International Journal of Engineering Research and Science & Technology

IJERST

ISSN : 2319-5991

www.ijerst.com

Email: editor@ijerst.com or editor.ijerst@gmail.com

International Journal of Engineering Research and Science & Technology

ISSN 2319-5991 www.ijerst.com

Vol. 21, Issue 2, 2025

NETWORK TRAFFIC ANALYSIS FOR INTRUSION DETECTION USING PYTHON

Yarramsetti Bhagya Sri Lakshmi, **21PD1A0591**, Department of CSE, West Godavari Institute Of Science and Engineering, Andhra Pradesh, India.

Nakka Gayathri Devi, 21PD1A0557, Department of CSE, West Godavari Institute Of Science and Engineering, Andhra Pradesh, India.

Madduri Harshavardhan,21PD1A0547, Department of CSE, West Godavari Institute Of Science and Engineering, Andhra Pradesh, India.

Yantrapragada Mahesh, 21PD1A0589, Department of CSE, West Godavari Institute Of Science and Engineering, Andhra Pradesh, India.

Mutyala Surya Sai Sree, 21JG1A0578, Department of CSE, West Godavari Institute Of Science and Engineering, Andhra Pradesh, India

Dr. M.Aravind Kumar, Professor, Department of ECE, West Godavari Institute Of Science and Engineering, Andhra Pradesh, India.

Dr. M.Aravind Kumar, Professor, Department of ECE, West Godavari Institute Of Science and Engineering, Andhra Pradesh, India.

ABSTRACT

Intrusion detection in network traffic is vital for cybersecurity, and traditional rule-based methods often fall short against evolving threats. This study leverages Machine Learning (ML) techniques using Python—specifically Regression, K-Nearest Neighbours (KNN), and Decision Trees (DT)—to enhance anomaly detection. Regression identifies unusual traffic variations, KNN classifies activities based on similarity, and DTs improve accuracy by learning from past attacks. By processing real-time network data and applying these models, the approach achieves higher accuracy and efficiency than conventional methods, emphasizing the crucial role of ML in modern cybersecurity defenses.

INTRODUCTION

With the rapid growth of digital networks, cyber threats and intrusions have become critical concerns, and traditional Intrusion Detection Systems (IDS), which rely on signature or rule-based methods, often fail to detect new threats. To overcome these limitations, this study employs Machine Learning techniques—Regression, K-Nearest Neighbors (KNN), and Decision Trees (DT)—to classify normal and malicious network traffic using historical data. Regression detects unusual patterns, KNN classifies activities based on similarity, and DT effectively distinguishes between benign and harmful traffic. Using Python-based ML frameworks, the proposed system enhances real-time detection and offers a scalable, adaptive, and automated solution for robust network security.

LITERATURE SURVEY

Intrusion Detection Systems (IDS) are vital for identifying and mitigating cybersecurity threats in real-time, but traditional methods face limitations against evolving attacks. Signature-based IDS, such as Snort and Suricata, rely on predefined rules to detect known threats, making them ineffective against zero-day attacks and requiring frequent updates. Anomaly-based IDS, on the other hand, detect deviations from normal network behavior using statistical or clustering techniques, offering better adaptability but often producing high false positive rates due to natural traffic variations. Studies by Mukherjee et al. (1994) and Denning (1987) underscore these limitations, highlighting the need for more intelligent approaches like

International Journal of Engineering Research and Science & Technology
ISSN 2319-5991 www.ijerst.com
Vol. 21, Issue 2, 2025

machine learning in intrusion detection.

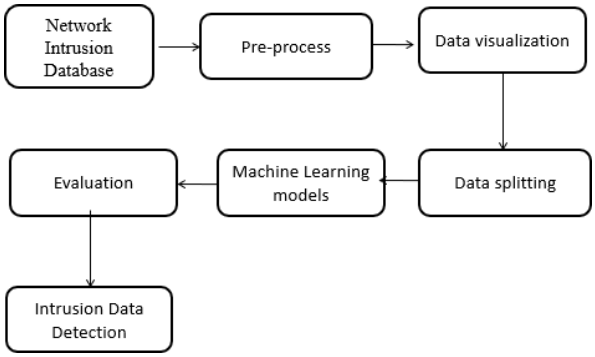
EXISTING METHOD

Traditional Network Traffic Analysis (NTA) uses rule-based and signature-based methods to detect threats by relying on predefined rules and known attack patterns, but these approaches fall short against zero-day and evolving cyber threats. To overcome this, machine learning techniques like Support Vector Machines (SVM), Decision Trees (DT), and Random Forests (RF) have been applied to automate and enhance NTA. SVMs are suited for classification, while DTs are effective in anomaly detection; however, these methods often demand large datasets and significant computational resources for optimal performance.

