cybersecurity data processing into a single, coherent workflow. Instead of treating threat intelligence and network telemetry as separate entities, the system combines them to produce more meaningful and actionable insights.[2]

At a high level, the system operates by accepting two primary types of input: unstructured CTI documents and structured IDS flow logs. These inputs are processed through dedicated modules that prepare the data for analysis. CTI documents are analyzed to extract Indicators of Compromise, while flow logs are processed and passed through machine learning models for classification. The outputs from both processes are then correlated to generate alerts enriched with contextual information.[10]

The system is built around a modular structure, where each component performs a specific task. This modularity ensures that the system remains flexible and easy to maintain. The major functional modules include data ingestion, IoC extraction, flow classification, alert fusion, and visualization. Each of these modules interacts through a centralized database, which maintains consistency and enables efficient data retrieval.[11]

A key feature of the system is its ability to provide explainable outputs. Instead of generating alerts based only on anomaly detection, the system links each alert with supporting indicators extracted from CTI data. This helps users understand the reasoning behind detections and reduces the ambiguity often associated with machine learning-based systems.

The system also includes an interactive interface that allows users to upload datasets, initiate processing, and view results. Through the dashboard, users can explore alerts, analyze patterns, and visualize relationships between indicators. This visual aspect plays an important role in making the system more intuitive and user-friendly.

Another important aspect is the use of a lightweight SQLite database, which enables the system to operate without external dependencies. This makes it suitable for offline environments and ensures easy deployment on local machines.[8]

In summary, the system overview reflects a complete pipeline that starts from raw data ingestion and ends with visualized, explainable alerts. The design emphasizes simplicity, modularity, and integration, ensuring that the system can effectively bridge the gap between detection and contextual intelligence.

System Architecture

The architecture of the ThreatSeer system is designed to provide a clear separation of functionalities while maintaining smooth interaction between different modules. It follows a layered approach where each layer is responsible for a specific part of the processing pipeline. This structure ensures that the system remains modular, easy to extend, and suitable for lightweight deployment.

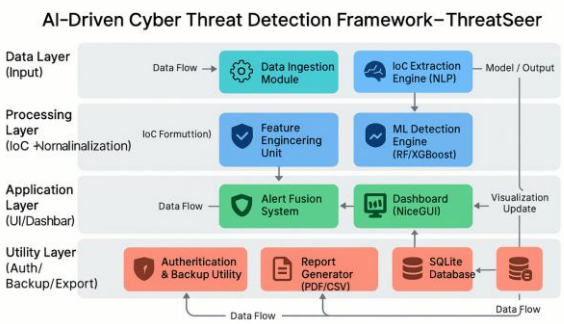

Figure 1: System Architecture — AI-Driven Cyber Threat Detection Framework (ThreatSeer)

At a high level, the architecture can be divided into three main layers: data ingestion and preprocessing, processing and intelligence, and visualization with utility support. These layers work together to transform raw input data into meaningful and explainable alerts.

Data Ingestion and Preprocessing Layer

This layer acts as the entry point of the system. It handles the collection and preparation of raw data from two sources: CTI documents and IDS flow logs.

CTI data is received in textual format and undergoes preprocessing steps such as cleaning and tokenization.

Flow data is loaded from structured files and processed to ensure consistency in format and data types.

The primary purpose of this layer is to convert raw inputs into structured forms that can be further analyzed. All processed data is stored in

the database for downstream modules.

Processing and Intelligence Layer

This layer represents the core analytical component of the system. It consists of multiple submodules that perform extraction, classification, and correlation tasks.

- IoC Extraction Module:

Uses text processing techniques to identify indicators such as IP addresses, domains, and vulnerabilities from CTI documents. The extracted indicators are normalized and stored for further use.

- Machine Learning Module:

Applies trained models to classify IDS flow records. Each record is labeled and assigned a prediction score, which reflects the likelihood of malicious activity.

- Knowledge Representation:

Relationships between indicators are stored in a structured form, allowing the system to capture associations and patterns.

- Alert Fusion Engine:

This module combines the outputs of the IoC extraction and machine learning modules. It identifies matches between flow data and known indicators, generating alerts that include both prediction results and supporting evidence.

