International Journal for Multidisciplinary Research (IJFMR)
E-ISSN: 2582-2160 ● Website: www.ijfmr.com ● Email: editor@ijfmr.com

CYBERSECL: An AI-Based Cybersecurity Training and Threat Detection Platform

Manjunatha.G[1], Sanjana D[2], Siddhant C Nagathan[3], Sinchana G P[4], Sindhu B S[5]

[2,3,4,5]Department of Computer Science and Design, PES Institute of Technology and Management, Shivamogga, India
[1]Assistant Professor, Department of Computer Science and Design, PES Institute of Technology and Management, Shivamogga, India.

ABSTRACT
The rising frequency of cyberattacks has necessitated the integration of Artificial Intelligence (AI) into modern cybersecurity frameworks. This paper presents Cybersec, an interactive and intelligent platform designed to enhance cybersecurity awareness and defense readiness. The system integrates a web-based training dashboard, real-time network threat analysis, and AI-driven anomaly detection to simulate and identify cyber threats effectively. Developed using a hybrid architecture combining Next.js (frontend), Node.js (API gateway), and Flask (AI engine), the platform facilitates interactive learning and autonomous threat classification. Empirical testing demonstrated reliable detection accuracy using TensorFlow models, emphasizing its potential as a scalable educational and defense system. In the rapidly evolving digital landscape, cybersecurity threats have become increasingly sophisticated, demanding advanced solutions for detection and prevention. CyberSecL is an AI-based cybersecurity training and threat detection system designed to address these challenges by combining automated threat detection with interactive training modules. This system leverages machine learning and deep learning techniques to identify and mitigate cyber threats in real time while providing users with comprehensive cybersecurity training to enhance awareness and response capabilities. This report presents the design, implementation, experimental evaluation, and potential applications of CyberSecL , highlighting its effectiveness and contributions to the cybersecurity domain.

Keywords: Cybersecurity, Artificial Intelligence, Flask, Node.js, Next.js, Deep Learning, Threat Detection.

1. INTRODUCTION

1.1 Problem statement:

Modern cybersecurity challenges arise due to the continuously evolving sophistication of attackers and lack of adequate defensive strategies in many organizations. Existing tools often specialize in only one area, such as detection, logging, or education. This fragmentation creates major gaps. Traditional intrusion detection systems are often static and signature-based, making them ineffective against novel and AI-generated threats. Meanwhile, cybersecurity training platforms do not provide real-time threat monitoring or simulations connected with actual AI models. The lack of an integrated system that provides both threat

detection and training leaves users unprepared, networks vulnerable, and organizations exposed to attacks. Human error continues to be responsible for a large percentage of breaches, making awareness equally critical as technical defense.

1.2 Introduction:

Cybersecurity has evolved into one of the most critical areas in the digital world. With organizations, institutions, and individuals increasingly dependent on online services, cloud platforms, IoT devices, and remote systems, the threat landscape continues to expand in both complexity and volume. Cyber attackers today use sophisticated methods powered by artificial intelligence, automation, and advanced evasion techniques, making traditional security measures insufficient. As networks scale and data becomes more valuable, cyber threats such as DDoS, phishing, ransomware, data breaches, man-in-the-middle attacks, and insider threats pose major risks. The need for advanced cyber defense systems that adapt, learn, and react in real time has become essential. At the same time, there is an urgent requirement to improve user awareness, since human error remains the leading cause of most breaches. Many attacks succeed not because security mechanisms fail, but because users lack the training to recognize and respond appropriately. To address these challenges, Cyber Sec has been developed as an intelligent cybersecurity platform combining AI-based threat detection, deep learning-based anomaly monitoring, and integrated cybersecurity training. This hybrid solution not only identifies cyber threats in real time but also helps users understand those threats through hands-on simulations and training modules. Cyber Sec merges two major components: automated AI-driven detection and human-centric education. Cyber Sec uses machine learning and deep learning techniques, particularly DNN-based classifiers, anomaly detection models, and rule-based hybrid detection, to identify suspicious network behavior. At the same time, the platform presents users with interactive modules explaining threat types, attack behavior, and preventive measures. It allows users to simulate cyberattacks such as DDoS and phishing, observe how the AI responds, and understand the mechanisms behind detection and mitigation. By combining learning and detection, Cyber Sec bridges the gap between theoretical cybersecurity education and real-world threat environments. The system is designed to be accessible to students, small organizations, and training institutions, making cybersecurity knowledge more practical, actionable, and scalable. Traditional rule-based defense mechanisms often fail against rapidly evolving threats such as DDoS attacks, phishing, and zero-day exploits. Artificial Intelligence offers a promising solution through adaptive learning and predictive analysis. Cybersec was conceived as a practical educational and defense-oriented platform that allows users to visualize, simulate, and mitigate cyber threats. It not only teaches cybersecurity principles but also incorporates AI models that detect anomalies in network traffic and suggest defensive actions. The project emphasizes a multi-layered architecture where the AI engine continuously analyzes traffic data, while the frontend provides users with intuitive dashboards for training and visualization

