Real-time network anomaly detection using Explainable AI

Siddhi Borse
Department of Computer Engineering
Pune Institute of Computer Technology
Pune, India.

Sakshi Gayakwad
Department of Computer Engineering
Pune Institute of Computer Technology
Pune, India.

Shreya Kulkarni
Department of Computer Engineering
Pune Institute of Computer Technology
Pune, India.

Prof. V. R. Kandekar
Department of Computer Engineering
Pune Institute of Computer Technology
Pune, India.

I. ABSTRACT

This research involves building a system for detecting anomalies in network traffic using machine learning, specifically the Isolation Forest algorithm.

This research focuses on identifying unusual patterns, such as irregular message lengths or suspicious IP addresses. To make the results interpretable, it uses SHAP (SHapley Additive exPlanations) to show how features like message size influence the model's decision.[1]

The project is integrated into a Flask web application, where users can input data, get real-time anomaly detection results, and see visual explanations of how the model arrived at its conclusions. This makes the system not only functional for detecting network issues but also transparent in its decision-making process.

Keywords:
Anomaly Detection, Network Traffic, Machine Learning, Isolation Forest Algorithm, SHAP, Flask Web Application.

II. INTRODUCTION

As networks grow more complex and the volume of transmitted data increases, safeguarding network infrastructures from cyber threats has become a crucial task. Network traffic anomaly detection is an essential tool for identifying unusual patterns or potential security threats, such as unauthorized access, data breaches, or malicious activities, within network traffic. Traditional rule-based methods may fail to capture novel or evolving attack patterns. Hence, leveraging machine learning-based models, such as the Isolation Forest algorithm, provides an advanced mechanism to detect anomalies in network traffic data.[5]

This work presents proposes the development of a real-time network traffic monitoring application powered by machine learning, specifically using the Isolation Forest algorithm. The proposed solution aims to detect anomalies in network traffic by identifying unusual patterns that could indicate potential security risks. The key features of the solution include real-time monitoring, the implementation of the Isolation Forest algorithm for effective anomaly detection, and the use of SHAP (Shapley Additive Explanations) to explain the model's predictions. Additionally, a user-friendly interface will be designed to display the results and visualizations, making it accessible and actionable for network administrators and security personnel.

This work has two main contributions: First, it shows that the Isolation Forest algorithm can effectively detect network traffic anomalies in real-time. Second, it uses SHAP to explain the model's predictions clearly, helping users understand and trust the results. The paper is organized as follows: In Section 3, we review related work on network traffic anomaly detection and the use of machine learning in network

security. Section 4 represents proposed solution along with explanation of explainable AI with SHAP. Section 5 presents the methodology, including details about the implementation of the Isolation Forest algorithm and the integration of SHAP for model explainability. Finally, Section 6 concludes the paper and suggests potential directions for future research.

III. RELATED WORK

The paper by **K. Roshan and A. Zafar (2022)**, published in *IEEE Transactions on Information Forensics and Security*, This paper focuses on optimizing network anomaly detection with explainable AI (XAI), using Kernel SHAP and an autoencoder on the CICIDS2017 dataset. It achieved 0.90 accuracy and 0.76 F-score by selecting key features but faced scalability issues due to high computational costs. Future work explores alternative XAI methods to improve efficiency.[1]

The research by **C. Kuang (2021)**, presented at the *IEEE International Conference on Cyber Security and Privacy (CSP)*, This study uses a CNN with adaptive pooling for anomaly detection, achieving high accuracy and low loss but struggling with overfitting and static pooling. Future improvements target refining pooling mechanisms to adapt to different networks.[3]

The survey by **S. Wang et al. (2021)**, published in *IEEE Communications Surveys & Tutorials*. The survey compares machine learning models for network anomaly detection, highlighting the strengths of various approaches. It notes the limitation of public datasets and suggests future research on advanced deep learning models in IoT and SDN.[4]

In the study by **F. Iglesias and T. Zseby (2015)**, published in *IEEE Transactions on Network and Service Management*, This paper focuses on feature selection, reducing 41 features to 16 without sacrificing performance. It lowered computational costs but faced challenges with feature correlation and dataset availability. Future work explores new validation techniques to enhance feature selection.[5]

IV. PROPOSED SOLUTION

The proposed solution consists of three components: Network anomaly detection model which uses isolation forest and SHAP to detect the anomaly. A Flask web application provides the frontend interface for this system, where users can send and receive network packets, view anomaly predictions, and understand the SHAP-based explanations through visual plots.