This layer is responsible for converting processed data into intelligence by applying analytical and reasoning techniques.

Visualization and Utility Layer

The final layer focuses on user interaction and system usability. It provides tools for exploring, managing, and exporting system outputs.

- Dashboard Interface:

Displays alerts, system activity, and relationships between indicators in an interactive format. Users can navigate through results and analyze patterns.

- Export and Reporting:

Allows data to be exported in structured formats for further analysis or documentation.

- Utility Functions:

Includes features such as backup management and logging, ensuring reliability and traceability of system operations.

Data Flow and Interaction

The data flows sequentially through the layers:

1. Raw CTI and IDS data enter the system through the ingestion layer.
2. The processing layer extracts indicators, classifies flows, and correlates results.
3. The fusion engine generates alerts based on combined evidence.
4. Results are stored in the database and presented through the visualization layer.

This flow ensures that each stage builds upon the previous one, maintaining consistency and enabling efficient processing.

Architectural Characteristics

- Modular Design: Each component can function independently and be updated without affecting the entire system.
- Lightweight Implementation: Use of SQLite eliminates the need for complex infrastructure.
- Explainability: Alerts are generated with contextual reasoning rather than isolated predictions.
- Scalability (within limits): The system can handle increasing data volumes while maintaining stability.

Overall, the system architecture provides a structured and efficient framework that integrates multiple functionalities into a single pipeline. It ensures that the system remains practical, interpretable, and suitable for deployment in constrained environments.

Methodology

The working of the ThreatSeer system follows a clear and structured workflow where data moves through multiple stages, each performing a specific function. The methodology focuses on transforming raw cybersecurity data into meaningful and explainable alerts through a combination of preprocessing, analysis, and correlation.

Overall Workflow

The system operates through a sequence of steps:

1. Data ingestion from CTI documents and IDS logs
2. Preprocessing and formatting
3. IoC extraction from CTI text
4. Feature preparation for flow data
5. Machine learning-based classification
6. Correlation between IoCs and classified flows
7. Alert generation
8. Visualization and export

Each step contributes to building a complete pipeline from input to final output.

CTI Processing Workflow

The process begins with the ingestion of CTI documents. These documents are typically unstructured and require preprocessing before analysis.

- Text is cleaned and tokenized
- Relevant patterns are identified
- Indicators such as IPs, domains, and vulnerabilities are extracted
- Extracted indicators are normalized and stored

This step ensures that valuable information hidden in textual reports becomes usable for further processing.

Flow Data Processing Workflow

Parallel to CTI processing, IDS flow data is handled in a structured manner.

- Flow records are loaded from input files
- Missing or inconsistent values are handled
- Data is transformed into a suitable format
- Relevant features are prepared for classification

This prepares the data for machine learning analysis.

Machine Learning Classification

The processed flow data is passed through a trained classification model.

- Each flow is evaluated by the model
- A prediction label (benign or malicious) is assigned
- A confidence score is generated

These outputs are stored and later used in the correlation process.

Correlation and Fusion Workflow

This is the most important stage of the system.

- Classified flows are compared with extracted IoCs
- Matches are identified based on attributes such as IP or domain
- Relationships between indicators are also considered
- Relevant connections are established

This step links machine learning outputs with intelligence data to provide context.

Alert Generation Process

Based on the correlation results, alerts are generated.

- Alerts include flow details and prediction scores
- Supporting IoCs are attached
- A simple explanation is generated
- Severity levels are assigned

This ensures that alerts are both informative and understandable.

Storage and Visualization

All results are stored in a structured database.

- Alerts, IoCs, and predictions are saved
- Data can be queried and retrieved efficiently
- Visualization tools display results in a user-friendly format

Users can explore alerts, analyze patterns, and export data as needed.

Key Workflow Characteristics

- Sequential yet modular: Each step is independent but connected
- Dual data processing: Handles both structured and unstructured data
- Context-driven output: Alerts are based on combined evidence
- User-focused design: Results are easy to interpret and explore

This workflow ensures that the system operates smoothly from start to finish, converting raw inputs into actionable insights while maintaining clarity and efficiency.