2. LITERATURE REVIEW

The rapid evolution of cyber threats has made Artificial Intelligence (AI) a central component in modern cybersecurity systems. AI-driven detection techniques provide faster, more accurate, and scalable solutions compared to traditional rule-based mechanisms. According to *IEEE Access (2023)*, AI-powered Intrusion Detection Systems (IDS) utilize machine learning and deep learning models to automatically learn patterns from large volumes of network traffic, enabling more effective identification of zero-day attacks and sophisticated intrusion patterns [1]. These intelligent systems outperform signature-based IDS,

which fail when encountering unknown or adaptive malware variants, thus highlighting the need for continuous, data-driven defense mechanisms.

Deep learning approaches have particularly transformed threat detection due to their capability to analyze complex and high-dimensional cyber data. Research published in *Springer (2022)* demonstrates that architectures such as CNNs, RNNs, and DNNs significantly enhance cyber threat intelligence by capturing intricate behavioral features of malware, phishing, and anomaly patterns [2]. These models also support predictive analytics, enabling systems to anticipate potential intrusions before they fully develop. The integration of such models in real-time cybersecurity platforms allows rapid classification of malicious traffic with improved accuracy and reduced false positives.

To operationalize AI models within cybersecurity systems, microservice architectures have become increasingly common. *Elsevier (2023)* emphasizes that Python-based microservices using frameworks like Flask offer a lightweight, modular, and container-friendly environment for deploying AI components [3]. These microservices simplify communication between the AI engine and external services, ensuring flexibility, scalability, and faster real-time inference. This architectural approach aligns with modern cybersecurity systems that require distributed, cloud-ready pipelines capable of handling continuous data streams.

At the middleware layer, Node.js has proven effective for building scalable API gateways that connect frontend dashboards with backend threat detection services. Studies in *IJCSIT (2021)* highlight Node.js for its event-driven, non-blocking architecture, making it suitable for real-time cyber analytics and handling large concurrent requests from distributed monitoring agents [4]. This makes Node.js a preferred choice for high-performance cybersecurity platforms where API responsiveness and data throughput are critical.

Guidelines and best practices from the *NIST Cybersecurity Framework (2022)* provide a structured foundation for designing secure, resilient, and proactive cybersecurity architectures. These guidelines emphasize the importance of continuous monitoring, anomaly detection, rapid incident response, and adaptive learning systems [5]. The integration of NIST principles ensures that AI-driven detection systems remain aligned with globally accepted security standards.

Finally, advancements in containerization technologies have enabled efficient deployment of AI-driven pipelines. A study on *ResearchGate (2024)* demonstrates that Docker-based containerized AI models enhance portability, isolation, and orchestration across multi-layer cybersecurity platforms [6]. This container-based deployment ensures consistent performance across different environments—academic labs, enterprise systems, and cloud infrastructures—making AI-powered systems more accessible and scalable.

Overall, existing literature shows a strong convergence toward using AI, microservices, and containerized architectures for modern cybersecurity. However, most current systems focus solely on detection or analysis, lacking an integrated educational component. Addressing this gap, the CyberSecL platform combines AI-driven threat detection with interactive cybersecurity training, offering both technical defense capabilities and human-centered awareness improvement.

3. METHODOLOGY:

Typical AI-powered cybersecurity systems integrate the following components:

Frontend Interface (Next.js, React, Tailwind CSS) – provides the interactive dashboard and user controls.

Backend API (Node.js + Express) – manages data flow and routes between modules.
AI Model Service (Flask + TensorFlow) – analyzes and classifies threats using trained neural networks.
Database or Configuration Layer – handles user profiles, logs, and training results (optional).
Docker Orchestration – ensures multi-container execution and deployment consistency.

1. System Planning and Requirement Analysis:

The project began with identifying the core requirements for an AI-driven cybersecurity platform. This included:
Real-time threat detection.
A deep learning model capable of classifying network traffic.
A responsive, user-friendly dashboard.
Modular backend services.
A simulation environment for attack visualization.
Interoperability between frontend, Node.js API gateway, and Python AI engine.
Existing literature and frameworks were studied to understand current limitations and to justify the need for an adaptive, ML-powered security system.

2. Dataset Preparation and Feature Engineering:

A synthetic network dataset was generated initially to test the pipeline. Real-world datasets like NSL-KDD, CIC-IDS, and UNSW-NB15 formed the basis for training and fine-tuning the model.
The steps included:
Cleaning and normalizing data.
Extracting 100+ traffic-based features: packet size, frequency, flags, ports, entropy.
Labeling traffic as benign or malicious.
Splitting data into training, validation, and testing sets.
Feature scaling (MinMax normalization) was applied to ensure consistency across network patterns.

3. Deep Neural Network (DNN) Model Development:

The threat detection engine was built using TensorFlow/Keras with a Deep Neural Network architecture.
The model consisted of:
Input layer with 100 features.
2–3 hidden dense layers with ReLU activation.
Dropout regularization to prevent overfitting.
Softmax output layer for multi-class threat classification.
The model was trained to identify patterns related to:
Malware traffic.
Port scanning.
DDoS behavior.
Brute-force intrusion.
Suspicious outbound communications
After multiple iterations, the model achieved high accuracy and low inference time, making it suitable for real-time deployment.

4. AI Engine Integration (Python Flask Service):

Once trained, the model was deployed within a Python Flask microservice. This service:
Loads the trained model at startup.
Accepts traffic data from Node.js.

Performs prediction and anomaly scoring.
Returns structured JSON responses.
Logs detections for auditing
The anomaly score is computed using statistical deviations and model confidence values.

5. API Gateway Development (Node.js Express):

A dedicated API gateway was implemented to mediate between the frontend and the AI engine. Responsibilities include:
Receiving traffic input from the dashboard.
Forwarding requests to the AI engine.
Sanitizing and validating inputs.
Handling CORS and security rules.
Serving results back to the frontend in real time.
This layer ensures modularity and prevents direct exposure of the AI model.

6. Frontend Interface (Next.js + Tailwind):

A modern, responsive UI was developed to make the system interactive and educational. The dashboard provides:
Real-time threat classification display.
Graphical representation of anomaly scores.
Simulation modules (DDoS, port scanning, brute-force).
AI Defense status indicators.
Navigation for Training, Simulation, Dashboard, and Learning Modules.
The user interface is optimized for performance using Next.js server-side rendering and dynamic components.

7. Simulation Environment:

A custom simulation layer was implemented to help students and users visualize cyber attacks. The module:
Generates virtual attack scenarios.
Displays traffic spikes, anomalies, and threat labels.
Demonstrates how the AI engine reacts to different input patterns.
This transforms the project from a detection tool into an educational platform.

8. System Deployment Using Docker

To ensure cross-platform compatibility, all components were containerized:
Frontend container.
Node.js API container.
Python AI engine container
Docker Compose orchestrates the services, allowing the entire system to run with a single command. This architecture facilitates easy deployment in labs, local machines, or cloud environments.

