iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 **Issue:** III **Month of publication:** March 2025

DOI: https://doi.org/10.22214/ijraset.2025.67599

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue III Mar 2025- Available at www.ijraset.com

AI-Driven Network Anomaly Detection System

P. Hari Babu[1], CH. Venkata Pavan[2], P. Issac[3], T. K. Ganesh[4], K. Jayanth[5]
[1]*Dept of CSE, Raghu Engineering College*
[2, 3, 4, 5]*Dept of CSE, Raghu Institute of Technology*

Abstract: ***In today's digital landscape, the increasing sophistication of cyber threats necessitates more advanced security solutions. Traditional Network Intrusion Detection Systems (NIDS) often rely on signature-based methods, which can struggle to detect novel or evolving attacks. To address this limitation, this project focuses on enhancing anomaly detection by incorporating advanced AI techniques, specifically behavioral analysis. By continuously profiling normal network behavior, the system can identify deviations that may indicate potential threats. This proactive approach allows for real-time anomaly detection, improving network security by identifying threats before they cause significant damage.***

The core of this system lies in the integration of machine learning algorithms and generative models to differentiate between benign and malicious activities with high accuracy. By leveraging behavioral patterns and historical network data, the AI-driven NIDS can adapt to new attack methods, making it more effective than traditional rule-based approaches. Machine learning techniques enable the system to learn from past anomalies, refining its detection capabilities over time. Additionally, generative models can simulate attack scenarios, helping the system recognize subtle anomalies that might otherwise go unnoticed. This predictive ability strengthens network defenses by identifying potential risks before they escalate.

Beyond detection, the proposed NIDS aims to predict future network anomalies, allowing organizations to implement preventive security measures. The system's intelligent threat assessment helps cybersecurity teams respond more efficiently, minimizing false positives while ensuring that real threats are addressed promptly. As cyber threats continue to evolve, having an adaptive and self-learning security mechanism becomes crucial. This AI-powered approach enhances the overall resilience of network infrastructures, making it a valuable asset in the fight against cybercrime. Through continuous learning and refinement, the system will provide a more efficient and reliable solution to modern cybersecurity challenges.

Keywords: AI-driven Network Intrusion Detection System (NIDS), Anomaly detection, Deep learning models, Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), Long Short-Term Memory (LSTM), Feature extraction, Real-time threat detection

I. INTRODUCTION

In the modern digital landscape, network security has become a critical concern due to the increasing sophistication of cyber threats. Traditional Network Intrusion Detection Systems (NIDS) face significant limitations, primarily relying on static rule-based methods that struggle to detect new and evolving attacks. This project aims to develop an AI-driven NIDS to enhance anomaly detection and improve overall network security. By leveraging advanced machine learning techniques, the system can analyze network behavior dynamically, identifying deviations that may indicate potential cyber threats. The need for a more intelligent and adaptive security solution drives the motivation behind this project.

Existing NIDS solutions often suffer from high false-positive rates and an inability to detect zero-day attacks, making them less effective in real-world scenarios. To address these challenges, the proposed AI-driven NIDS integrates behavioral analysis, deep learning models, and generative models to create a robust detection mechanism.

Behavioral analysis helps in profiling normal network activities, while deep learning techniques enhance the system's ability to recognize complex attack patterns. Generative models further improve detection by simulating network traffic, allowing the system to learn from both normal and anomalous data. This hybrid learning approach ensures a more accurate and adaptive response to cybersecurity threats.

The objectives of this project revolve around developing a comprehensive NIDS that not only detects but also predicts network anomalies. The system aims to reduce false positives, improve detection accuracy, and adapt to new threats without requiring constant manual updates. Additionally, the project focuses on implementing real-time threat analysis and response mechanisms to enhance network security. By combining various AI techniques, the system will provide a more efficient and intelligent defense against cyber threats, making it a valuable contribution to modern cybersecurity solutions.

