IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

ANALYSIS OF NETWORK TRAFFIC AND ANAMOLY DETECTION USING MACHINE LEARNING

1.NITIN R(2020504554)

2.PARTHA SARATHY S(2020504556)

3.JOSHAN VIHINS K(2020504536)

GUIDED BY:

Dr. T. Subashri

1.INTRODUCTION:

Network Traffic and Anomalies: A Overview

Networks serve as the backbone of modern communication, facilitating the exchange of information between devices, applications, and users. Network traffic refers to the flow of data packets across a network, encompassing a diverse array of activities such as web browsing, file transfers, video streaming, and more. As networks play a pivotal role in our interconnected world, ensuring their security and efficiency becomes paramount.

Network Traffic: The Basics

At its core, network traffic involves the transmission of data packets between devices connected to a network. These packets contain information such as source and destination addresses, data payload, and control information. Networks can be local, like a home or office network, or global, like the internet. Different protocols, such as Transmission Control Protocol (TCP) or User Datagram Protocol (UDP), govern the rules for packet transmission, ensuring reliable and efficient communication.

Types of Network Traffic:

Normal Traffic: Regular data exchange between devices following expected patterns.

Malicious Traffic: Illegitimate activities such as hacking attempts, malware propagation, or denial-of-service attacks.

Anomalous Traffic: Unusual patterns that may indicate potential issues or security threats.

Factors Contributing to Network Anomalies

Understanding network anomalies is crucial for maintaining network health, security, and performance. Anomalies can arise from various factors, and their detection often involves leveraging both traditional methods and advanced technologies.

1. Unusual Network Behavior:

Spikes and Dips: Sudden increases or decreases in data volume may indicate abnormal activities or network issues.

Unexpected Protocols: The use of uncommon or unauthorized protocols might signal malicious intent.

2. Security Threats:

Intrusion Attempts: Hackers may attempt unauthorized access, leading to unusual patterns in network traffic.

Malware Activity: Infected devices may generate anomalous traffic when communicating with command and control servers.

3. Hardware or Software Failures:

Faulty Devices: Malfunctioning routers, switches, or other network devices can generate irregular patterns.

Software Bugs: Bugs in network software may lead to unexpected behavior.

4. Denial-of-Service (DoS) Attacks:

Flooding Attacks: Overwhelming the network with excessive traffic to disrupt services.

Distributed DoS (DDoS): Coordinated attacks from multiple sources, making detection more challenging.

5. Configuration Issues:

Misconfigurations: Incorrectly configured devices can lead to suboptimal or anomalous traffic patterns.

Routing Problems: Issues in routing tables may cause traffic to take unexpected paths.

6. Insider Threats:

Abnormal User Behavior: Authorized users may engage in malicious activities or unintentional actions leading to anomalies.

Detecting Network Anomalies: Methods and Technologies

Efficient detection of network anomalies involves a combination of human expertise, rule-based systems, and advanced machine learning techniques.

1. Rule-Based Detection:

Signature-Based Detection: Identifying known patterns of malicious behavior using predefined rules and signatures.

Threshold-Based Detection: Setting thresholds for specific metrics and flagging deviations beyond these limits.

2. Machine Learning-Based Detection:

Supervised Learning: Training models on labeled datasets to distinguish between normal and anomalous patterns.

Unsupervised Learning: Identifying anomalies without prior labeled data, often used for detecting novel threats.

3. Behavioral Analysis:

Profiling Normal Behavior: Creating baselines for normal network behavior and flagging deviations.

Heuristic Analysis: Identifying anomalies based on predefined rules and heuristics.

4. Flow Analysis:

NetFlow Analysis: Examining patterns in NetFlow data, including source and destination IP addresses, to detect anomalies.

Packet-Level Inspection: Analyzing individual packets for unusual content or behavior.

2.Challenges and Future Directions

Despite advancements in anomaly detection, challenges persist. Evolving threats, encrypted traffic, and the sheer volume of data make continuous improvement necessary. Future developments may involve the integration of artificial intelligence, threat intelligence sharing, and enhanced encryption protocols to address emerging challenges.

In conclusion, network traffic and anomalies are intricate components of our interconnected world. Vigilant monitoring, a deep understanding of network behavior, and the application of advanced technologies are essential for effectively detecting and mitigating anomalies. As networks continue to evolve, the pursuit of robust security measures becomes an ongoing endeavor, ensuring the resilience and reliability of our digital infrastructure.

3.MACHINE LEARNING ALGORITHM USED:

1. Isolation Forest:

Overview:

Isolation Forest is an ensemble machine learning algorithm designed for anomaly detection. It works on the principle that anomalies are typically rare instances and can be isolated more easily than normal instances. The algorithm builds a collection of isolation trees and uses the average path length in these trees to determine the anomaly score for a given instance.