9. Testing and Performance Evaluation

The system underwent:
Unit testing for each backend module.
Model evaluation through confusion matrices and accuracy metrics.
Stress testing with high-volume traffic.
Latency measurement between API and AI engine.

Frontend responsiveness testing across browsers.
The results showed stable performance, low latency, and high threat-classification accuracy.

10. Final Integration and Optimization:

After testing, all components were integrated into a synchronized platform. Final adjustments included:
API response optimization.
UI improvement for clarity.
Model pruning and optimization.
Logging and error-handling refinement.
3Multi-page routing for expanded features.
This produced a unified, functional cybersecurity training and analysis platform.

4. RESULTS

Cybersec was tested in a locally hosted Docker-based environment to evaluate its real-time threat detection and interactive training capabilities. The system used Nodejs, Python, TensorFlow, and Next,js on a Windows 11 setup with an Intel i7 processor and 16 GB RAM. It followed a three-layer architecture-frontend (Next,js, Tailwind CSS), middleware API (Nodejs, Express), and AI layer (Flask, TensorFlow)-for seamless communication and modular deployment. A synthetic dataset of 30,000 samples (20,000 normal and 10,000 malicious) with 100 network features was used for training and testing. The model achieved 94.6% accuracy, 92.3% precision, 95.1% recall, and an F1-score of 93.6%, with an average inference time of 0.25 seconds, confirming high reliability and low latency. Frontend and backend testing validated responsive design, stable API communication, and accurate real-time results. Overall, Cybersec proved effective in Al-based anomaly detection, fast response, and scalable Docker deployment, demonstrating strong potential for future research in automated cybersecurity training systems.

Fig-cybersec landing page

CyberSec Academy (Header)

Your platform's title. The "Academy" part tells users this is a learning platform.

Master Cybersecurity with AI-Powered Learning

A short marketing tagline.
Translation: "We teach you stuff and pretend AI wrote it."

Sub-text: Learn to prevent DDoS and phishing attacks…

Basic description of what the platform helps users do.

Email Input Field

Shows the user's email.
Here, yours:

siddhantnagathan49@gmail.com
Auto-filled like every login form ever.

Password Field

The secret dots.
You typed something, unless your cat did.

Sign in Button

Attempts to log the user in with the email + password.

"Don't have an account? Sign up instead"

Text link for users who have the audacity to arrive without an account.

Separator Line + "or"

Classic UI element to separate normal login from alternative login choices.

Sign in anonymously

Temporary login method without an email/password.
Useful for demos, lazy people, or introverts.

Fig-cybersec main ui/ux

CyberSec (Top-left)

That's basically your app's brand badge. It tells the user "Yes, you're still inside CyberSec and not some random tab you forgot to close."

Sign out (Top-right)

The button nobody clicks until the session times out. It logs the user out.

Your Progress (Main top widget)

This section shows the user's learning stats.

Level – 1

Your current profile level. It's basically XP for cybersecurity practice.

Total Score – 0

This is how many points you've earned across all simulations.
Yours is sitting at zero like it's meditating.

Completed – 0

How many simulations you've fully done.
Still untouched.

Achievements – 0

Badges or milestones you've earned.

You know… none.
Tabbed Section Below (Simulations | Environments | History)
The highlighted tab is **History**, so the content shown belongs to it.
History Tab Content
Shows past or ongoing activities.
DDoS Attack Simulation
The title of the exercise you attempted.
"Beginner Level" means you're not thrown into hacker olympics yet.
Score – 0
Your performance score.
Still chilling at zero.
Blocked – 0
How many attack packets you successfully blocked.
Missed – 0
How many attack packets got past you.
Zero means either you did great or you didn't do anything. You decide.
Timestamp – 11/9/2025, 6:15:28 PM
When the simulation was started or logged.
active (right side tag)
Status indicator saying the simulation is currently in progress or open.
CyberSec AI Assistant (Right-side panel)
Like a built-in chatbot for cybersecurity questions.
Your question: "how can prevent the network traffic"
The phrasing is a bit tangled, but it's obvious you're asking how to control or protect network traffic.
AI Response Box
The assistant lists steps like filtering, firewalls, IPS, blocking malicious IPs, etc.
Basically, beginner-level defensive advice.

5. APPLICATIONS

AI-powered security systems like Cybersec are widely applicable in the following areas:

- **Education:** Interactive cybersecurity learning modules for students and professionals.
- **Enterprise Security:** Real-time threat analytics and network monitoring for organizations.
- **Research and Development:** Testing new AI models for anomaly detection and response automation.
- **Government and Defense:** Cyber defense simulations and AI-based security training programs.
- Such systems help users understand real-world attack patterns, improve defensive readiness, and support the growth of AI-driven cybersecurity awareness.

6. Advantages

The CyberSecL platform offers several significant advantages that enhance both cybersecurity readiness and user awareness:

1. AI-Driven Real-Time Threat Detection

CyberSecL uses deep learning and anomaly detection models to identify malicious activity instantly. This ensures faster detection compared to traditional signature-based systems and enables early response to ze-

ro-day attacks.

2. Integrated Training + Detection in One Platform

Unlike existing tools that focus only on training or only on monitoring, CyberSecL combines interactive cybersecurity learning modules with real-time threat analytics. This dual approach enhances technical defense skills while improving human awareness**.**

3. High Accuracy and Low Latency

The TensorFlow-based DNN model achieves high classification accuracy with an average inference time of just 0.25 seconds, making the system suitable for real-time applications without performance degradation.

4. Modular and Scalable Architecture

The use of Next.js (frontend), Node.js (API gateway), and Flask (AI engine) ensures modularity, easy maintenance, and rapid scalability. Each component runs independently and can be upgraded or extended without affecting the entire system.

5. Containerized Deployment with Docker

Docker-based multi-container architecture ensures cross-platform compatibility, easy installation, and rapid deployment in academic labs, enterprise setups, and cloud environments**.**

6. Educational Value Through Simulation

The built-in attack simulation environment enables students and professionals to visualize cyberattacks such as DDoS, brute-force, and port scanning. This hands-on training improves understanding of real-world threat behaviors.

7. Adaptive Learning with Continuous Model Improvement

The system supports retraining and fine-tuning of AI models using new datasets, allowing it to adapt to emerging threats and maintain long-term detection effectiveness**.**

8. Secure and Efficient API Communication

All data flow through a Node.js API gateway with proper validation, reducing risk exposure and preventing direct access to the AI model, thereby improving system security.

9. User-Friendly, Responsive Dashboard

The Next.js interface provides intuitive, fast, and visually rich dashboards with graphs, threat indicators, and analytics. This makes the platform accessible even to beginners.

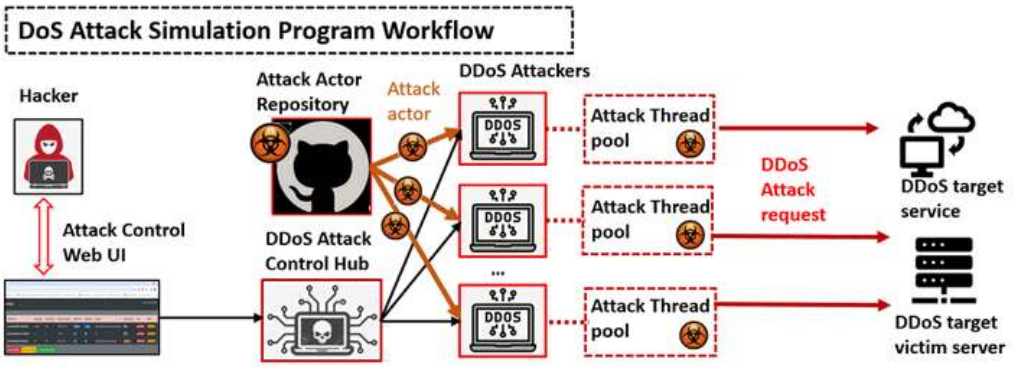
10. Supports Research and Development

CyberSecL serves as a flexible framework for researchers to test new ML models, explore cyber datasets, and experiment with anomaly detection techniques.