PROPOSED METHOD

The proposed Network Traffic Analysis for Intrusion Detection system utilizes Machine Learning (ML) algorithms to detect anomalies in real-time traffic by collecting and preprocessing network log data through cleaning, normalization, and feature selection. It employs regression models to track behavioral trends, K-Nearest Neighbours (KNN) to classify activities by comparing with historical data, and Decision Trees (DT) to improve accuracy by learning from past attacks. Continuously updated with real-time threat intelligence, the system adapts to evolving threats. Python-based libraries like Scikit-learn and Pandas support efficient model training and evaluation, aiming to enhance detection accuracy, reduce false positives, and strengthen network security.

BLOCK DIAGRAM



A working method for Network Traffic Analysis for Intrusion Detection using Python involves capturing real-time network packets using libraries like Scapy or PyShark, and extracting relevant features such as IP addresses, ports, protocols, and packet sizes. These features are then preprocessed and fed into a machine learning model—such as Random Forest, SVM, or a deep learning model like LSTM—trained on labeled datasets (e.g., NSL-KDD, CICIDS2017) to classify traffic as normal or malicious. The system continuously monitors the network, flags suspicious patterns based on the model's output, and logs or alerts administrators about potential intrusions for timely mitigation.

ADVANTAGES:

1. High Detection Accuracy
2. Real-Time Threat Detection
3. Adaptive Learning

APPLICATIONS:

1. Enterprise Network Security
2. Financial Sector Protection
3. Cloud Infrastructure Security

SOFTWARE REQUIREMENTS:

1. Jupyter Notebook
2. Anaconda navigator
3. OpenCV

HARDWARE REQUIREMENTS:

1. System: i5core
2. Hard Disk: 120 GB or above
3. Ram: 4 GB (min) or above

International Journal of Engineering Research and Science & Technology

ISSN 2319-5991 www.ijerst.com

Vol. 21, Issue 2, 2025

CONCLUSION

This methodology offers a robust solution for real-time network traffic analysis and intrusion detection using Python and machine learning, employing techniques like Regression, KNN, and Decision Trees to accurately classify malicious and benign traffic, strengthening cybersecurity defenses and reducing risks in modern networks.

REFERENCES

1. Srikavya, K. "Intrusion Detection System." [Online]. Available: https://github.com/srikavya26/Intrusion-Detection-System. [Accessed: Mar. 2, 2025].
2. Duggar, K. "Network Intrusion Detection System using Machine Learning." [Online]. Available: https://github.com/khushiduggar/Network-Intrusion-Detection-System-using-Machine-Learning. [Accessed: Mar. 2, 2025].
3. GeeksforGeeks. "Intrusion Detection System Using Machine Learning Algorithms." [Online]. Available: https://www.geeksforgeeks.org/intrusion-detection-system-using-machine-learning-algorithms/. [Accessed: Mar. 2, 2025].
4. Dimtics. "Network Intrusion Detection Using Machine Learning Techniques." [Online]. Available: https://github.com/dimtics/Network-Intrusion-Detection-Using-Machine-Learning-Techniques. [Accessed: Mar. 2, 2025].
5. Jadhav, Y., Deshpande, H., Garole, A., Sawant, K., and Patil, A. "Network Intrusion Detection System Using Decision Tree." *Journal of Network Security*, vol. 12, no. 2, pp. 22-33, 2024. [Online]. Available: https://journals.stmjournals.com/jons/article=2024/view=152713/. [Accessed: Mar. 2, 2025].
6. IEEE Xplore. "Network Intrusion Detection Using K-Nearest Neighbors (KNN) and Recurrent Neural Networks (RNN)." [Online]. Available: https://ieeexplore.ieee.org/document/10629867. [Accessed: Mar. 2, 2025].