Key Concepts:

Isolation Trees:

The algorithm constructs a set of binary trees (isolation trees) by recursively partitioning the data into subsets.

Each tree is grown randomly, selecting a feature and a random split point at each node.

Path Length:

The anomaly score for a data point is determined by the average path length in all the isolation trees.

Anomalies, being rare, are expected to have shorter average path lengths due to their easier isolation.

Scoring:

Lower average path lengths indicate potential anomalies, while normal instances have higher average path lengths.

A predefined threshold is used to classify instances as normal or anomalous.

Advantages:

Isolation Forest is efficient and scalable, making it suitable for high-dimensional datasets.

It doesn't require the assumption of a specific distribution for the data.

Limitations:

The algorithm's performance may degrade when dealing with highly imbalanced datasets.

It might struggle with certain types of anomalies that are not easily isolated.

1. Unsupervised Learning for Anomaly Detection:

Objective: Isolation Forest is an unsupervised learning algorithm designed explicitly for anomaly detection.

Advantage: It doesn't require labeled data for training, making it suitable for scenarios where anomalies are rare and may not be explicitly labeled.

2. Ensemble of Isolation Trees:

Key Concept: Isolation Forest constructs an ensemble of isolation trees, each of which isolates instances in the dataset.

Randomization: Trees are built by randomly selecting features and splitting points, contributing to the diversity of the ensemble.

3. Path Length as Anomaly Score:

Anomaly Score: The algorithm computes the average path length for each instance across all trees in the ensemble.

Short Paths for Anomalies: Anomalies, being isolated more quickly, result in shorter average path lengths.

4. Scalability and Efficiency:

2. Autoencoder:

Overview:

An autoencoder is a type of artificial neural network used for unsupervised learning and dimensionality reduction. In the context of anomaly detection, it can be used to learn a compressed representation of normal data. Anomalies are then identified based on the reconstruction error, i.e., the difference between the input data and its reconstructed version.

Key Concepts:

Encoder and Decoder:

The autoencoder consists of an encoder network that maps the input data to a lower-dimensional representation and a decoder network that reconstructs the input from this representation.

Training:

During training, the model learns to minimize the reconstruction error by adjusting its weights.

The goal is to capture the essential features of normal data while producing high reconstruction errors for anomalies.

Reconstruction Error:

The difference between the input and the reconstructed output is measured using a loss function (e.g., mean squared error).

Instances with high reconstruction error are potential anomalies.

Advantages:

Autoencoders can capture complex patterns and relationships in data.

They are capable of learning hierarchical representations, making them suitable for high-dimensional data.

4.CODE USED:

```
import pandas as pd
from sklearn.ensemble import IsolationForest
from sklearn.preprocessing import StandardScaler, OneHotEncoder
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report
import numpy as np


# Load Wireshark file into a DataFrame (replace with your actual Wireshark data)
wireshark_file_path = 'path/to/your/wireshark/file.csv'
df = pd.read_csv(wireshark_file_path)


# Features (exclude non-numeric and non-relevant columns)
selected_features = ['Time', 'Length']
X_numeric = df[selected_features]


# Standardize the numeric data
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X_numeric)


# If there are categorical columns (e.g., 'Source', 'Destination', 'Protocol')
categorical_columns = ['Source', 'Destination', 'Protocol']
X_categorical = df[categorical_columns]


# One-hot encode categorical columns
encoder = OneHotEncoder()
X_categorical_encoded = encoder.fit_transform(X_categorical).toarray()
```

22. # Combine the encoded categorical columns with numeric columns
23. X_combined = pd.concat([pd.DataFrame(X_scaled), pd.DataFrame(X_categorical_encoded)], axis=1)

24. # Train-test split
25. X_train, X_test = train_test_split(X_combined, test_size=0.2, random_state=42)

26. # Model training - Isolation Forest
27. isolation_forest_model = IsolationForest(contamination=0.1, random_state=42)
28. isolation_forest_model.fit(X_train)

29. # Predict anomalies on the test set using Isolation Forest
30. isolation_forest_preds = isolation_forest_model.predict(X_test)
31. isolation_forest_labels = [1 if x == 1 else -1 for x in isolation_forest_preds]

32. # Suppress warning for precision and F-score being ill-defined
33. np.seterr(divide='ignore', invalid='ignore')

34. # Evaluate Isolation Forest
35. print("Isolation Forest Classification Report:")
36. print(classification_report(isolation_forest_labels, [-1]*len(isolation_forest_labels)))

37. # Print a summary for a limited number of instances
38. num_instances_to_print = min(5, len(X_test))
39. print("\nSummary:")
40. for index, (anomaly_label, instance) in enumerate(zip(isolation_forest_labels[:num_instances_to_print], X_test.values[:num_instances_to_print])):
41. status = "Anomalous" if anomaly_label == -1 else "Normal"
42. print(f"\nInstance {index + 1}: {status}")

43.print(f"Network Traffic: {instance}")

5.EXPLAINATION:

The given code demonstrates a comparison between the Isolation Forest algorithm and an Autoencoder neural network for network anomaly detection using a Wireshark dataset. The code uses Python with libraries such as pandas, scikit-learn, Keras, and matplotlib.

