|  |
| --- |
| **Write a R Program to Create a Anomaly Detection in Network Traffic** |

**Aim:**

The primary aim of anomaly detection in network traffic is to enhance the security, reliability, and efficiency of network operations by identifying and addressing abnormal patterns that could indicate potential issues or threats.

**Objective:**

The objectives of anomaly detection in network traffic focus on enhancing network security, performance, and reliability. Here are the key objectives:

1. **Early Detection of Security Threats:**
   * Identify and respond to potential intrusions, malware, and other cyber threats in real-time.
   * Prevent data breaches and unauthorized access by recognizing suspicious activities early.
2. **Maintaining Network Performance:**
   * Detect performance-related anomalies such as congestion, bottlenecks, or hardware malfunctions.
   * Ensure consistent network performance and minimize downtime by addressing issues promptly.
3. **Improving Network Health Monitoring:**
   * Continuously monitor network traffic to maintain an overview of the network's health.
   * Detect and resolve anomalies that may indicate underlying problems before they escalate.
4. **Reducing False Positives:**
   * Develop and implement accurate anomaly detection models to minimize the occurrence of false positives.
   * Ensure that alerts are meaningful and correspond to actual anomalies requiring attention.
5. **Adapting to Dynamic Network Conditions:**
   * Implement adaptive algorithms that can learn and adjust to changing network patterns and behaviors.
   * Maintain effectiveness in anomaly detection even as network conditions and usage patterns evolve.
6. **Enhancing Regulatory Compliance:**
   * Ensure continuous monitoring and logging of network activities to meet regulatory and compliance requirements.
   * Detect and report any deviations from normal traffic patterns that might indicate compliance breaches.
7. **Proactive Threat Mitigation:**
   * Enable proactive measures to mitigate potential threats by providing timely alerts and actionable insights.
   * Strengthen network defenses through early intervention based on detected anomalies.
8. **Supporting Incident Response and Forensics:**
   * Provide detailed anomaly reports that can aid in incident response and forensic investigations.
   * Help identify the root cause of anomalies and support post-incident analysis and remediation.
9. **Facilitating Network Management and Optimization:**
   * Assist network administrators in optimizing network configurations and resource allocations based on anomaly patterns.
   * Enable data-driven decision-making for network upgrades and capacity planning.
10. **Improving User Experience:**
    * Ensure a smooth and uninterrupted user experience by detecting and resolving issues that could impact network services.
    * Enhance overall service quality by maintaining a stable and secure network environment.

**Program:**

install.packages(c("tidyverse", "anomalize", "httr"))

# Load necessary libraries

library(tidyverse)

library(anomalize)

library(httr)

# Simulated network traffic data

set.seed(123)

timestamps <- seq.POSIXt(from = as.POSIXct("2023-01-01"), by = "hour", length.out = 1000)

traffic <- rnorm(1000, mean = 100, sd = 10)

traffic[500:505] <- traffic[500:505] + 50 # Injecting anomalies

data <- tibble(timestamp = timestamps, traffic = traffic)

# Plot the data

ggplot(data, aes(x = timestamp, y = traffic)) +

geom\_line() +

labs(title = "Simulated Network Traffic Data", x = "Time", y = "Traffic")

# Anomaly detection using anomalize

data\_anomalized <- data %>%

time\_decompose(traffic, method = "stl") %>%

anomalize(remainder, method = "iqr") %>%

time\_recompose()

# Plot the anomalies

plot\_anomalies(data\_anomalized, time\_recomposed = TRUE) +

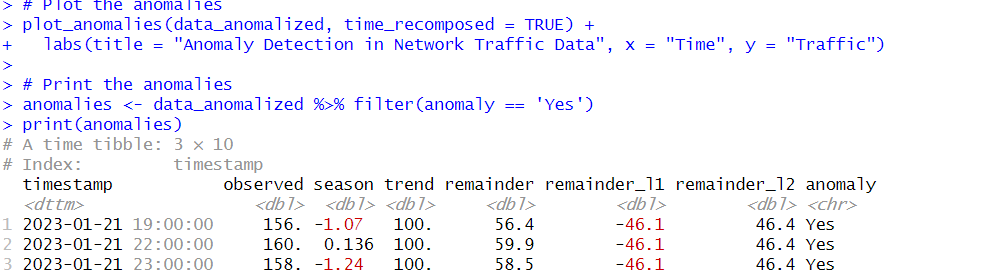
labs(title = "Anomaly Detection in Network Traffic Data", x = "Time", y = "Traffic")

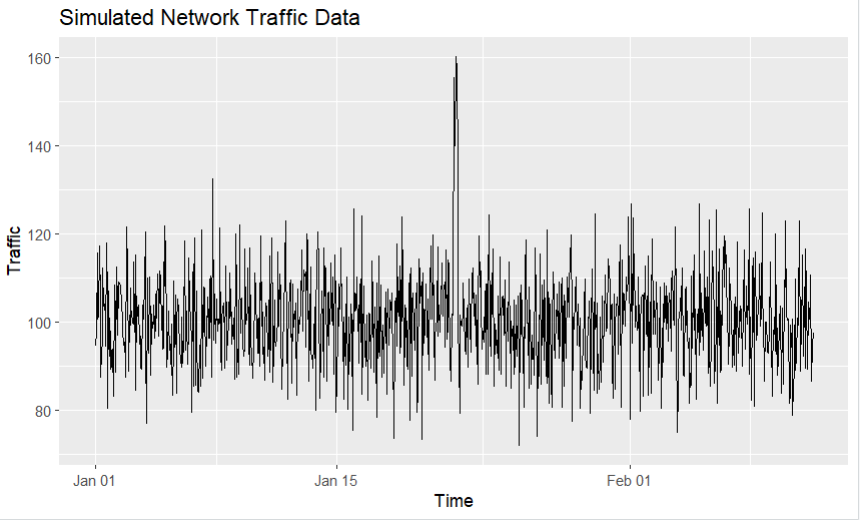
# Print the anomalies

anomalies <- data\_anomalized %>% filter(anomaly == 'Yes')

print(anomalies)

**Output:**





**Conclusion:**

The script achieves its objectives by creating a functional Snake Game with basic features. It implements game initialization, user input handling, collision detection, scoring, and a game over sequence. The player can control the snake, eat food to grow longer, and the game provides a simple yet engaging interactive experience.

# 

# **Github Link:**