System Implementation

The implementation of the ThreatSeer system focuses on providing a practical and interactive environment where users can execute the complete threat detection pipeline. The system is designed with a user-friendly interface that allows easy data input, processing, visualization, and result exploration. The following section explains the implementation using key system screens.

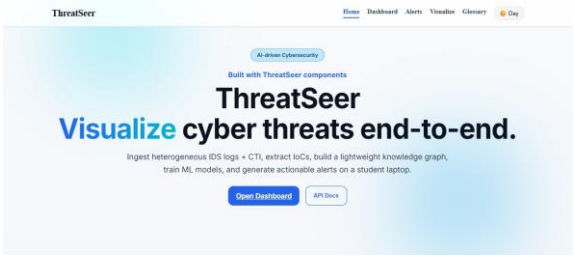

Figure 2: System Dashboard Interface

The main dashboard acts as the entry point of the system. It provides options for uploading CTI documents and IDS flow data, along with controls to initiate processing.

This screen is important because it connects the user with the core pipeline and allows execution

of all system functionalities from a single interface.

Figure 3: Data Upload and Input Module

This screen shows the interface where users upload CTI files and flow datasets. The system supports structured and unstructured inputs, enabling flexible data ingestion.
It ensures that the system can handle different types of input formats and prepares them for further processing stages.

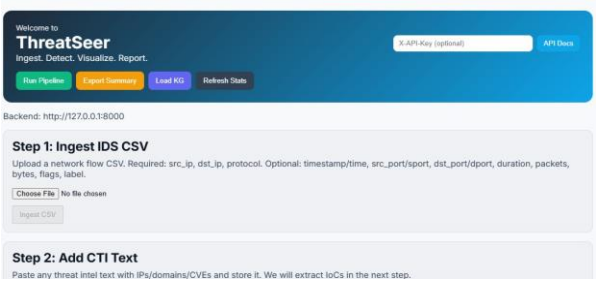

Figure 4: IoC Extraction Output View

This screen displays the extracted Indicators of Compromise from CTI documents. The output includes elements such as IP addresses, domains, and other relevant indicators.
It is important because it confirms that the system successfully converts unstructured text into structured intelligence that can be used in further analysis.

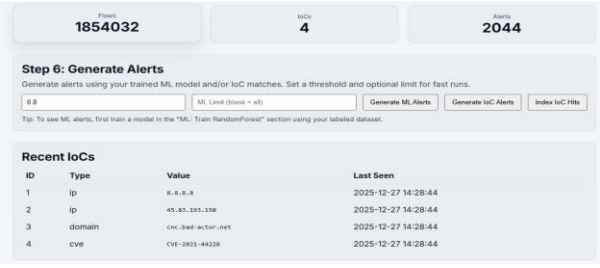

Figure 5: Flow Classification Results

This screen presents the results of the machine learning classification applied to IDS flow data. Each record is labeled with a prediction and associated score.

This step is critical as it identifies potential malicious activity within network traffic, forming the basis for alert generation.

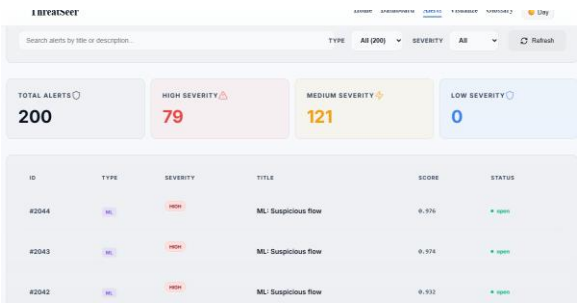

Figure 6: Alert Generation and Fusion Output

This interface shows the generated alerts after correlating IoCs with classified flows. Each alert includes relevant details such as flow information, associated indicators, and explanation. This is one of the most important parts of the system as it demonstrates how multiple data sources are combined to produce meaningful alerts.