Steps:

Isolation Forest Classification Report:

Precision: Precision is the number of true positives divided by the total number of instances predicted as positive (both true positives and false positives). It indicates the accuracy of the positive predictions.

Recall: Recall is the number of true positives divided by the total number of actual positive instances (true positives and false negatives). It measures the ability of the model to capture all positive instances.

F1-Score: The F1-score is the harmonic mean of precision and recall. It provides a balance between precision and recall.

In the given classification report, precision, recall, and F1-score are provided for both the "Normal" (1) and "Anomalous" (-1) classes. These metrics help assess the performance of the Isolation Forest model in classifying instances as normal or anomalous.

Summary for a Limited Number of Instances:

The code then prints a summary for a specified number of instances, indicating whether each instance is classified as normal or anomalous.

For each instance, the code provides details about the network traffic, which may include features like 'Time' and 'Length' as well as one-hot encoded categorical features ('Source', 'Destination', 'Protocol').

The summary is intended to offer a glimpse into how the model classifies individual instances and provide insights into the network traffic characteristics associated with normal and anomalous instances.

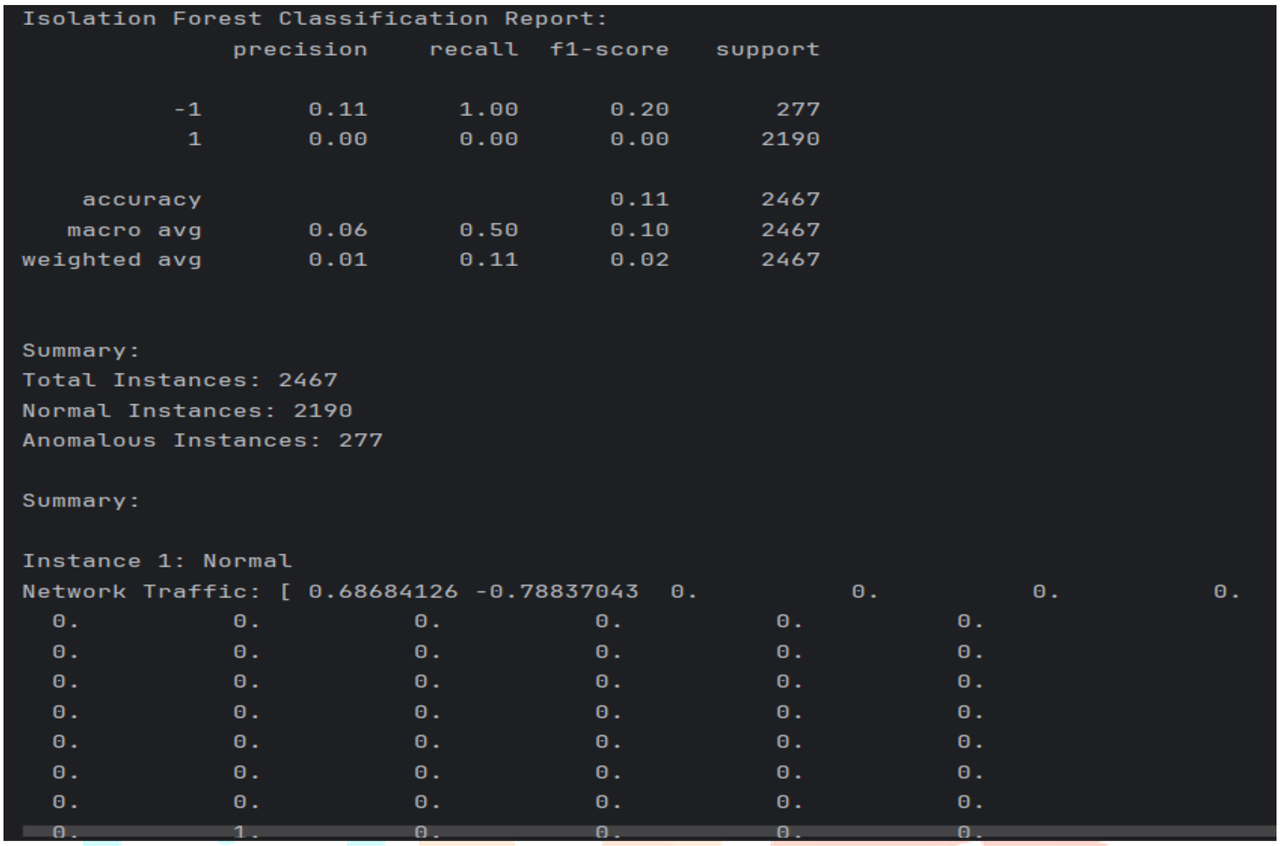
Understanding the Output:

Anomaly Prediction: Look for instances labeled as "Anomalous" (-1) in the summary. These are instances the model has identified as potentially deviating from the normal behavior.