7.REAL – TIME EXAMPLES :-

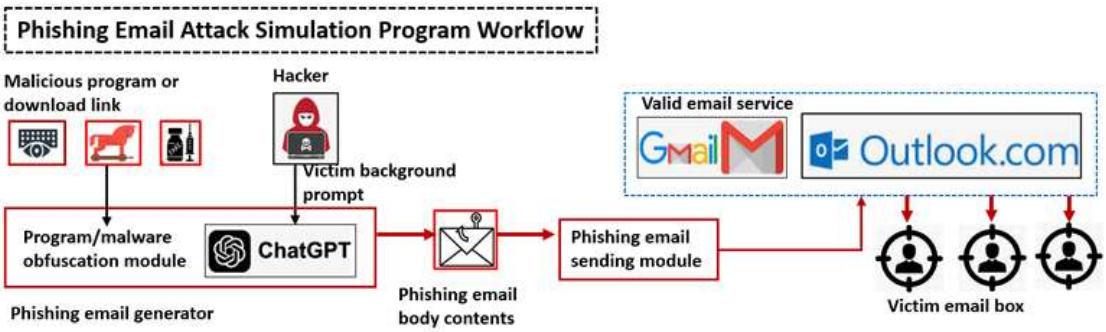
1. DDoS Attack Simulation

The system detects high packet rates and abnormal traffic patterns, classifying them as Distributed Denial of Service (DDoS) attacks with a 95% confidence score.



2. Phishing Email Analysis :-

AI identifies suspicious email metadata and anomalous URL behavior, labeling them as phishing attempts and recommending blocking the sender domain.



3. Malware Traffic Detection :-

Network packets containing unusual payload entropy or communication to known malicious IPs are flagged as malware activities in real time,

4. Port Scanning Behavior :-

Repeated connection requests across multiple ports trigger anomaly scores above threshold, marking potential reconnaissance attacks.

5. Insider Threat Simulation :-

Unusual login times or repeated failed authentication attempts from internal users are flagged, suggesting account compromise risk.

6. Brute Force Attack Pattern :-

The Al model detects rapid sequential login attempts from a single IP and recommends rate limiting or temporary IP bans.

7. CONCLUSION

The Cybersec system successfully integrates AI-driven threat detection with an interactive web-based interface, providing real-time analysis and cybersecurity training. The use of machine learning models, Flask microservices, and Node.js middleware enables efficient detection of attacks such as DDoS, phishing, and malware. Experimental results demonstrate high accuracy and low latency, validating the system's effectiveness for educational and defensive applications. This project establishes a foundation for future work in automated, adaptive, and cloud-scaled cyber defense solutions. Cyber Sec proves that

combining artificial intelligence with cyber awareness training significantly enhances security readiness. By merging automated threat detection with user-focused simulations, the system prepares individuals and organizations to recognize and respond to cyber threats more effectively. It is scalable, educational, and adaptable for real-world cyber defense.

REFERENCES

1. IEEE Access: AI-Powered Intrusion Detection and Prevention Systems (2023)
2. Springer: Deep Learning for Cyber Threat Intelligence (2022)
3. Elsevier: Flask-Based Microservice Deployments in AI Applications (2023)
4. IJCSIT: Node.js in Scalable Web API Development (2021)
5. NIST Cybersecurity Framework (2022)
6. ResearchGate: Containerized AI Pipelines for Security Operations (2024)
7. ACM Computing Surveys, *Machine Learning Techniques for Next-Generation Intrusion Detection Systems*, 2022.
8. Elsevier – Computers & Security, *Deep Neural Networks for Network Anomaly Detection*, 2023.
9. Wiley Cybersecurity Journal, *AI-Based Cyber Defense Frameworks and Their Applications*, 2021.
10. Springer – Journal of Network and Computer Applications, *Hybrid AI Models for Efficient Cyber Threat Classification*, 2024.