A. Objectives

The primary objective of this project is to develop an AI-driven Network Intrusion Detection System (NIDS) that enhances the detection and prediction of network anomalies by leveraging advanced artificial intelligence (AI) techniques. The system is designed to overcome the limitations of traditional methods and provide a more intelligent, adaptive, and reliable defense mechanism for modern networks. The key objectives of the proposed system are as follows:

1) Enhance Anomaly Detection with Behavioral Analysis

A major objective of this project is to enhance the ability of the NIDS to detect anomalies by implementing behavioral analysis. The system will profile normal network behavior through the use of machine learning models, which will help establish a dynamic baseline. This will allow the system to continuously adapt to changing network conditions, making it more capable of detecting deviations indicative of potential threats. By analyzing network traffic patterns over time, the system will recognize subtle anomalies that may otherwise be missed, leading to improved detection accuracy and fewer false positives

2) Predict Network Anomalies in Real-Time

Another critical objective is to enable the system to predict network anomalies before they result in significant damage. By leveraging predictive machine learning models, the system will not only detect existing threats but will also forecast potential future anomalies based on observed trends and patterns in the network traffic. This predictive capability will allow network administrators to take preemptive actions, reducing the risk of attack and improving overall network security.

3) Implement Deep Learning Models for Advanced Detection

The integration of deep learning models is a central objective of the project. Techniques such as Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) will be used to analyze complex network traffic patterns, enabling the system to detect sophisticated and previously unseen threats. CNNs are particularly effective at identifying spatial hierarchies in the data, while RNNs can capture temporal relationships within traffic patterns. These models will enhance the NIDS's ability to analyze large volumes of data and detect both known and unknown threats with high accuracy.

4) Address Data Imbalance with Generative Models

An important challenge in network anomaly detection is the issue of data imbalance, where normal network traffic significantly outnumbers malicious traffic. To tackle this, the system will incorporate Generative Adversarial Networks (GANs) and Variational Autoencoders (VAEs). These generative models will help address the imbalance by generating synthetic samples of malicious traffic, thereby augmenting the training data and improving the model’s ability to accurately identify rare threats. This will improve the system’s performance in detecting both known and novel attacks.

5) Hybrid Approach for Comprehensive Threat Detection

The system will utilize a hybrid approach that combines supervised and unsupervised learning techniques to enhance its detection capabilities. Supervised learning will be used to detect known attacks based on labeled data, while unsupervised learning will identify anomalous behaviors or previously unseen threats. This approach ensures that the system can detect both known and unknown threats more effectively, reducing the rate of false positives and enhancing the overall reliability of the system.

II. EXISTING SYSTEM

Traditional Network Intrusion Detection Systems (NIDS) primarily rely on two conventional detection methods: Signature-Based Detection and Statistical Anomaly Detection. Signature-based detection operates by comparing network activity against predefined threat signatures. When a match is found, the activity is flagged as a potential threat. This approach is highly effective in identifying known attacks with minimal false positives. However, its inability to detect new, unknown, or zero-day threats significantly limits its adaptability and effectiveness in modern cybersecurity landscapes. Statistical anomaly detection establishes a baseline of normal network behavior using statistical models, flagging deviations as potential threats. While this method can identify irregular patterns, it often suffers from high false positive rates, especially in dynamic network environments. Additionally, it struggles with efficiently analyzing large-scale, high-dimensional network traffic, which can result in missed detections or unreliable alerts. These limitations highlight the need for more adaptive and intelligent intrusion detection approaches.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue III Mar 2025- Available at www.ijraset.com

III. PROPOSED SYSTEM

A. Programming Languages

Python is used for developing the AI-driven Network Intrusion Detection System (NIDS) due to its rich ecosystem for machine learning and data analysis. It simplifies data preprocessing, model training, and integration with web applications, making it ideal for network security tasks.

B. Libraries and Frameworks

Flask is used for building the web API, enabling seamless interaction with administrators. Tkinter provides a user-friendly GUI, while Scikit-learn, TensorFlow, and Keras power the machine learning models. Pandas and NumPy handle data processing, and Matplotlib/Seaborn aid in visualization.

C. Machine Learning and Deep Learning Models

Random Forest classifies network anomalies, while CNNs and RNNs detect complex attack patterns. LSTM networks capture long-term dependencies in network traffic, improving threat identification and response.

D. System Integration

A Flask-based API facilitates communication between system components, while real-time data processing ensures timely threat detection. The system supports automated responses to mitigate cyber threats.