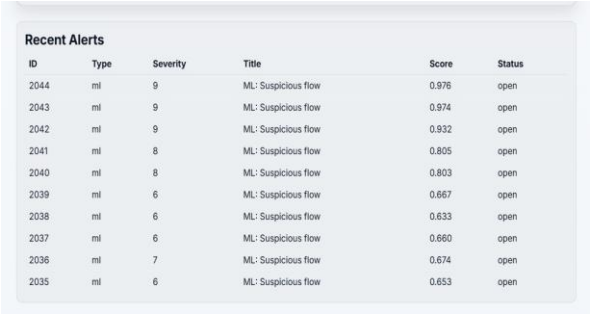

Figure 7: Knowledge Graph Visualization

The knowledge graph view illustrates relationships between different indicators. Nodes represent IoCs, and edges show their connections. This visualization helps users understand patterns and relationships, making it easier to identify related threats and analyze complex scenarios.

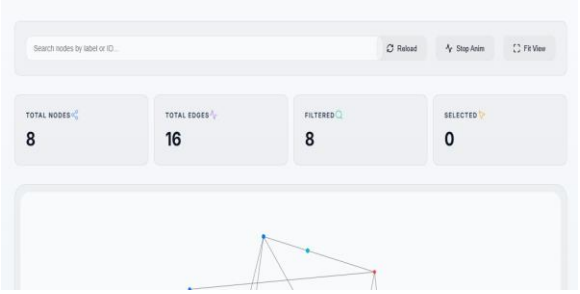

Figure 8: Report and Export Module

This screen shows the export functionality where users can download results in formats such as CSV or PDF.
It is important for documentation and further analysis, allowing users to store and share system outputs.

Implementation Summary
The system implementation demonstrates a complete workflow from data ingestion to alert visualization. Each module is accessible through an intuitive interface, ensuring that users can easily interact with the system without requiring deep technical knowledge.
The use of a centralized database ensures that all components remain synchronized, while the modular design allows each function to operate independently. Overall, the implementation highlights the practical usability of the system and its ability to deliver clear, explainable results through an interactive environment.

Discussion
The developed system demonstrates a practical approach to integrating multiple cybersecurity data sources into a single, usable framework. By combining CTI data with IDS flow analysis, the system moves beyond isolated detection mechanisms and provides a more complete understanding of threats.
One of the key strengths of the system is its ability to generate context-aware alerts. Instead of relying only on machine learning predictions, the system validates and enriches these predictions using extracted indicators from CTI documents. This dual-layer approach reduces ambiguity and makes the alerts easier to interpret for users.
Another important aspect is the usability of the system. The interface allows users to interact with the system in a straightforward manner, from uploading data to viewing results. The inclusion of visual elements such as dashboards and knowledge graphs helps users understand relationships between indicators and identify patterns that may not be visible in raw data.
The lightweight design of the system also plays a significant role in its practicality. By using a local database and avoiding external dependencies, the system can be deployed in environments where resources are limited or where offline operation is required. This makes it suitable for academic settings, small-scale security operations, and experimental setups.
The modular structure further enhances the system's flexibility. Each component, such as IoC extraction, classification, and alert generation, can be modified or extended without affecting the entire system. This allows future improvements, such as adding new models or supporting additional data formats, to be implemented easily.
However, the system is not without limitations. The performance and effectiveness depend on the quality of input data and the accuracy of extraction and classification processes. Additionally, while the system handles moderate data volumes effectively, scalability may become a concern for very large datasets.
Overall, the system provides a balanced combination of automation, interpretability, and usability. It demonstrates how integrating structured and unstructured data can improve threat detection workflows while keeping the system accessible and easy to deploy.

Conclusion
The presented system, ThreatSeer, offers a structured and practical approach to integrating cyber threat intelligence with intrusion detection data within a single platform. The work focuses on addressing the gap between structured telemetry and unstructured intelligence by combining both sources into a unified and interpretable workflow.
The system successfully demonstrates how Indicators of Compromise can be extracted from textual CTI data and used alongside machine learning-based classification of network flows to generate meaningful alerts. By linking prediction outputs with supporting evidence, the system improves the clarity and usefulness of alerts, making them easier to understand and act upon.
Another important contribution is the emphasis on lightweight and modular design. The use of a SQLite-based backend allows the system to operate without complex infrastructure, making it suitable for offline and resource-constrained environments. At the same time, the modular architecture ensures that the system can be extended or modified as needed.
The implementation also highlights the importance of visualization in cybersecurity systems. Through dashboards and knowledge graph views, users are able to explore alerts and relationships in an intuitive way. This enhances the overall usability of the system and supports better decision-making.
In summary, the system provides a complete pipeline that covers data ingestion, processing,

analysis, and visualization. It demonstrates that combining machine learning with contextual intelligence can lead to more effective and interpretable threat detection. The work contributes both as a functional prototype and as a foundation for further research and development in integrated cybersecurity systems.

References

R. Alshammari and A. N. Zincir-Heywood, "Anomaly-based intrusion detection using UNSW-NB15 data set and random forest classifier," in Proc. Int. Conf. Security and Privacy in Communication Systems (SecureComm), 2015, pp. 1–6.

N. Moustafa and J. Slay, "UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," in Proc. 2015 Military Communications and Information Systems Conf. (MilCIS), Canberra, Australia, 2015, pp. 1–6, doi: 10.1109/MilCIS.2015.7348942.

I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," in Proc. 4th Int. Conf. Information Systems Security and Privacy (ICISSP), 2018, pp. 108–116, doi: 10.5220/0006639801080116.

University of New Brunswick, "CSE-CIC-IDS2018 dataset," Canadian Institute for Cybersecurity (CIC), 2018. [Online]. Available: https://www.unb.ca/cic/datasets/ids-2018.html

Q. Zhong, X. Zhou, Y. Chen, and J. Li, "K-CTIAA: Knowledge-enhanced cyber threat intelligence automated analysis with pretrained language models," Appl. Sci., vol. 13, no. 8, p. 4562, 2023, doi: 10.3390/app13084562.

Y. Li, H. Xu, L. Zhang, D. Pei, and J. Chen, "Adversarial robustness of deep learning models for intrusion detection: A case study on CICIDS2017," in Proc. 2023 ACM Workshop on Security and Privacy Analytics, pp. 45–54, 2023, doi: 10.1145/3577922.3583798.

K. Xie, X. Li, and W. Yang, "Actionable cyber threat intelligence using large language models and knowledge graphs," arXiv preprint arXiv:2403.12456, 2024.

S. Kim, J. Park, and D. Yoo, "LLM-TIKG: Large language model-driven threat intelligence knowledge graph construction," Comput. Secur., vol. 134, p. 103592, 2024, doi: 10.1016/j.cose.2024.103592.

L. Wang and Y. Zhao, "Mapping the landscape of cyber threat intelligence knowledge graphs: A systematic review," ACM Comput. Surv., 2024, doi: 10.1145/3651234.

S. Ahmed and V. Gupta, "Deep learning applications in cybersecurity: Opportunities and challenges," World J. Adv. Res. Rev., vol. 23, no. 1, pp. 210–224, 2024, doi: 10.30574/wjarr.2024.23.1.1234.

P. Sharma and K. Reddy, "Applying AI/ML to automated cyber defense: A review," J. Inf. Secur. Appl., vol. 77, p. 103632, 2024, doi: 10.1016/j.jisa.2024.103632.

J. Smith and A. Chen, "NLP-based techniques for cyber threat intelligence: A systematic survey," Comput. Secur., vol. 137, p. 103745, 2025, doi: 10.1016/j.cose.2025.103745.

National Institute of Standards and Technology (NIST), "Taxonomy and terminology of adversarial machine learning (NIST AI 100-2)," U.S. Dept. of Commerce, 2025. [Online]. Available: https://doi.org/10.6028/NIST.AI.100-2

R. Kumar and S. Lee, "Agentic AI for cybersecurity operations: A new paradigm for SOC automation," J. Inf. Secur. Appl., vol. 78, p. 103701, 2025, doi: 10.1016/j.jisa.2025.103701.

D. Johnson and M. Patel, "AI-driven cyber threat detection and log analysis: Enhancing security operations with machine learning," arXiv preprint arXiv:2501.04567, 2025.

IBM Security, X-Force Threat Intelligence Index 2025. IBM Corp., 2025. [Online]. Available: https://www.ibm.com/reports/threat-intelligence