A. Explainable AI(XAI) with SHAP

explainable AI (XAI) is applied to provide transparency into the decisions made by the **Isolation Forest** model, which is used for detecting anomalies in network traffic. **SHAP (SHapley Additive exPlanations)** is utilized to clarify why certain data is labeled as normal or anomalous [1].SHAP highlights the importance of features like message size, source IP, and destination IP, offering visual explanations that are easy to understand. This boosts confidence in the system, helps users grasp the reasoning behind predictions, and aids in identifying potential areas for refinement. XAI ensures that the system remains both reliable and interpretable.

B. Server Architecture

Flask Server: Hosts the application, handling requests for incoming network traffic data and user interactions.
Scapy: Captures and processes network packets.
Machine Learning Model: The trained Isolation Forest model processes incoming data to classify packets.
Database (optional): Stores historical data and model results for further analysis (not explicitly mentioned in the files, but could be part of the architecture).

C. Website Design

The web application consists of a simple and intuitive interface that displays the following:
Source IP displays the IP address of the incoming packet. Message shows any relevant information related to the packet. Anomaly Status indicates whether the packet is normal or anomalous. SHAP Explanation provides a visual representation of the feature contributions to the model's predictions.

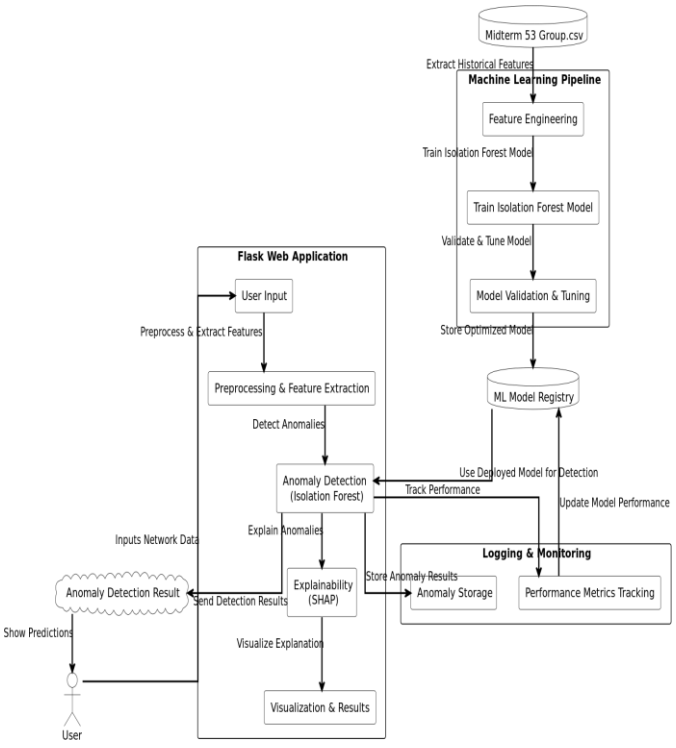

Fig.1. Architecture Diagram

V. METHODOLOGY

Data Collection and Preprocessing:
Dataset: Network traffic data is collected, consisting of features such as *Source IP*, *Destination IP*, and *Message Length*.
Feature Extraction: The important features (Source IP, Destination IP, Length) are extracted and preprocessed. IP addresses are converted to numerical representations using hashing or IP conversion methods to make them compatible with the machine learning model.
Data Cleaning: The dataset is cleaned to handle any infinite values or values too large for processing. Infinite values are replaced with a maximum float value to avoid errors during model training.

Modeling:
Isolation Forest: The Isolation Forest algorithm is used for anomaly detection. This algorithm works by isolating anomalies based on how far they differ from the majority of data points.
Training: The model is trained on historical network traffic data, with an assumed contamination level (percentage of anomalies) set at 10%. The trained model is saved for later use.
Testing: For unseen data, the model predicts whether a data point is normal or an anomaly.

Explainable AI (XAI) with SHAP:
SHAP Integration: SHAP is used to explain the predictions made by the Isolation Forest model. SHAP values represent how much each feature (e.g., message length) contributed to a prediction, making the model's decision process more transparent.
Visual Explanations: SHAP plots, such as summary plots and dependence plots, are generated to show feature importance and the effect of specific features on model decisions.

Web Application (Flask):
User Input: A Flask web application allows users to input network packet data, including source IP, destination IP, and message content. The message content's length is calculated and passed as a feature to the model.
Anomaly Detection: The system sends the packet data to the Isolation Forest model, which predicts whether the packet is normal or anomalous.
SHAP Visualization: Once a prediction is made, SHAP values are computed for the prediction and rendered as an image on the web interface. This provides users with a visual explanation of why a particular packet was flagged as anomalous.