E. Development Environment

The project is developed using PyCharm and VS Code, with Git for version control. It is compatible with Linux, Windows, and macOS, supporting collaborative development and debugging.

F. Deployment Environment

The system runs on Linux-based servers, ensuring robust network monitoring. Docker is used for containerization, simplifying deployment and scalability across cloud and on-premise environments.

G. Modules Used in the Project

The system includes modules for data collection (Wireshark, Pyshark), preprocessing (Pandas, NumPy), anomaly detection (Scikit-learn, TensorFlow), classification, blocking (iptables integration), and user interface (Tkinter, Flask API).

H. Algorithms

Signature-based and anomaly-based detection techniques are combined with machine learning models like Random Forest, SVMs, CNNs, and RNNs. A hybrid learning approach enhances accuracy and adaptability against evolving cyber threats.

IV. SOFTWARE REQUIREMENTS

The AI-driven Network Intrusion Detection System (NIDS) is implemented using Python as the primary language for machine learning models, data preprocessing, and server-side application development. JavaScript may be used for front-end development if required.

For machine learning, TensorFlow and Keras facilitate deep learning models like CNNs and RNNs, while Scikit-learn is used for traditional algorithms such as Random Forest and SVM. PyTorch is included for advanced deep learning applications, and OpenCV can be utilized for visual anomaly detection when needed.

Flask is the primary web framework, enabling real-time API interactions with NIDS models. Django is an optional choice for implementing extensive web-based features such as user management and monitoring dashboards.

Data storage relies on MySQL or PostgreSQL for structured logs and user actions, while MongoDB or other NoSQL databases can be used to manage large, unstructured network traffic data. This ensures efficient logging and retrieval of security events.

Tkinter is employed to provide a graphical interface for real-time alerts and anomaly detection insights. Additionally, Plotly and Matplotlib are used for data visualization, helping in monitoring network traffic trends and system performance metrics.

The system supports multiple operating systems, with Linux (Ubuntu or CentOS) preferred for deployment due to stability and security. Windows and macOS are viable for development and testing, ensuring flexibility across different environments.

V. HARDWARE REQUIREMENTS

To effectively deploy and run the AI-driven Network Intrusion Detection System (NIDS), robust server hardware is essential. A multi-core processor, such as Intel i7/i9 or AMD Ryzen, is recommended to handle the computational demands of deep learning model training and real-time network traffic analysis. The system should have a minimum of 16 GB RAM, with 32 GB preferred, to ensure smooth processing of large datasets and machine learning operations. Storage requirements include at least a 500 GB SSD to accommodate network traffic logs, model parameters, and training datasets.

A dedicated GPU, such as NVIDIA GTX 1660 or RTX 3060, can significantly accelerate deep learning tasks, particularly for processing-intensive models like Convolutional Neural Networks (CNNs). This is crucial for reducing training time and enabling real-time anomaly detection. Additionally, the system should have a high-speed Network Interface Card (NIC) capable of handling at least 1 Gbps or 10 Gbps data speeds to support real-time network traffic monitoring and processing.

To maintain system reliability and prevent data loss, backup systems and redundant network connections are necessary. These measures ensure high availability and minimize downtime in case of hardware failure or maintenance activities. Having a robust backup strategy also safeguards critical logs and model data, enhancing the overall security and efficiency of the system.