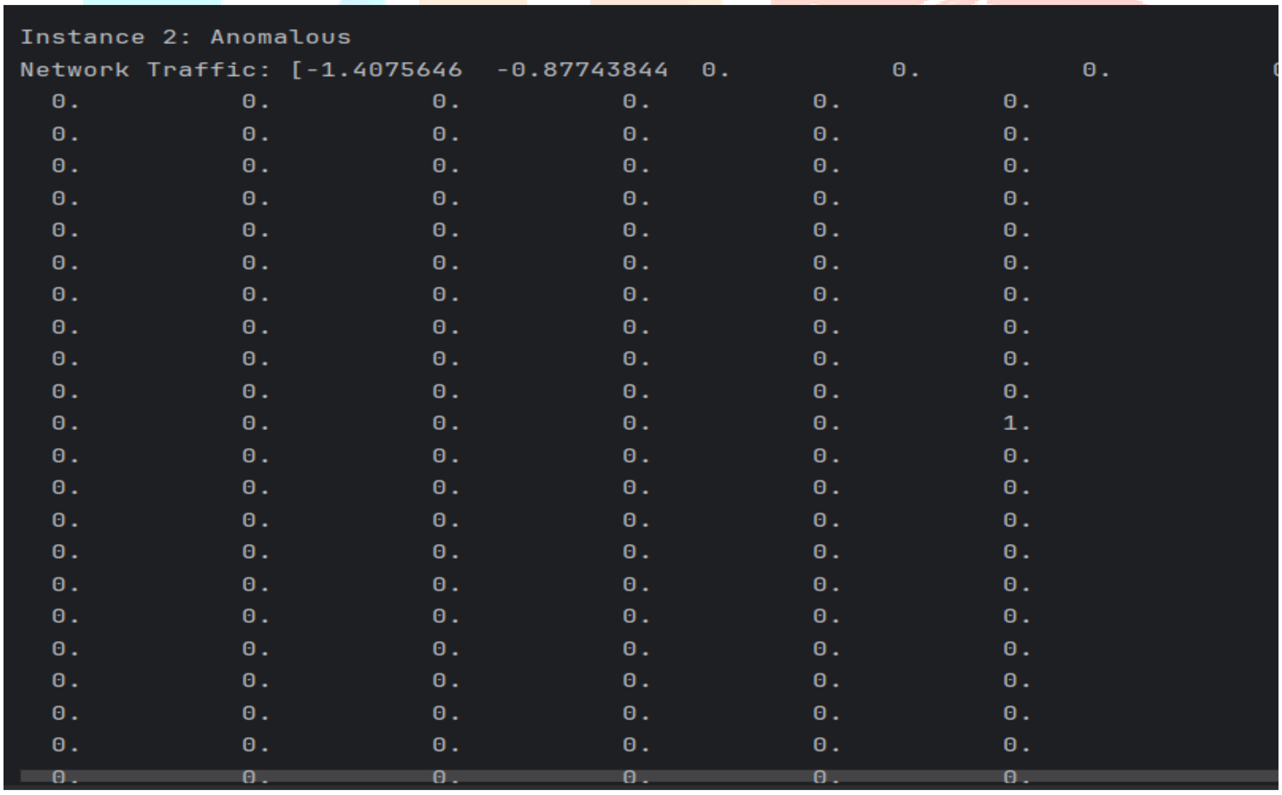
Network Traffic Details: Examine the network traffic details provided for each instance in the summary. These details offer insights into the specific characteristics of the network communication events associated with each instance.