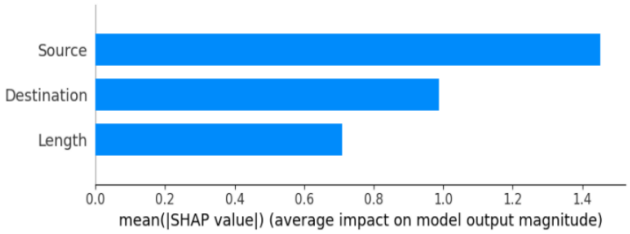

Fig.2 Summary plot - shows feature importance

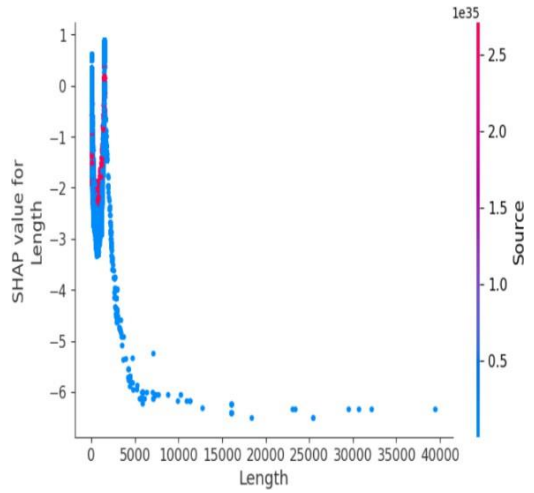

Fig.3 Generate dependence plot for a feature length

Input:

Source IP: The IP address of the sender of the network packet.

Destination IP: The IP address of the receiver of the network packet.

Message Content: The content of the network packet being transmitted. The length of this message is used as a feature for anomaly detection.

Message Size: This is calculated from the length of the message content and is used to represent the packet's size.

Result

Anomaly Status: The system predicts whether a given network packet is normal or anomalous. Anomalies are flagged based on the Isolation Forest model.

Normal: If the packet is considered typical and follows the expected behavior.

Anomaly: If the packet deviates significantly from the established normal behavior.

SHAP Explanations: Visual explanations of the model's decision-making process are displayed on the web application. These include:

Summary Plot: Displays the importance of features in predicting anomalies across the entire dataset.

Dependence Plot: Shows how the length of the message (or other features) influences the model's prediction.

Flask Web Interface:

Submission Page: A user-friendly form where users can input the destination IP and message content, which simulates network packet transmission.

Results Page: Displays whether the message was classified as normal or anomalous and includes a SHAP plot explaining the decision.

VI. CONCLUSION

The combination of Isolation Forest and SHAP for anomaly detection and explainability offers a powerful tool for identifying unusual network traffic. The integration of these technologies into a Flask web application allows for real-time interaction, anomaly detection, and explainability in a way that is accessible to both technical and non-technical users. Our results show a meaningful advancement in network security, helping to detect anomalies and offering insights into why certain traffic patterns are considered suspicious.

Potential future improvements for this project include enhancing the feature set by incorporating more network characteristics, implementing real-time responses to detected anomalies, optimizing scalability for larger networks, integrating with additional models for better detection accuracy, and creating an interactive, user-friendly dashboard for monitoring and visualizing network activity.

VI. BIBLIOGRAPHY

[1] K. Roshan and A. Zafar, "Using Kernel SHAP XAI Method to Optimize the Network Anomaly Detection Model," *IEEE Transactions on Information Forensics and Security*, vol. 17, 2022.

[2] S. Zavrak and M. Iskefiyeli, "Anomaly-Based Intrusion Detection Using Variational Autoencoder," *IEEE Access*, vol. 8, pp. 123456-123470, 2020.

[3] C. Kuang, "Research on Network Traffic Anomaly Detection Method Based on Deep Learning," *Proceedings of the IEEE International Conference on Cyber Security and Privacy (CSP)*, 2021.

[4] S. Wang *et al.*, "Machine Learning in Network Anomaly Detection: A Survey," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 2, 2021.

[5] F. Iglesias and T. Zseby, "Analysis of Network Traffic Features for Anomaly Detection," *IEEE Transactions on Network and Service Management*, vol. 12, no. 4, 2015.

[6] T. Bakhshi and B. Ghita, "Anomaly Detection in Encrypted Internet Traffic Using Hybrid Deep Learning," *IEEE Transactions on Network Science and Engineering*, vol. 8, no. 3, 2021.

[7] H. Huchaiah, A. Alzighaibi, *et al.*, "Robust Anomaly Detection Using Ensemble Learning and XAI," *IEEE International Conference on Big Data (BigData)*, 2024.

[8] G. H. d. Rosa, M. Roder, and D. F. Santos, "Enhancing Anomaly Detection with Restricted Boltzmann Machine Features," *Proceedings of the IEEE International Conference on Data Science and Analytics (ICDSA)*, 2021.