VI. RESULTS

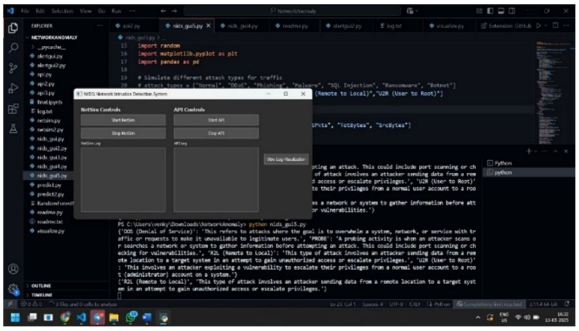


Fig 1: Step – 1

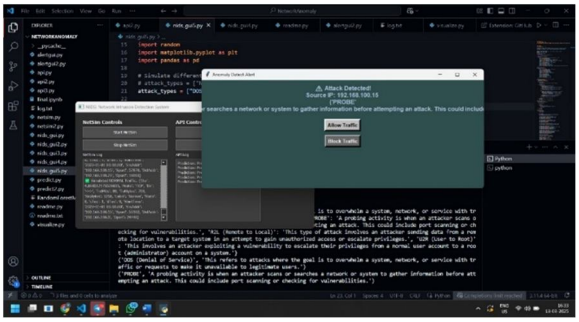


Fig 2: Step – 2

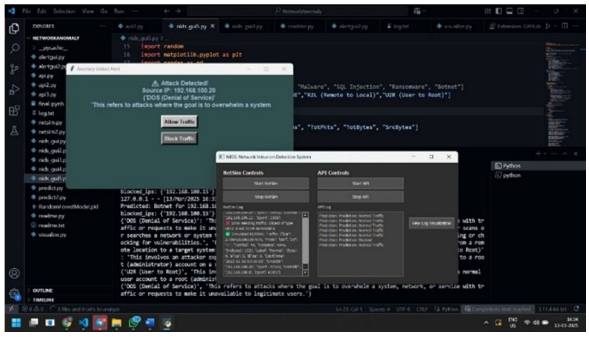


Fig 3: Step – 3

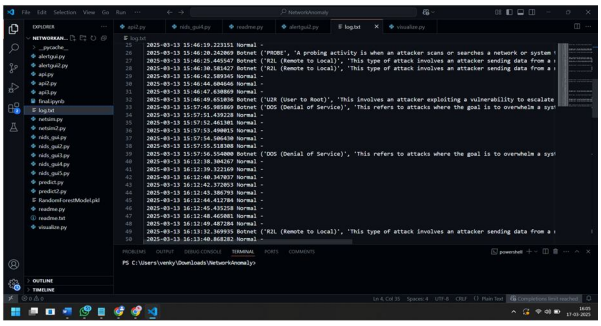


Fig 4: Logs Of Attacks

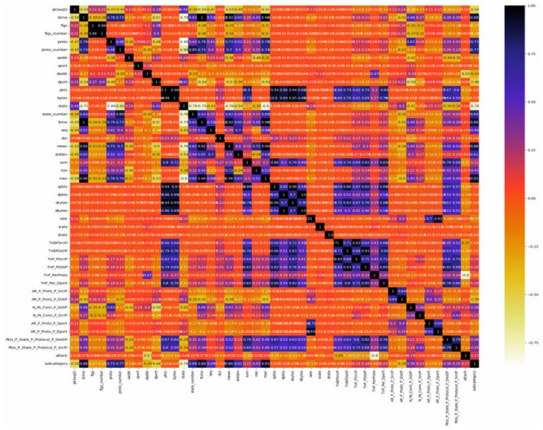


Fig 5: Data analysis and visualization

VII. FUTURE SCOPE

The AI-driven NIDS has the potential for significant future advancements, ensuring better adaptability and effectiveness in combating evolving cyber threats. One key area of growth is the integration of reinforcement learning to enhance decision-making, allowing the system to automatically refine its anomaly detection models based on real-time feedback. This would improve accuracy while reducing false positives, making it more reliable for large-scale deployment.

Another promising direction is cloud-based and edge computing integration, enabling real-time intrusion detection with minimal latency. By deploying the NIDS on cloud platforms, organizations can scale security monitoring dynamically, while edge computing would allow for faster local threat detection, reducing dependency on centralized data centers. Additionally, the use of federated learning can enhance privacy and security by training models across distributed networks without sharing sensitive data.

The system could also benefit from blockchain-based security enhancements, where a decentralized ledger records intrusion attempts, making attack data tamper-proof and more transparent for cybersecurity teams. Moreover, collaborative threat intelligence sharing between organizations can help improve detection models by providing a broader dataset of known attack patterns.

In the long run, this AI-powered NIDS could evolve into a fully autonomous security system capable of self-healing and auto-mitigation of threats. By integrating with AI-driven response mechanisms, the system could take corrective actions in real-time, such as blocking malicious traffic or isolating compromised devices. These advancements will ensure that cybersecurity remains proactive, adaptive, and resilient against emerging cyber threats.