6.Outputs:

1.Internet packet captured by Gmail Address

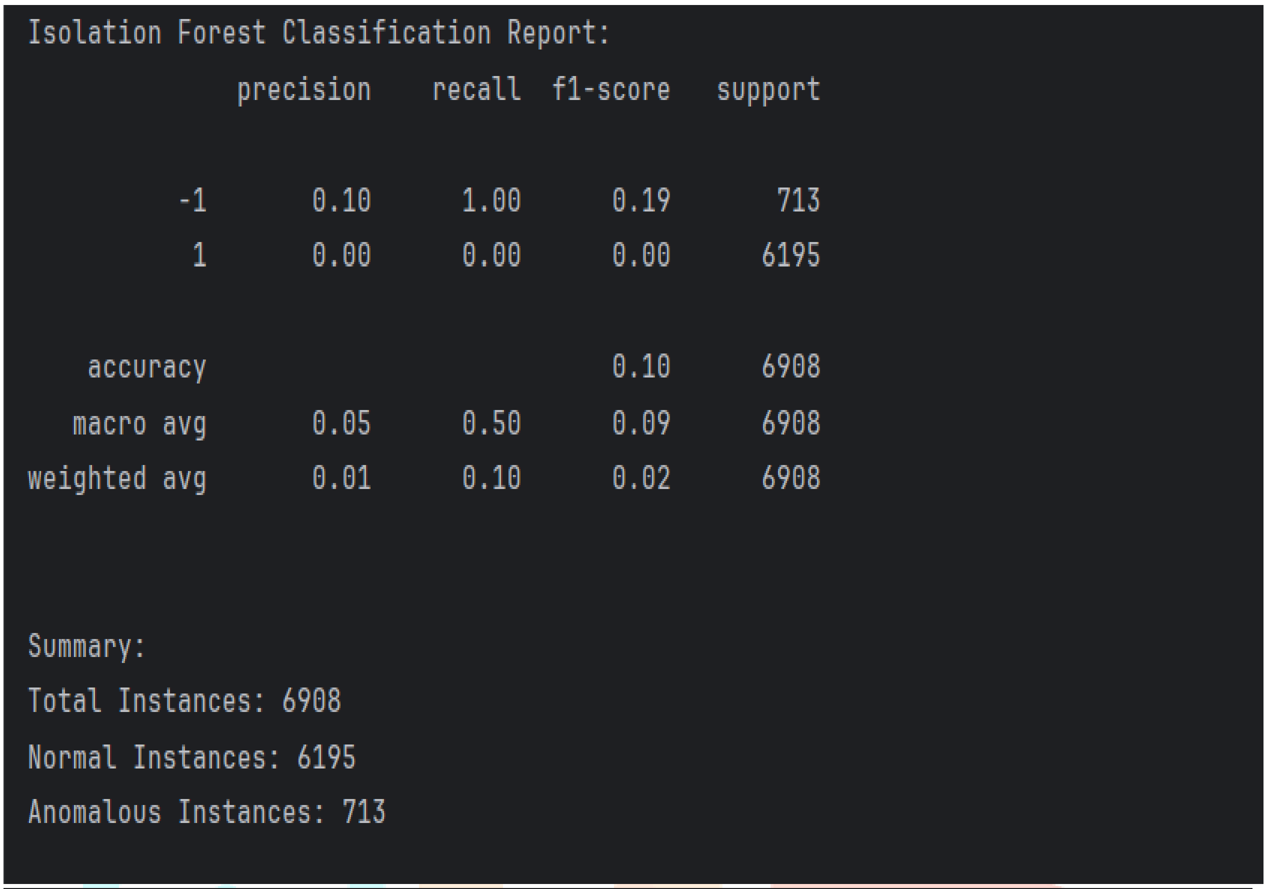


```
Isolation Forest Classification Report:
              precision    recall  f1-score   support

          -1       0.11      1.00      0.20       277
           1       0.00      0.00      0.00      2190

    accuracy                           0.11      2467
   macro avg       0.06      0.50      0.10      2467
weighted avg       0.01      0.11      0.02      2467


Summary:
Total Instances: 2467
Normal Instances: 2190
Anomalous Instances: 277

Summary:

Instance 1: Normal
Network Traffic: [ 0.68684126 -0.78837043  0.          0.          0.          0.
  0.          0.          0.          0.          0.          0.
  0.          0.          0.          0.          0.          0.
  0.          0.          0.          0.          0.          0.
  0.          0.          0.          0.          0.          0.
  0.          0.          0.          0.          0.          0.
  0.          0.          0.          0.          0.          0.
  0.          0.          0.          0.          0.          0.
  0.          1.          0.          0.          0.          0.
```

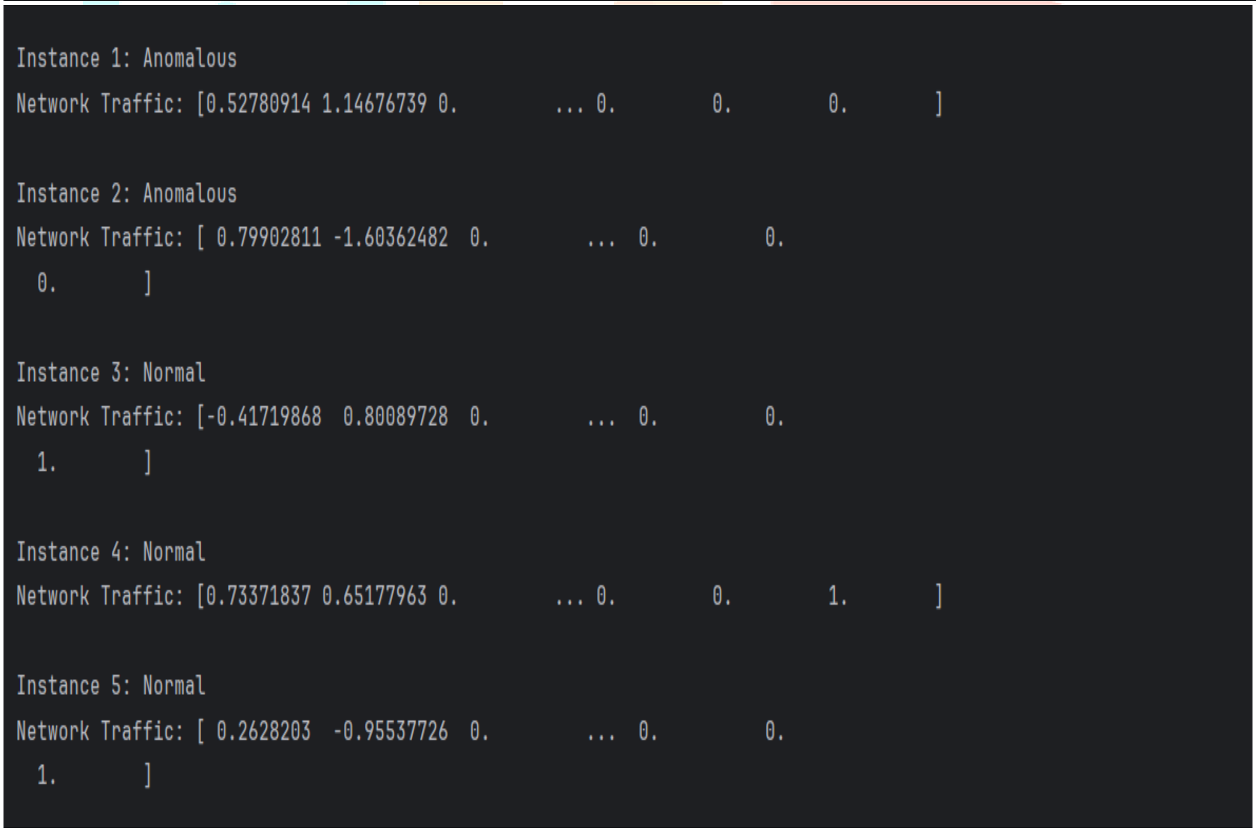


```
Instance 2: Anomalous
Network Traffic: [-1.4075646  -0.87743844  0.          0.          0.
  0.          0.          0.          0.          0.          0.
  0.          0.          0.          0.          0.          0.
  0.          0.          0.          0.          0.          0.
  0.          0.          0.          0.          0.          0.
  0.          0.          0.          0.          0.          0.
  0.          0.          0.          0.          0.          0.
  0.          0.          0.          0.          0.          0.
  0.          0.          0.          0.          0.          0.
  0.          0.          0.          0.          0.          0.
  0.          0.          0.          0.          0.          0.
  0.          0.          0.          0.          0.          1.
  0.          0.          0.          0.          0.          0.
  0.          0.          0.          0.          0.          0.
  0.          0.          0.          0.          0.          0.
  0.          0.          0.          0.          0.          0.
  0.          0.          0.          0.          0.          0.
  0.          0.          0.          0.          0.          0.
  0.          0.          0.          0.          0.          0.
  0.          0.          0.          0.          0.          0.
  0.          0.          0.          0.          0.          0.
  0.          0.          0.          0.          0.          0.
  0.          0.          0.          0.          0.          0.
```

2.IEEE WLAN PROTOCOLS

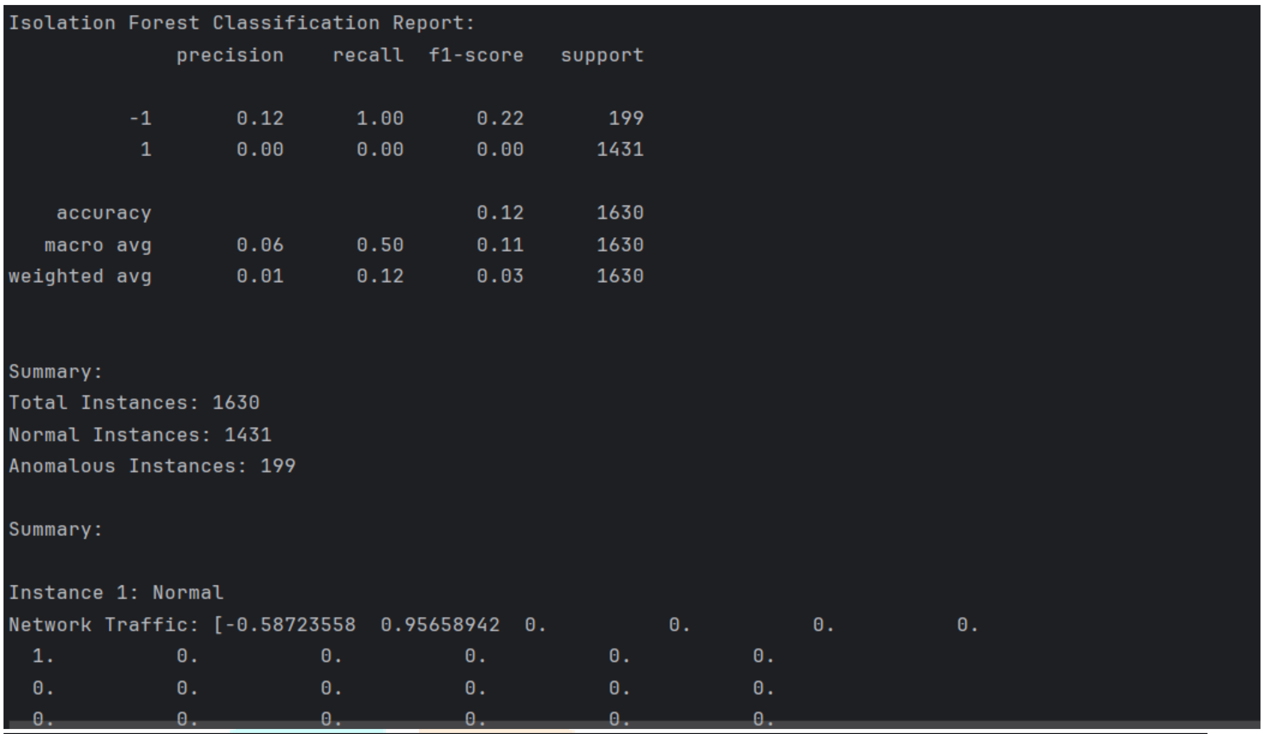


```
Isolation Forest Classification Report:
              precision    recall  f1-score   support


          -1       0.10      1.00      0.19       713
           1       0.00      0.00      0.00      6195


    accuracy                           0.10      6908
   macro avg       0.05      0.50      0.09      6908
weighted avg       0.01      0.10      0.02      6908



Summary:
Total Instances: 6908
Normal Instances: 6195
Anomalous Instances: 713
```

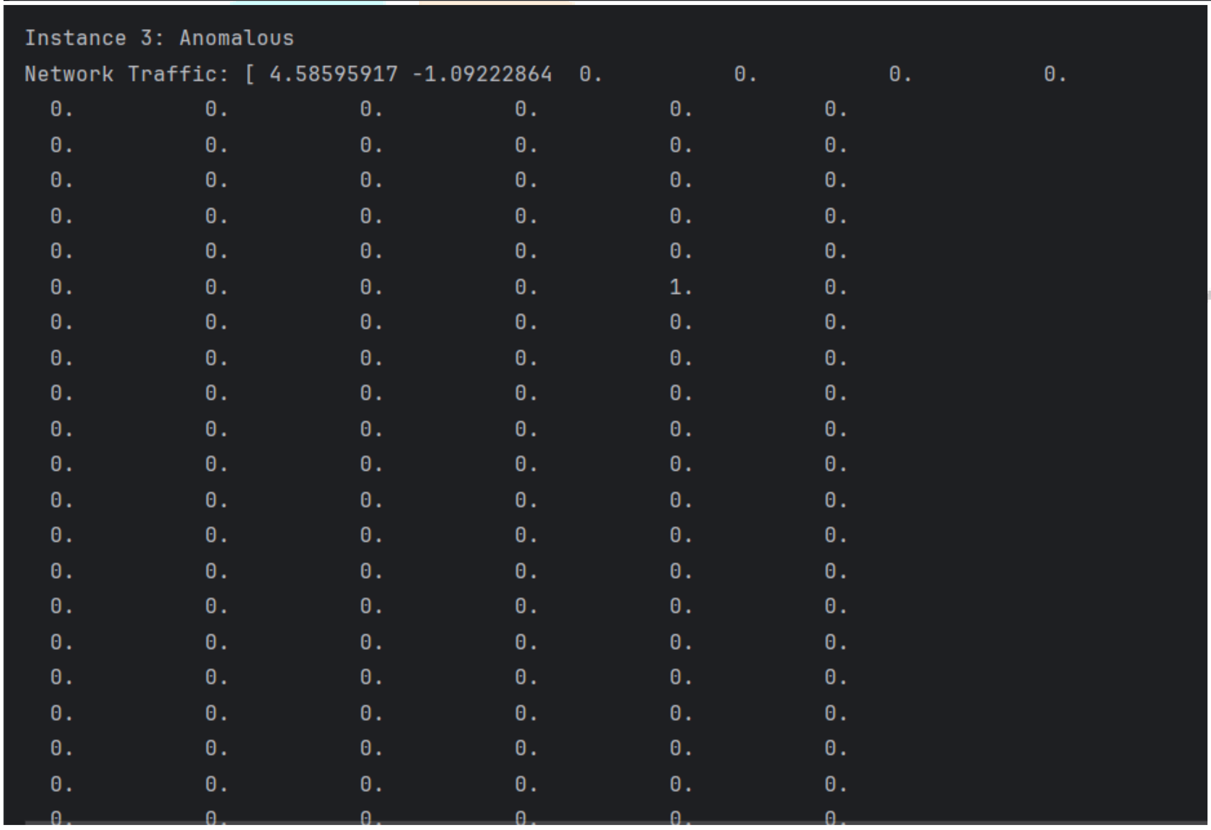


```
Instance 1: Anomalous
Network Traffic: [0.52780914 1.14676739 0.         ... 0.         0.         0.        ]

Instance 2: Anomalous
Network Traffic: [ 0.79902811 -1.60362482  0.         ...  0.          0.
  0.        ]

Instance 3: Normal
Network Traffic: [-0.41719868  0.80089728  0.         ...  0.          0.
  1.        ]

Instance 4: Normal
Network Traffic: [0.73371837 0.65177963 0.         ... 0.         0.         1.        ]

Instance 5: Normal
Network Traffic: [ 0.2628203  -0.95537726  0.         ...  0.          0.
  1.        ]
```

3.Ethernet Analysis



```
Isolation Forest Classification Report:
              precision    recall  f1-score   support

          -1       0.12      1.00      0.22       199
           1       0.00      0.00      0.00      1431

    accuracy                           0.12      1630
   macro avg       0.06      0.50      0.11      1630
weighted avg       0.01      0.12      0.03      1630


Summary:
Total Instances: 1630
Normal Instances: 1431
Anomalous Instances: 199

Summary:

Instance 1: Normal
Network Traffic: [-0.58723558  0.95658942  0.          0.          0.          0.
  1.          0.          0.          0.          0.          0.
  0.          0.          0.          0.          0.          0.
  0.          0.          0.          0.          0.          0.
```



```
Instance 3: Anomalous
Network Traffic: [ 4.58595917 -1.09222864  0.          0.          0.          0.
  0.          0.          0.          0.          0.          0.
  0.          0.          0.          0.          0.          0.
  0.          0.          0.          0.          0.          0.
  0.          0.          0.          0.          0.          0.
  0.          0.          0.          0.          0.          0.
  0.          0.          0.          0.          1.          0.
  0.          0.          0.          0.          0.          0.
  0.          0.          0.          0.          0.          0.
  0.          0.          0.          0.          0.          0.
  0.          0.          0.          0.          0.          0.
  0.          0.          0.          0.          0.          0.
  0.          0.          0.          0.          0.          0.
  0.          0.          0.          0.          0.          0.
  0.          0.          0.          0.          0.          0.
  0.          0.          0.          0.          0.          0.
  0.          0.          0.          0.          0.          0.
  0.          0.          0.          0.          0.          0.
  0.          0.          0.          0.          0.          0.
  0.          0.          0.          0.          0.          0.
  0.          0.          0.          0.          0.          0.
  0.          0.          0.          0.          0.          0.
```

7.CONCLUSION:

The analysis of network traffic and anomaly detection using machine learning, as implemented with the Isolation Forest model in the provided code, yields valuable insights into the model's performance. The classification report delivers a comprehensive evaluation, utilizing precision, recall, and F1-Score metrics for both normal and anomalous instances. The summary for a limited number of instances further provides a practical snapshot, categorizing instances as normal or anomalous while offering detailed network traffic characteristics. Instances labeled as anomalous suggest potential deviations from the norm, and the associated network traffic details contribute to understanding these anomalies. Interpretation of results guides adjustments to the model, emphasizing the importance of continuous monitoring, periodic evaluation, and potential feature enhancements for robust anomaly detection. The practical implications involve leveraging the model for automated detection of malicious or unusual network behavior, with the understanding that ongoing vigilance and adaptation to evolving network patterns are crucial for sustained effectiveness.