VIII. CONCLUSION

The AI-driven Network Intrusion Detection System (NIDS) enhances cybersecurity by detecting network anomalies in real time, significantly improving an organization's resilience against cyber threats. By continuously monitoring network traffic, the system can identify unusual patterns that may indicate potential security breaches. This proactive approach minimizes the risk of cyberattacks going undetected, ensuring that threats are addressed before they can cause significant harm. With advanced anomaly detection, organizations can maintain a secure digital environment while preventing unauthorized access and data breaches.

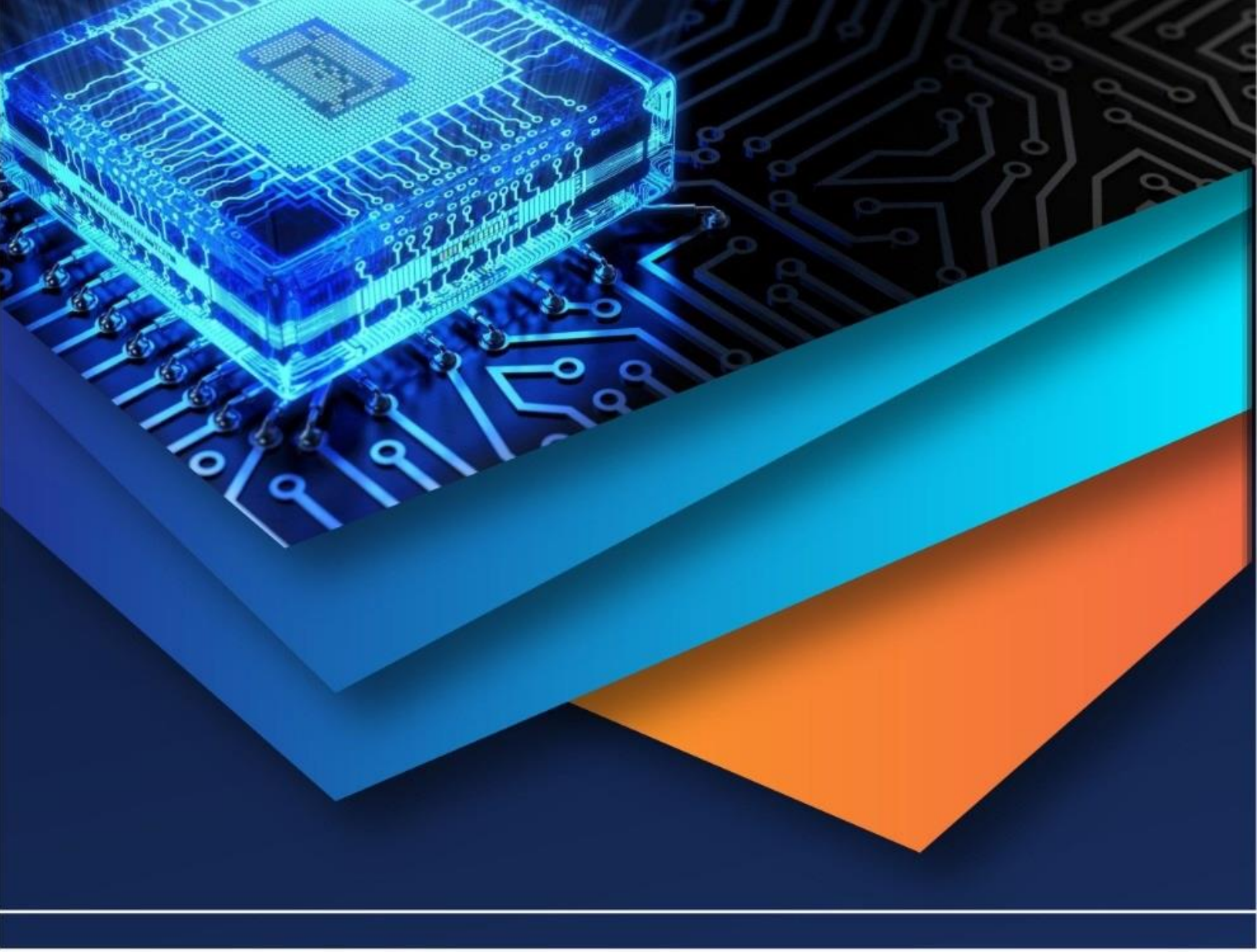
Leveraging machine learning, the system continuously learns from historical network behavior and adapts to emerging threats. Unlike traditional signature-based methods that rely on predefined attack patterns, this AI-powered solution can detect zero-day attacks and novel cyber threats. By analyzing network traffic in real-time, the system can recognize deviations from normal behavior, triggering alerts for potential security incidents. This early threat detection capability significantly reduces response time, allowing cybersecurity teams to take preventive measures before an attack escalates.

A key advantage of this AI-driven NIDS is its adaptability to different network structures, making it suitable for diverse environments, including enterprise networks, cloud infrastructures, and IoT ecosystems. Traditional methods often struggle to scale across various network configurations, but the AI-based approach ensures flexible deployment. Whether implemented in a corporate setting or a cloud-based platform, the system adjusts dynamically to the network's unique traffic patterns, ensuring robust security across different architectures.

Beyond detection, the system provides real-time alerts and generates detailed reports, allowing security teams to analyze incidents efficiently. The comprehensive logging of anomalies helps in forensic investigations, enabling organizations to understand attack patterns and refine their security strategies. By integrating with existing security infrastructure, such as SIEM (Security Information and Event Management) solutions, the system enhances overall threat visibility and incident response capabilities.

REFERENCES

[1] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," ACM Computing Surveys (CSUR), vol. 41, no. 3, pp. 1-58, 2009.
[2] This paper provides an extensive survey of anomaly detection techniques, including statistical and machine learning-based methods used in the detection of network anomalies.
[3] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: A review," ACM Computing Surveys (CSUR), vol. 31, no. 3, pp. 264-323, 1999.
[4] This reference discusses the concept of data clustering and its relevance to detecting anomalies in large datasets such as network traffic.
[5] K. H. Lee, S. P. Cho, and D. H. Kim, "A deep learning approach to network intrusion detection," in Proc. of the IEEE International Conference on Network Protocols, 2018, pp. 234-243.
[6] This paper discusses the application of deep learning models, including Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), for network intrusion detection.
[7] Y. Goodfellow, I. Pouget-Abadie, M. Mirza, et al., "Generative adversarial nets," in Proc. of Advances in Neural Information Processing Systems (NeurIPS), 2014, pp. 2672-2680.
[8] The foundational paper on Generative Adversarial Networks (GANs), detailing how GANs can be used for generating synthetic data to improve the training of machine learning models, especially in imbalanced datasets.
[9] L. D. Iglewicz, A. S. Das, and K. K. Raman, "Anomaly detection for the network security," in Proc. of the International Conference on Cyber Security, 2017.
[10] This paper highlights the use of anomaly detection techniques specifically for network security and compares traditional methods with machine learning-based approaches.
[11] S. Z. Li, H. L. Sun, and Y. J. Liu, "Variational autoencoders for anomaly detection in networks," Journal of Computational Intelligence and Neuroscience, vol. 2020, Article ID 723485, 2020.
[12] This study focuses on the use of Variational Autoencoders (VAEs) for detecting network anomalies, especially in systems with highly imbalanced datasets.
[13] R. S. Aljohani, S. S. R. S. A. Murthy, and L. M. Al-Samarraie, "A hybrid machine learning-based approach for anomaly detection in computer networks," in Proc. of the International Conference on Machine Learning and Applications (ICMLA), 2019, pp. 123-130.
[14] This paper explores the combination of supervised and unsupervised learning techniques in a hybrid approach for improving anomaly detection in network security systems.
[15] "Wireshark: Network Protocol Analyzer," [Online]. Available: https://www.wireshark.org/.
[16] Wireshark is one of the most commonly used tools for capturing and analyzing network traffic. It was essential in the data collection phase of this project.
[17] "Flask: Web Development, One Drop at a Time," [Online]. Available: https://flask.palletsprojects.com/.
[18] Flask is a lightweight web framework used to develop the API endpoints that interact with the NIDS for anomaly detection.
[19] "Tkinter: Python Interface to Tk GUI Toolkit," [Online]. Available: https://wiki.python.org/moin/TkInter.
[20] Tkinter is the library used to build the graphical user interface (GUI) for managing blocked IP addresses within the system.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY