# DDOS Project EDA

Firass Elhouat

2025-03-14

## Contents

# 1 Introduction

In today's digital landscape, Distributed Denial-of-Service (DDoS) attacks are among the most disruptive cybersecurity threats, impacting organizations across industries, including government agencies, educational institutions, healthcare providers, and businesses that rely on server and network access. These attacks overwhelm a target's network infrastructure with illegitimate traffic, rendering online services inaccessible to legitimate users and other stakeholders. The financial impact of a DDoS attack varies, but according to an insight report published in 2023 by Zayn Group, an attack can last on average around 68 minutes, totalling around $408,000 per attack, (Zayo Group (2024)). This figure accounts for lost revenue, cost of detection, recovery, and cybersecurity preventives.

Beyond the immediate disruptions, the long-term consequences of DDoS attacks can be severe. These include brand reputation damage, operational delay, legal ramifications, customer churn, and expensive mitigation efforts. A notable example of such cases occurred in 2011 when Sony's PlayStation (PSN) was hit by a large-scale DDoS Attack, which led to nearly a month of downtime service, (Garcia, D.M. (2021)). The attack not only caused financial losses but also damaged Sony's reputation, causing many users to migrate to competitors like Steam and Microsoft Xbox Live. This incident underscored how a DDoS attack can erode customer trust and market share, leaving lasting consequences for businesses.

# 2 Dataset Overview

To better understand and model DDoS attacks, we will use the CIC-DDoS2019 dataset that was generated using a CICFlowMeter-V3 which contains 225,745 rows, and 85 various variables that help distinguish between normal and attack-related traffic. Key variables include packet details, communication protocols, TCP flag indicators, and an attack indicator. A detailed description of these variables can be found in Appendix 1.1, and a summary of the dataset is presented in table 1.

By applying an exploratory data analysis (EDA), our goal is to identify patterns that differentiate normal traffic from malicious DDoS activity and explore the distribution of various categorical variables in the dataset. Through this analysis, we aim to develop a model that can provide inference DDoS attacks, helping us fully understand how these attacks occur, and what are some possible mitigation efforts that can be implemented to reduce the significant effects, and how each variable in our dataset explains our target variable.

# 3 Data Preprocessing & Preparation

In this section, we focus primarily on preparing the dataset for analysis and modeling. Data processing is often considered the most crucial stage, as the quality of data directly impacts the accuracy of our model, reliability of the results, reducing overfitting, and mitigation of any unnecessary errors.

## 3.1 Data cleaning

In the data processing stage, the first task was to re-code the label class, which is our response variable. Initially, the labels were denoted as DDoS and BENIGN, it was re-code to 1 and 0, respectively, to make them more suitable for modeling. Next, 13 of the categorical variables were converted to factor, to ensure efficient usage during the analysis and modeling stage. Through the data cleaning, it was essential to identify potential null values, and any errors that need to be isolated and handled, as often these can negatively affect our analysis. In this case, the dataset did not have any null values, however a number of variables did contain a number of errors.

For instance, flow_packets_s and flow_bytes_s, contained 85 of invalid rows, requiring us to drop them, then correctly converting the variables from char to numeric. Furthermore, these identified columns [fwd_avg_bytes_bulk, fwd_avg_packets_bulk, fwd_avg_bulk_rate, bwd_avg_bytes_bulk, bwd_avg_bulk_rate, and bwd_avg_packets_bulk] only contained zeros or zero variability, requiring the need to drop them as well. Furthermore, two pairs of variables were found to be identical, suggesting a potential issue with the data collection process. Specifically, the pair avg_fwd_segment_size and fwd_packet_length_mean, as well as the pair bwd_packet_length_mean and avg_bwd_segment_size. To address this redundancy and improve the dataset quality, the duplicate variables, avg_fwd_segment_size and avg_fwd_segment_size, were removed.

### 3.2 Handling class imbalance

In this section, we primarily explore the number of categorical variables in our dataset and examine how the distribution varies. Looking at our response variable, Label, we can see that it is fairly balanced, with the following distribution illustrated in fig 1 and table 2. In the CIC-DDoS2019 dataset, several categorical variables exhibit significant imbalance and incompleteness, which can skew model results, and lead to suboptimal performance. One such variable is the protocol variable, which represents different network protocols used. As the protocols used when the label is "BENIGN", is fairly distributed through protocol "0", "6" and "17", however, the the label is "DDoS", the distribution is significantly underrepresented, or in this case entirely absent in protocols "0" and "17", this is illustrated in figure 2 and table 3.

This pattern of imbalance is also observed in several other categorical variables, as illustrated in Figure 3, which shows the distribution of flag counts, and figure 4, which displays the distribution of the PSH/URG flags in FWD/BWD. Due to this imbalance and the presence of incomplete data, we must drop most of the variables shown in these figures 3 and 4. The only exceptions are PSH Flag Count and ACK Flag Count, as they are the only categorical variables that remain fairly balanced and complete.

Furthermore, based on figure 3 both PSH and ACK flag counts show a notable association when flagged, with distinguishing between BENIGN and DDoS traffic. These flags are more frequently observed in DDoS traffic, with PSH flags appearing in 58,185 instances compared to 21,088 in BENIGN traffic. Similarly, ACK flags occur in 70,006 instance of DDoS traffic, compared to 43,845 in BENIGN traffic.

To formaly test for a significant association, I conducted a Pearson's Chi-squared. The results indicate a strong association between PSH flag count and label: $(\chi^2 = 13843,\ df = 1,\ \text{p-value} = 2.2e-16)$ , as well as between ACK flag count and label: $(\chi^2 = 2127.4,\ df = 1,\ \text{p-value} = 2.2e-16)$.

# 4 Descriptive Statistics

In the summary statistics printed below, I've highlighted a few variables that raised concerns due to their extreme and invalid values. Specifically, flow_duration, flow_bytes_s, and flow_packets_s contained invalid values such as -1 and -2e+06, which are likely due to the data collection process. Additionally, init_win_bytes_backward and init_win_bytes_forward exhibited -1 values in approximately 50% of the rows. Since it was unclear whether these values should be converted to 0 or +1, and given their limited variability, I decided to drop these two variables from further analysis. Furthermore, much of the numeric variables displayed extreme right-skewness, which can obscure meaningful patterns. To address this, I've removed the invalid values and applied a log transformation to normalize the distributions. This adjustment provided a clearer visualization of traffic patterns, as seen in the box-plots in figure 5.

We can observe how DDoS traffic tends to have significantly longer flow duration in comparison to benign traffic, this is a distinguishing factor, as DDoS flows persist longer on average, with a higher median value. Furthermore, the median for the flow bytes per second is noticeably higher for DDoS traffic, reflecting the heavy bandwidth consumption typical for high volume attacks. Interestingly, flow packets per second shows a different trend, in comparison to flow bytes per second. To which the median value is actually lower for DDoS traffic than for benign traffic. However, this does not mean that DDoS attacks always have low packet rates, as the presence of numerous outliers suggest that certain attacks involve bursts of packets, which is often a characteristic of packet-flooding techniques.

```
##
##  Summary Statistics of Flow Duration
## ------------------------------------------------
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
##        -1     71238   1453164  16244095   8806652 119999937

##
##  Summary Statistics of Flow Bytes per Second
## ------------------------------------------------
```

```
##       Min.    1st Qu.     Median       Mean    3rd Qu.       Max.
##  -12000000         12       1133     585394      21580 2070000000
```

```
##
##  Summary Statistics of Flow Packets Per Second
## ------------------------------------------------
##       Min.    1st Qu.     Median       Mean    3rd Qu.       Max.
## -2000000.0        0.6        5.2    14241.2       70.4  3000000.0
```

```
##
##  Summary Statistics of Foward Packets Per Second
## ------------------------------------------------
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
##       0.0       0.4       2.3   12617.0      32.9 3000000.0
```

```
##
##  Summary Statistics of Backward Packets Per Second
## ------------------------------------------------
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
##       0.0       0.0       1.5    1641.9      21.0 2000000.0
```

# 5 Correlation Analysis

During an exploratory data analysis, it's essential to thoroughly investigate the variables within the dataset. As the data. The CIC-DDoS2019 dataset, gathered using CICFlowMeter-V3, includes a broad array of features, many of which are statistical summaries captured as individual variables. For example, features such as the mean, maximum, minimum, and standard deviation of packet lengths are recorded as separate variables, providing a detailed overview of the network flow characteristics. While these variables offer valuable insights, they can also introduce redundancy or multicollinearity, which may impact our model performance and lead to overfitting.

To highlight this, Figure 6 illustrates a correlation matrix heat map for a subset of 30 variables, providing valuable insights into the relationships between variables and highlights areas of concern, such as near-perfect correlations that suggest redundancy. This subset was chosen to make the correlation matrix more digestible while emphasizing key variables, while the remaining 29 variables will also be analyzed to identify any that may need to be dropped or transformed to improve model robustness.

For instance, flow_duration is strongly correlated with both flow_iat_max (0.92) and flow_iat_total (0.99), as further demonstrated in Figures 7 and 8. This insight underscores the need for careful variable selection to enhance model accuracy and avoid potential multicollinearity.

# 6 Key Findings

At this stage, we have identified 25 variables that will not be included in the remaining analysis, these exclusions are due to factors such as extreme imbalance, incomplete data, and/or containing only zero values. Additionally, identifier columns have been removed as they do not contribute to the remaining analysis. While in our descriptive statistics and correlation analysis, we had highlighted the need for careful assessment in selecting variables, and how much of them require transformation effects to address the extreme skewness in their distribution. As cleaning and transforming these variables have been particularly useful in distinguishing key differences between DDoS and BENIGN traffic, revealing patterns that were previously obscured. For future work, I will further assess these variables through the modeling and selection phase, ensuring that only the most informative features are retained.

Table 1: CIC-DDoS2019 Dataset Structure and Class Distribution

| | |
|---|---|
| Total number of records: | 225,745 |
| Total number of Variables (columns): | 85 |
| Total number of Continuous variables: | 65 |
| Total number of Categorical variables: | 13 |
| Instances of BENIGN (class 0): | 97,718 |
| Instances of DDoS (class 1): | 128,027 |

Table 2: Distribution of Attack Label

| Label | Instances |
|---|---|
| 0 | 97686 |
| 1 | 128025 |

Table 3: Distribution of Label and Protocol

| Label | Protocol: 0 | Protocol: 6 | Protocol: 17 |
|---|---|---|---|
| 0 | 54 | 64761 | 32871 |
| 1 | 0 | 128025 | 0 |

## 7 Data Visualization Figures & Tables

Figure 1: Distribution of DDoS Attack Labels

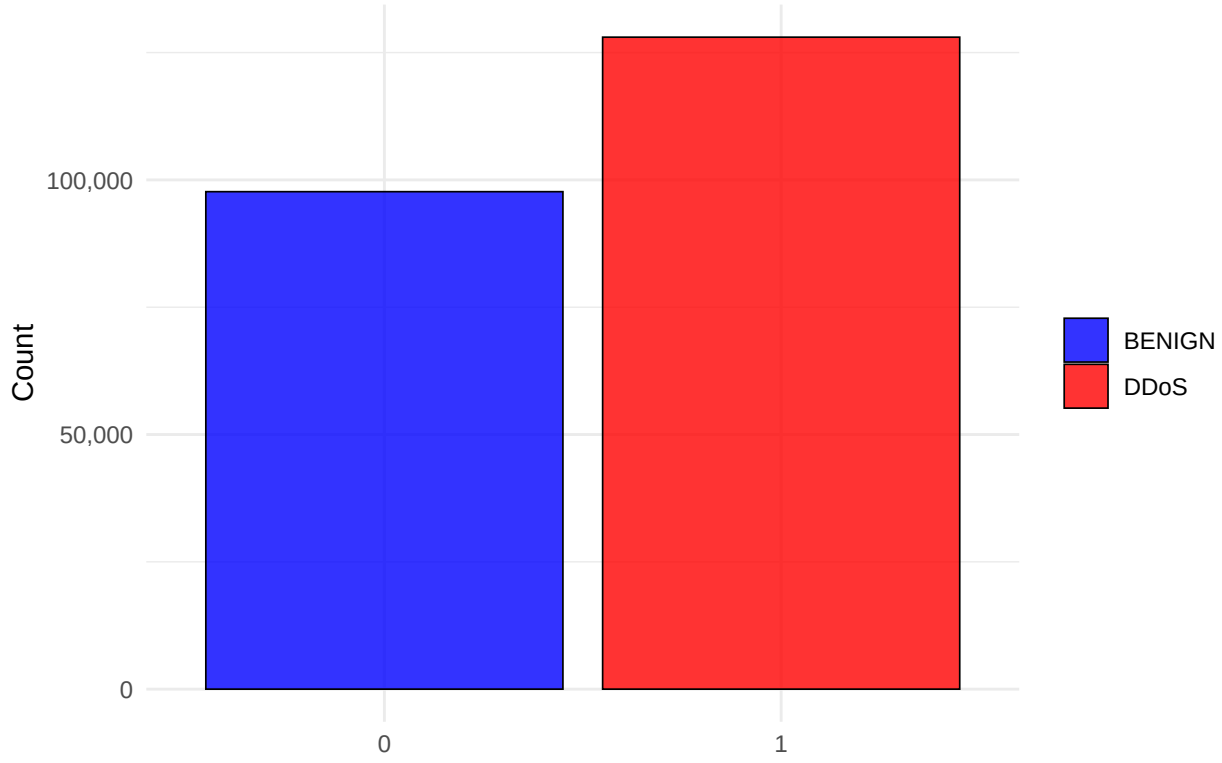


Figure 2: Distribution of DDoS Attack Labels by Protocol

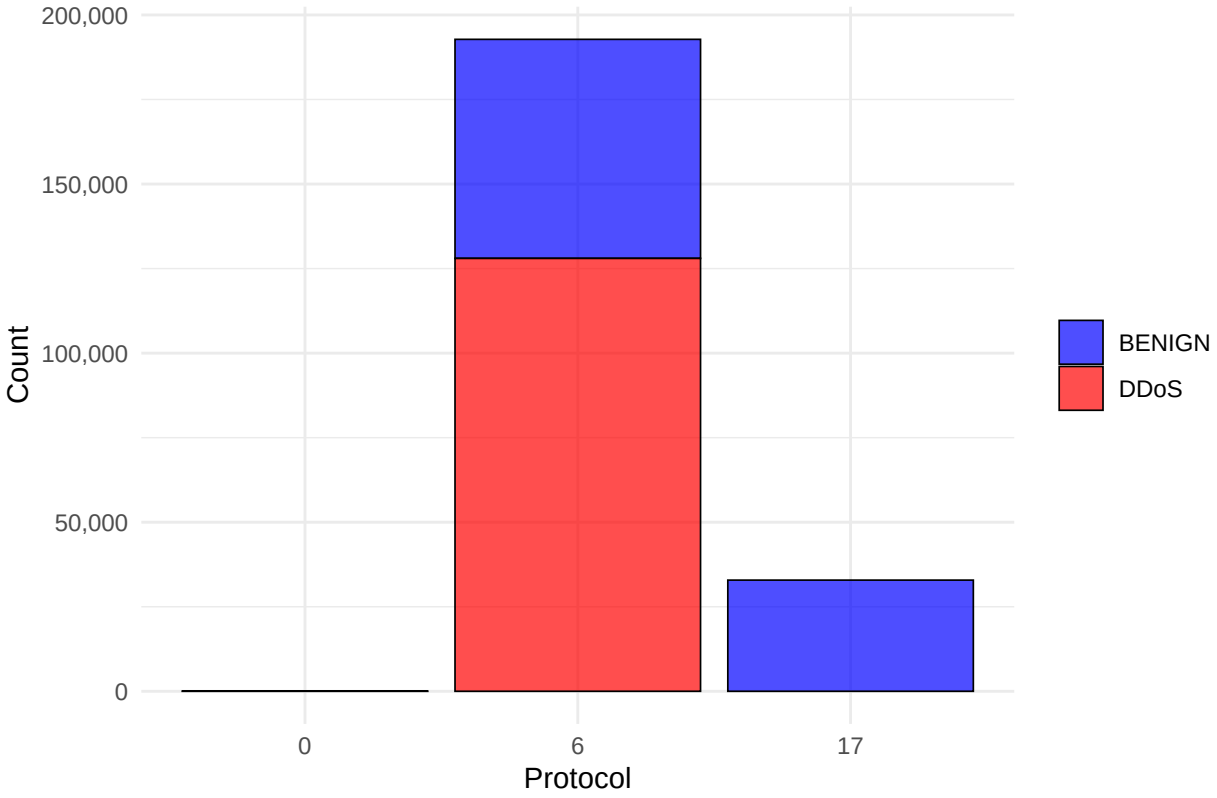


**Figure 3: Distribution of Flag Counts By Label**

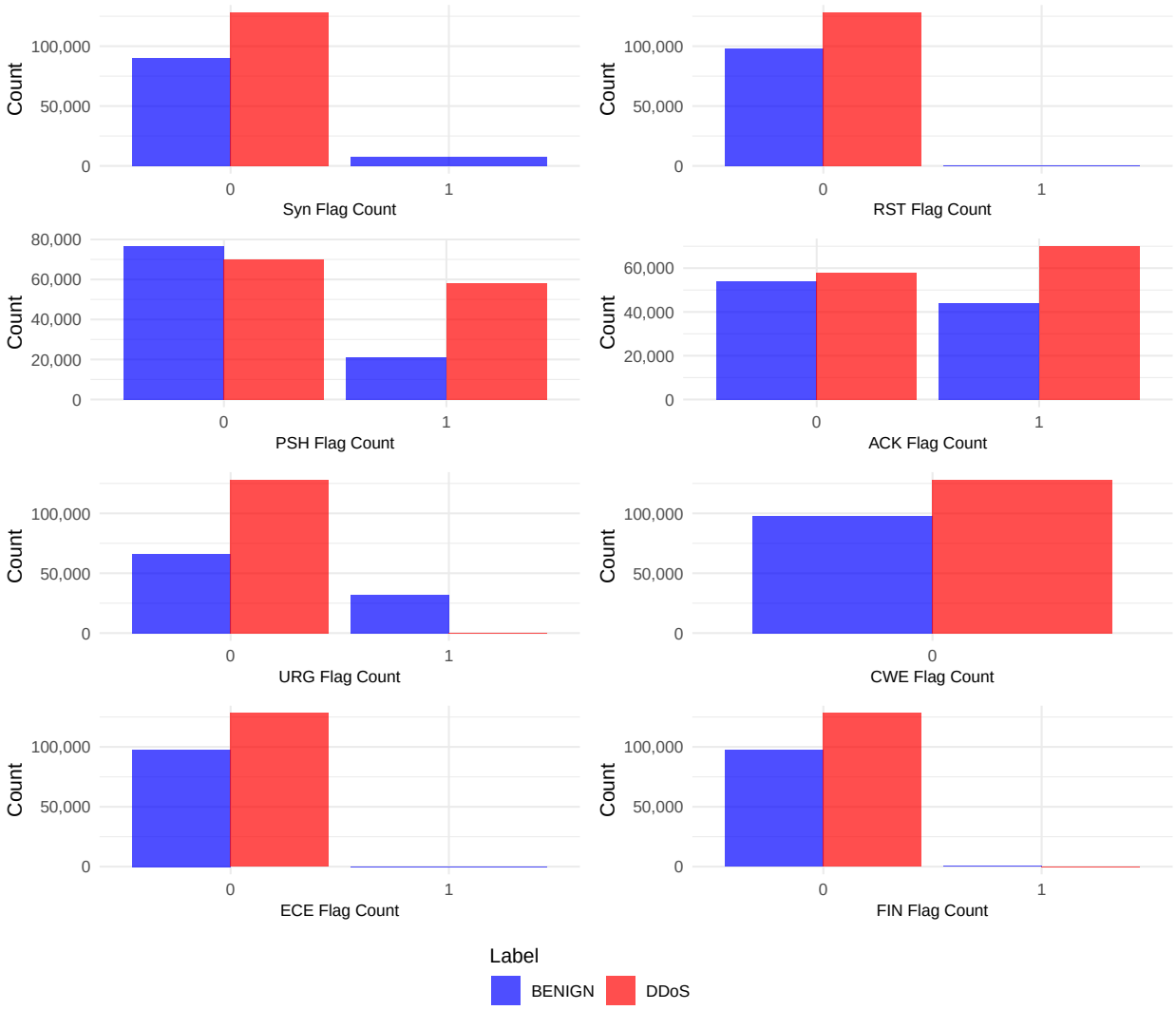


**Figure 4: Distribution of PSH/URG Flags in the FWD & BWD**

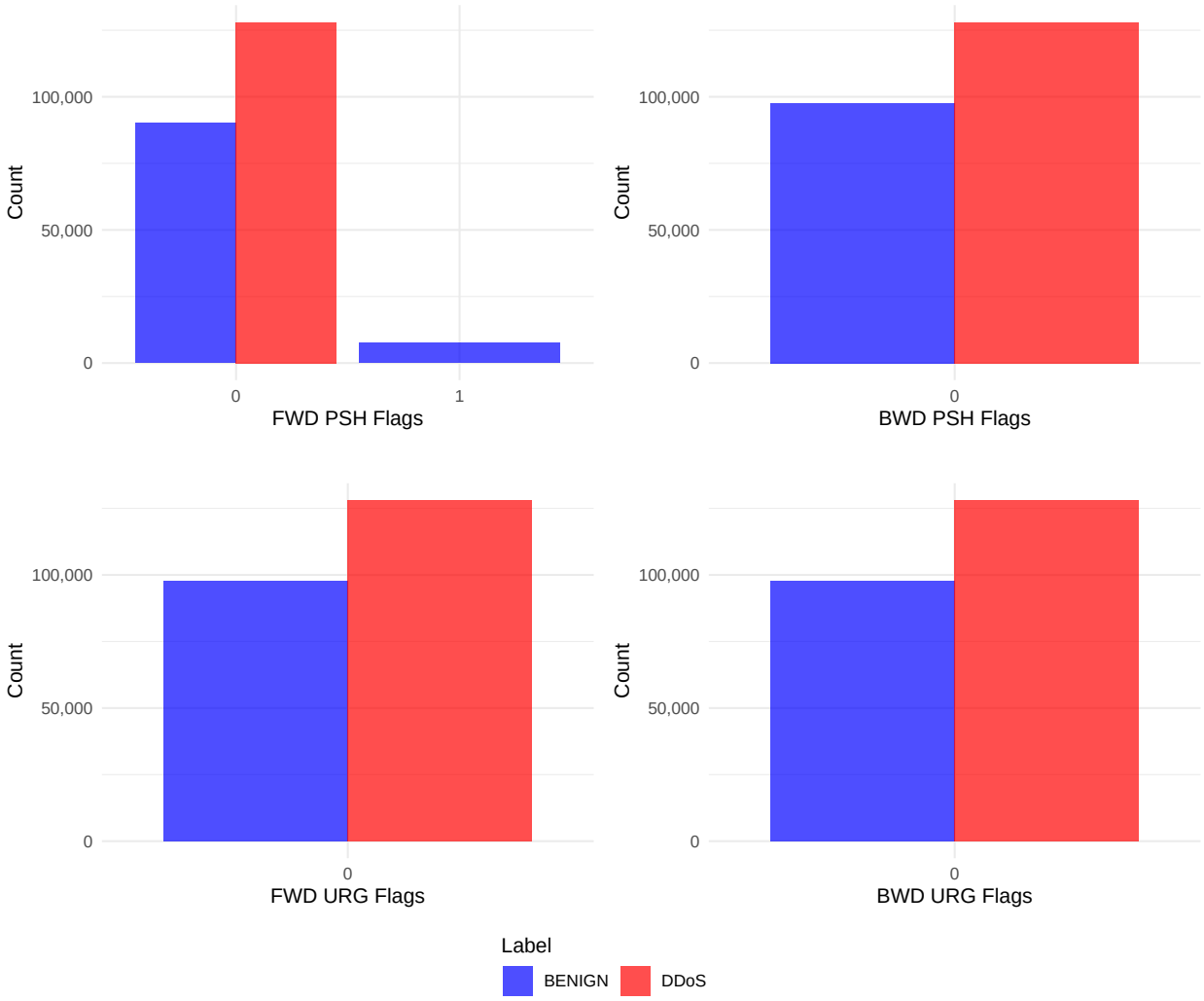


**Figure 5: Boxplots of Selected Variables After Transformation**

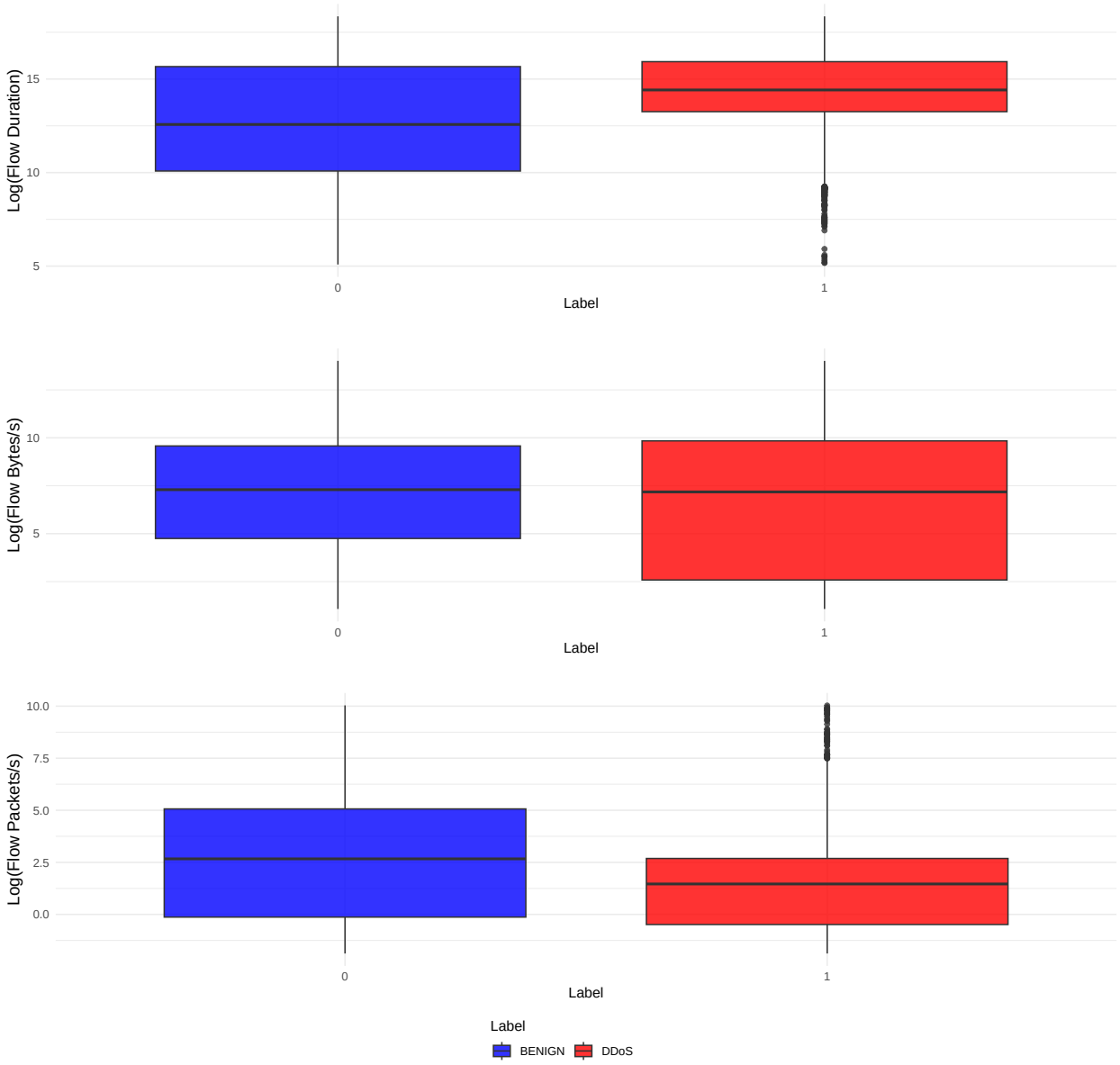


Table 4: Printed Head Portion of High Correlation Variables (|Correlation| >= 0.9)

| row | variable | correlation |
|---|---|---|
| flow_iat_max | flow_duration | 0.9202541 |
| fwd_iat_total | flow_duration | 0.9970539 |
| fwd_iat_max | flow_duration | 0.9179429 |
| total_backward_packets | total_fwd_packets | 0.9567136 |
| total_length_of_bwd_packets | total_fwd_packets | 0.9384698 |
| total_fwd_packets | total_backward_packets | 0.9567136 |

Figure 6: Correlation Matrix Heat Map Of 30 Variables

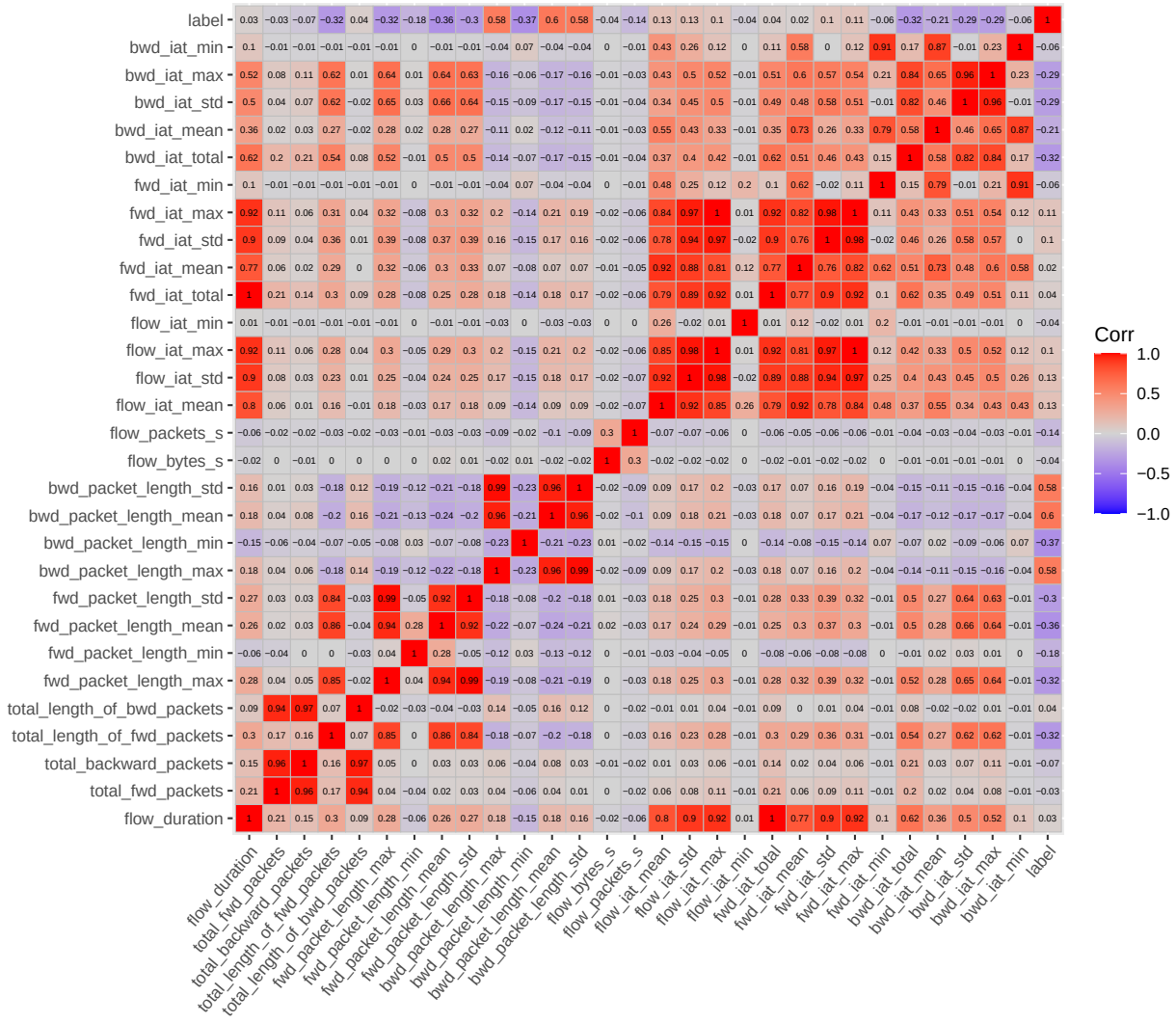


Figure 7: Scatter Plot of Flow duration and Flow Iat Max by label

Correlation: 0.92

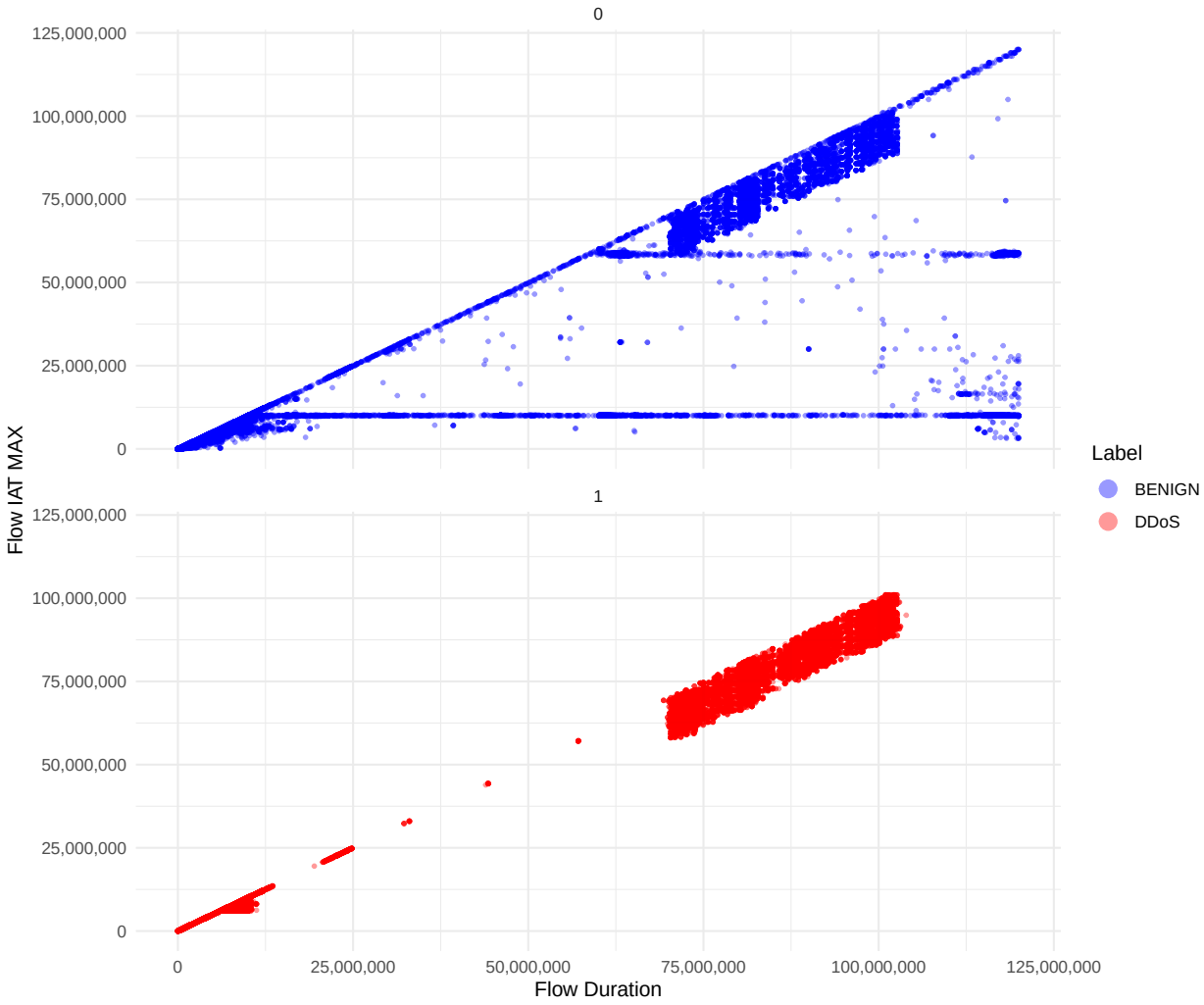
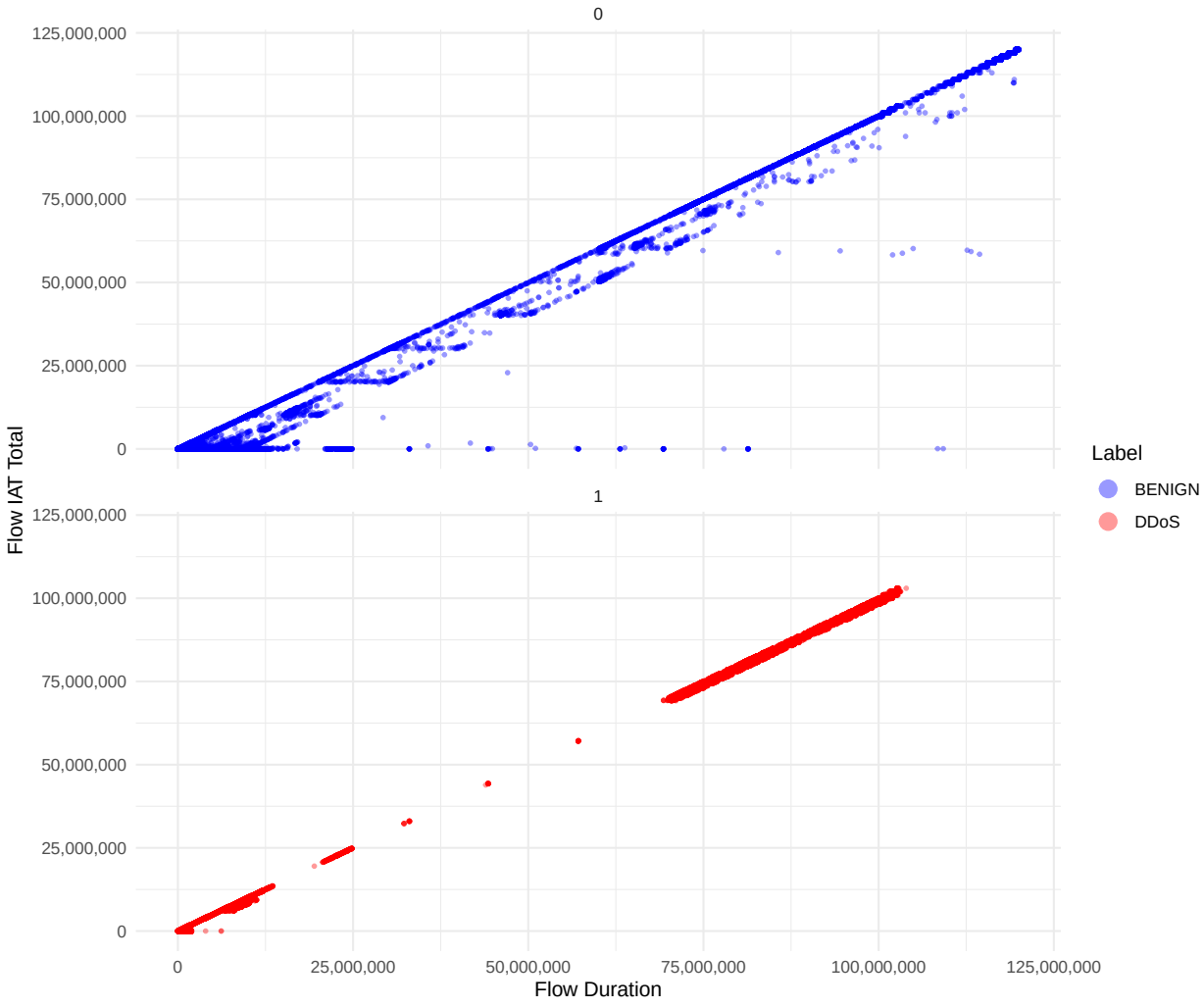


Figure 8: Scatter Plot of Flow duration and Flow Iat Total by label
Correlation: 0.99




# 8 Reference list


- Zayo Group (2024) *Average ddos attack cost businesses nearly half a million dollars in 2023, according to New Zayo Data: Press release: Zayo*, *Zayo.com*. Available at: https://www.zayo.com/newsroom/average-ddos-attack-cost-businesses-nearly-half-a-million-dollars-in-2023-according-to-new-zayo-data/ (Accessed: 18 March 2025).
- Garcia, D.M. (2021) *The 2011 PlayStation Network Hack – what actually happened?*, *WestSide Story*. Available at: https://wsswired.com/4837/entertainment-3/the-2011-playstation-network-hack-what-actually-happened/ (Accessed: 18 March 2025).


# 9 Appendix

## 9.1 Appendix 1.1: CIC-DDoS2019 Dataset Attributes Details

Table 5: Appendix 1.1: CIC-DDoS2019 Dataset Attributes Details

| Variable | Type | Details |
|---|---|---|
| flow_id | Identifier | A unqiue identifer assigned to each flow in the dataset. |
| source_ip | Identifier | The IP address of the device that originates the network traffic. |
| source_port | Identifier | The port number on the source device from which the network traffic is sent. |
| destination_ip | Identifier | The IP address of the device that received the network traffic. |
| destination_port | Identifier | The port number on the destination device that is used to receive the network traffic. |
| timestamp | Identifier | The time stamp at which the network flow occurred. |
| protocol | Categorical | Denotes the protocol used, with three levels; 0, 6 and 17 |
| fin_flag_count | Categorical | The number of packets with FIN (Finish) flag, which signals the termination of a connection, with two levels, 1 and 0. |
| syn_flag_count | Categorical | The number of packets with SYN (Synchronize) flag, used to initiate a TCP connection, with two levels, 1 and 0. |
| rst_flag_count | Categorical | The number of packets with RST (Reset) flag, used to abruptly terminate a connection, with two levels, 1 and 0. |
| psh_flag_count | Categorical | The number of packets with PSH (Push) flag, which tells the receiver to process data immediately rather than buffering it, with two levels, 1 and 0. |
| ack_flag_count | Categorical | The number of packets with ACK (Acknowledgment) flag, which confirms the receipt data, with two levels, 1 and 0. |
| urg_flag_count | Categorical | The number of packets with URG (Urgent) flag, indicating that certain data should be processed immediately, with two levels, 1 and 0. |
| cwe_flag_count | Categorical | Will be renamed to correct name : CWR (Congestion window reduced) flag, used in TCP congestion control to signal a reduced congestion window, with two levels, 1 and 0. |
| ece_flag_count | Categorical | The number of packets with ECE (Explicit Congestion Notification Eco) flag, indicates network congestion, with two levels, 1 and 0. |
| fwd_psh_flags | Categorical | Number of time the PSH (Push) flag was set in packets travelling in the forward direction, with two levels, 1 and 0. |
| bwd_psh_flags | Categorical | Number of time the PSH (Push) flag was set in packets travelling in the backward direction, with two levels, 1 and 0. |
| fwd_urg_flags | Categorical | Number of time the URG (Urgent) flag was set in packets travelling in the forward direction, with two levels, 1 and 0. |
| bwd_urg_flags | Categorical | Number of time the URG (Urgent) flag was set in packets travelling in the forward direction, with two levels, 1 and 0. |
| flow_duration | Continuous | The duration of the flow in microseconds |
| total_fwd_packets | Continuous | The total packets in the forward direction |
| total_backward_packets | Continuous | The total packets in the backward direction |
| total_length_of_fwd_packets | Continuous | The total size of a packet in the forward direction |
| fwd_packet_length_max | Continuous | The total size of a packet in the backward direction |
| fwd_packet_length_min | Continuous | The maximum size of a packet in the forward direction. |
| fwd_packet_length_mean | Continuous | The minimum size of a packet in the forward direction. |
| fwd_packet_length_std | Continuous | The mean size of a packet in the forward direction. |
| bwd_packet_length_max | Continuous | The standard deviation size of a packet in the forward direction. |
| bwd_packet_length_min | Continuous | The maximum size of a packet in the backward direction. |
| bwd_packet_length_min | Continuous | The minimum size of a packet in the backward direction. |
| bwd_packet_length_mean | Continuous | The mean size of a packet in the backward direction. |
| bwd_packet_length_std | Continuous | The standard deviation size of a packet in the backward direction. |
| flow_bytes_s | Continuous | The number of flow bytes per second. |
| flow_packets_s | Continuous | The number of flow packets per second. |
| flow_iat_mean | Continuous | The average inter-arrival time between packets within a flow. |
| flow_iat_std | Continuous | The standard deviation inter-arrival time between packets within a flow. |
| flow_iat_max | Continuous | The maximum inter-arrival time between packets within a flow. |
| flow_iat_min | Continuous | The minimum inter-arrival time between packets within a flow. |
| fwd_iat_total | Continuous | The total inter-arrival time between packets sent in the forward direction. |
| fwd_iat_mean | Continuous | The average inter-arrival time between packets sent in the forward direction. |
| fwd_iat_std | Continuous | The standard deviation inter-arrival time between packets sent in the forward direction. |
| fwd_iat_max | Continuous | The maximum inter-arrival time between packets sent in the forward direction. |
| fwd_iat_min | Continuous | The minimum inter-arrival time between packets sent in the forward direction. |
| bwd_iat_total | Continuous | The total inter-arrival time between packets sent in the forward direction. |
| bwd_iat_mean | Continuous | The average inter-arrival time between packets sent in the forward direction. |
| bwd_iat_std | Continuous | The standard deviation inter-arrival time between packets sent in the forward. direction. |
| bwd_iat_max | Continuous | The maximum inter-arrival time between packets sent in the forward direction. |
| bwd_iat_min | Continuous | The minimum inter-arrival time between packets sent in the forward direction. |
| fwd_header_length_41 | Continuous | The total bytes used for headers in the forward direction. |
| bwd_header_length | Continuous | The total bytes used for headers in the backward direction. |
| fwd_packets_s | Continuous | Number of forward packets transmitted per second. |
| bwd_packets_s | Continuous | Number of backward packets transmitted per second. |
| min_packet_length | Continuous | The minimum length of a packet. |
| max_packet_length | Continuous | The maximum length of a packet. |
| packet_length_mean | Continuous | The average length of a packet. |
| packet_length_std | Continuous | The standard deviation length of a packet. |
| packet_length_variance | Continuous | The variance length of a packet. |
| down_up_ratio | Continuous | Download and upload ratio. |
| average_packet_size | Continuous | The average size of a packet. |
| avg_fwd_segment_size | Continuous | The average size of data segments in the forward direction. |
| avg_bwd_segment_size | Continuous | The average size of data segments in the backward direction. |
| fwd_header_length_62 | Continuous | Length of a packet header in the forward direction. |
| fwd_avg_bytes_bulk | Continuous | Average number of bulk bytes rate in the forward direction. |
| fwd_avg_packets_bulk | Continuous | Average number of bulk packets rate in the forward direction. |
| fwd_avg_bulk_rate | Continuous | Average number of bulk rate in the forward direction. |
| bwd_avg_bytes_bulk | Continuous | Average number of bulk bytes rate in the backward direction. |
| bwd_avg_packets_bulk | Continuous | Average number of bulk packets rate in the backward direction. |
| bwd_avg_bulk_rate | Continuous | Average number of bulk rate in the backward direction. |
| subflow_fwd_packets | Continuous | The average number of packets in a sub flow in the forward direction. |
| subflow_fwd_bytes | Continuous | The average number of bytes in a sub flow in the forward direction. |
| subflow_bwd_packets | Continuous | The average number of packets in a sub flow in the backward direction. |
| subflow_bwd_bytes | Continuous | The average number of bytes in a sub flow in the backward direction. |
| init_win_bytes_forward | Continuous | The total number of bytes sent in initial window in the forward direction. |
| init_win_bytes_backward | Continuous | The total number of bytes sent in initial window in the backward direction. |
| act_data_pkt_fwd | Continuous | Actual number of packets sent in the forward direction. |
| min_seg_size_forward | Continuous | Minimum size of the data segment sent in the forward direction. |
| active_mean | Continuous | The average time a flow was active before becoming idle. |
| active_std | Continuous | The standard deviation time a flow was active before becoming idle. |
| active_max | Continuous | The maximum time a flow was active before becoming idle. |
| active_min | Continuous | The minimum time a flow was active before becoming idle. |
| idle_mean | Continuous | The average time a flow was idle before becoming active. |
| idle_std | Continuous | The standard deviation time a flow was idle before becoming active. |
| idle_max | Continuous | The maximum time a flow was idle before becoming active. |
| idle_min | Continuous | The minimum time a flow was idle before becoming active. |
| label | Categorical(Target) | BENIGN: Normal legitimate network traffic \| DDoS: Malicious network traffic |

## 9.2 Appendix 1.2: R-Code

```r
# --- loading libraries
library(tidyverse) # for extra functions
library(janitor) # for column name cleaning
library(viridis) # for color palettes
library(ggcorrplot)# for correlation plot
library(ggnewscale) # for extra plot functions
library(car) # for VIF function
library(scales) # for extra plot functions
library(gridExtra) # extra plot functions (grid functions)
library(grid) # extra plot functions (grids)
library(kableExtra) # for tables fuctions
```

```r
# ---- loading data
DDoS_dataset <- read_csv("Data/Friday-WorkingHours-Afternoon-DDos.pcap_ISCX.csv") %>% clean
```

```r
# recode label variable to 1 and 0.
DDoS_dataset <- DDoS_dataset %>%
  mutate(label = ifelse(label %in% c("DDoS", "BENIGN"),
                        dplyr::recode(label, "DDoS" = 1, "BENIGN" = 0),
                        label))

# converting several variables to factor
DDoS_dataset <- DDoS_dataset %>%
  mutate(
    label = as.factor(label),
    protocol = as.factor(protocol),
    fwd_psh_flags = as.factor(fwd_psh_flags),
    bwd_psh_flags = as.factor(bwd_psh_flags),
    fwd_urg_flags = as.factor(fwd_urg_flags),
    bwd_urg_flags = as.factor(bwd_urg_flags),
    fin_flag_count = as.factor(fin_flag_count),
    syn_flag_count = as.factor(syn_flag_count),
    rst_flag_count = as.factor(rst_flag_count),
    psh_flag_count = as.factor(psh_flag_count),
    ack_flag_count = as.factor(ack_flag_count),
    urg_flag_count = as.factor(urg_flag_count),
    cwe_flag_count = as.factor(cwe_flag_count),
    ece_flag_count = as.factor(ece_flag_count),
    flow_packets_s = as.numeric(flow_packets_s),
    flow_bytes_s = as.numeric(flow_bytes_s)
  )

DDoS_dataset <- DDoS_dataset %>%
  filter(is.finite(as.numeric(as.character(flow_packets_s))) &
         is.finite(as.numeric(as.character(flow_bytes_s))))

# --- check for null values
null_value <- sum(is.na(DDoS_dataset))
```

```r
# contingency table for lable and psh flag count
psh_flag <- table(DDoS_dataset$psh_flag_count, DDoS_dataset$label)

# Perform Chi-Squared test
chisq.test(psh_flag)

# contingency table for lable and ack flag count
ack_flag <- table(DDoS_dataset$ack_flag_count, DDoS_dataset$label)

# Perform Chi-Squared test
chisq.test(psh_flag)
```

```r
# Define the plots for syn_flag_count
plot1 <- ggplot(DDoS_dataset, aes(x = as.factor(syn_flag_count), fill = as.factor(label))) +
  geom_bar(stat = "count", position = "dodge", alpha = 0.7) +
  labs(x = "Syn Flag Count", y = "Count", fill = "Label") +
  theme_minimal() +
  scale_fill_manual(values = c("blue", "red")) +
  scale_y_continuous(labels = label_comma()) +
  theme(legend.position = "none", axis.title.x = element_text(size = 9))

# Define the plots for rst_flag_count
plot2 <- ggplot(DDoS_dataset, aes(x = as.factor(rst_flag_count), fill = as.factor(label))) +
  geom_bar(stat = "count", position = "dodge", alpha = 0.7) +
  labs(x = "RST Flag Count", y = "Count", fill = "Label") +
  theme_minimal() +
  scale_fill_manual(values = c("blue", "red")) +
  scale_y_continuous(labels = label_comma()) +
  theme(legend.position = "none", axis.title.x = element_text(size = 9))

# Define the plots for psh_flag_count
plot3 <- ggplot(DDoS_dataset, aes(x = as.factor(psh_flag_count), fill = as.factor(label))) +
  geom_bar(stat = "count", position = "dodge", alpha = 0.7) +
  labs(x = "PSH Flag Count", y = "Count", fill = "Label") +
  theme_minimal() +
  scale_fill_manual(values = c("blue", "red")) +
  scale_y_continuous(labels = label_comma()) +
  theme(legend.position = "none", axis.title.x = element_text(size = 9))

# Define the plots for ack_flag_count
plot4 <- ggplot(DDoS_dataset, aes(x = as.factor(ack_flag_count), fill = as.factor(label))) +
  geom_bar(stat = "count", position = "dodge", alpha = 0.7) +
  labs(x = "ACK Flag Count", y = "Count", fill = "Label") +
  theme_minimal() +
  scale_fill_manual(values = c("blue", "red")) +
  scale_y_continuous(labels = label_comma()) +
  theme(legend.position = "none", axis.title.x = element_text(size = 9))

# Define the plots for urg_flag_count
plot5 <- ggplot(DDoS_dataset, aes(x = as.factor(urg_flag_count), fill = as.factor(label))) +
  geom_bar(stat = "count", position = "dodge", alpha = 0.7) +
  labs(x = "URG Flag Count", y = "Count", fill = "Label") +
  theme_minimal() +
```

```
  scale_fill_manual(values = c("blue", "red")) +
  scale_y_continuous(labels = label_comma()) +
  theme(legend.position = "none", axis.title.x = element_text(size = 9))

# Define the plots for cwe_flag_count
plot6 <- ggplot(DDoS_dataset, aes(x = as.factor(cwe_flag_count), fill = as.factor(label)))
  geom_bar(stat = "count", position = "dodge", alpha = 0.7) +
  labs(x = "CWE Flag Count", y = "Count", fill = "Label") +
  theme_minimal() +
  scale_fill_manual(values = c("blue", "red")) +
  scale_y_continuous(labels = label_comma()) +
  theme(legend.position = "none", axis.title.x = element_text(size = 9))

# Define the plots for ece_flag_count
plot7 <- ggplot(DDoS_dataset, aes(x = as.factor(ece_flag_count), fill = as.factor(label)))
  geom_bar(stat = "count", position = "dodge", alpha = 0.7) +
  labs(x = "ECE Flag Count", y = "Count", fill = "Label") +
  theme_minimal() +
  scale_fill_manual(values = c("blue", "red")) +
  scale_y_continuous(labels = label_comma()) +
  theme(legend.position = "none", axis.title.x = element_text(size = 9))


plot8 <- ggplot(DDoS_dataset, aes(x = as.factor(fin_flag_count), fill = as.factor(label)))
  geom_bar(stat = "count", position = "dodge", alpha = 0.7) +
  labs(x = "FIN Flag Count", y = "Count", fill = "Label") +
  theme_minimal() +
  scale_fill_manual(values = c("blue", "red")) +
  scale_y_continuous(labels = label_comma()) +
  theme(legend.position = "none", axis.title.x = element_text(size = 9))

# Define the plots for dummyplot1 to extract legend
dummyplot1 <- ggplot(DDoS_dataset, aes(x = as.factor(syn_flag_count),
                                       fill = as.factor(label))) +
  geom_bar(stat = "count", position = "dodge", alpha = 0.7) +
  scale_fill_manual(values = c("0" = "blue", "1" = "red"),
                    labels = c("0" = "BENIGN", "1" = "DDoS"),
                    guide = guide_legend(title = "Label", direction = "horizontal")) +
  theme_minimal() +
  guides(fill = guide_legend(title = "Label",
                             title.position = "top",
                             title.hjust = 0.5)) +
  theme(legend.position = "bottom")


# function to extract the legend
g_legend <- function(a.gplot) {
  tmp <- ggplot_gtable(ggplot_build(a.gplot))
  leg <- which(sapply(tmp$grobs, function(x) x$name) == "guide-box")
  tmp$grobs[[leg]]
}
legend <- g_legend(dummyplot1)
```

```r
# Helper function to extract the legend
g_legend <- function(a.gplot) {
  tmp <- ggplot_gtable(ggplot_build(a.gplot))
  leg <- which(sapply(tmp$grobs, function(x) x$name) == "guide-box")
  tmp$grobs[[leg]]
}

# Define the plots for fwd_psh_flags
plot_1 <- ggplot(DDoS_dataset, aes(x = fwd_psh_flags, fill = as.factor(label))) +
  geom_bar(stat = "count", position = "dodge", alpha = 0.7) +
  labs(x = "FWD PSH Flags", y = "Count", fill = "Label") +
  scale_fill_manual(values = c("blue", "red")) +
  theme_minimal() +
  ggtitle("") +
  scale_y_continuous(labels = label_comma()) +
  theme(legend.position = "none")  # Remove the legend from this plot

# Define the plots for bwd_psh_flags
plot_2 <- ggplot(DDoS_dataset, aes(x = bwd_psh_flags, fill = as.factor(label))) +
  geom_bar(stat = "count", position = "dodge", alpha = 0.7) +
  labs(x = "BWD PSH Flags", y = "Count", fill = "Label") +
  scale_fill_manual(values = c("blue", "red")) +
  theme_minimal() +
  ggtitle("") +
  scale_y_continuous(labels = label_comma()) +
  theme(legend.position = "none")

# Define the plots for fwd_urg_flags
plot_3 <- ggplot(DDoS_dataset, aes(x = fwd_urg_flags, fill = as.factor(label))) +
  geom_bar(stat = "count", position = "dodge", alpha = 0.7) +
  labs(x = "FWD URG Flags", y = "Count", fill = "Label") +
  scale_fill_manual(values = c("blue", "red")) +
  theme_minimal() +
  ggtitle("") +
  scale_y_continuous(labels = label_comma()) +
  theme(legend.position = "none")

# Define the plots for bwd_urg_flags
plot_4 <- ggplot(DDoS_dataset, aes(x = bwd_urg_flags, fill = as.factor(label))) +
  geom_bar(stat = "count", position = "dodge", alpha = 0.7) +
  labs(x = "BWD URG Flags", y = "Count", fill = "Label") +
  scale_fill_manual(values = c("blue", "red")) +
  theme_minimal() +
  ggtitle("") +
  scale_y_continuous(labels = label_comma()) +
  theme(legend.position = "none")

# Define the plots for dummyplot2
dummyplot2 <- ggplot(DDoS_dataset, aes(x = fwd_psh_flags, fill = as.factor(label))) +
  geom_bar(stat = "count", position = "dodge", alpha = 0.7) +
  scale_fill_manual(values = c("0" = "blue", "1" = "red"),
                    labels = c("0" = "BENIGN", "1" = "DDoS"),
                    guide = guide_legend(title = NULL)) +
```

```
  theme_minimal() +
  guides(fill = guide_legend(title = "Label", ncol = 2))

# Extract the legend from the dummy plot
legend <- g_legend(dummyplot2)
```

```
# function to check zero variability due to constant only zeros
check_zero_columns <- function(data) {
  zero_cols <- sapply(data, function(col) all(col == 0, na.rm = TRUE))
  return(names(data)[zero_cols])
}

# define the columns from the function
zero_columns <- check_zero_columns(DDoS_dataset)

# Print columns that contain only zeros
if (length(zero_columns) > 0) {
  print(paste("Columns with only zeros:", paste(zero_columns, collapse = ", ")))
} else {
  print("No columns contain only zeros.")
}
```

```
# dropping due to imbalance and zero variability issues
# 17 total protocol, fin_flag_count, syn_flag_count urg_flag_count
# ece_flag_count, rst_flag_count, cwe_flag_count, fwd_psh_flags, bwd_psh_flags
# fwd_urg_flags, bwd_urg_flags, fwd_avg_bytes_bulk, fwd_avg_packets_bulk
# fwd_avg_bulk_rate, bwd_avg_bytes_bulk, bwd_avg_bulk_rate, bwd_avg_packets_bulk

# --- remove unwanted variables
DDoS_dataset0 <- DDoS_dataset %>%
  select(-flow_id, -source_ip,
         -source_port, -destination_ip,
         -destination_port, -timestamp,
         -protocol, -syn_flag_count,
         -fin_flag_count, -urg_flag_count,
         -ece_flag_count, -rst_flag_count,
         -cwe_flag_count, -fwd_psh_flags,
         -bwd_psh_flags, -fwd_urg_flags,
         -bwd_urg_flags, -fwd_avg_bytes_bulk,
         -fwd_avg_packets_bulk, -fwd_avg_bulk_rate,
         -bwd_avg_bytes_bulk, -bwd_avg_bulk_rate,
         -bwd_avg_packets_bulk, -init_win_bytes_forward,
         -init_win_bytes_backward)
```

```
print_summary <- function(data, var_name) {
  cat("\n Summary Statistics of", var_name, "\n")
  cat("-----------------------------------------------\n")
  summary_output <- capture.output(summary(data))
  cat(paste(summary_output, collapse = "\n"), "\n")
}

# Print summaries for selected variables
print_summary(DDoS_dataset0$flow_duration, "Flow Duration")
```

```r
print_summary(DDoS_dataset0$flow_bytes_s, "Flow Bytes per Second")
print_summary(DDoS_dataset0$flow_packets_s, "Flow Packets Per Second")
print_summary(DDoS_dataset0$fwd_packets_s, "Foward Packets Per Second")
print_summary(DDoS_dataset0$bwd_packets_s, "Backward Packets Per Second")
```

```r
# Boxplot for flow_duration after log transformation
box1 <- DDoS_dataset0 %>%
  filter(flow_duration != -1) %>%
  mutate(
    Q5 = quantile(flow_duration, 0.05, na.rm = TRUE),
    Q95 = quantile(flow_duration, 0.95, na.rm = TRUE)
  ) %>%
  filter(flow_duration >= Q5 & flow_duration <= Q95) %>%
  ggplot(aes(x = factor(label), y = log(flow_duration), fill = factor(label))) +
  geom_boxplot(alpha = 0.8) +
  labs(title = "",
       x = "Label",
       y = "Log(Flow Duration)") +
  scale_fill_manual(values = c("0" = "blue", "1" = "red"),
                    labels = c("0" = "BENIGN", "1" = "DDoS")) +
  theme_minimal() + theme(legend.position = "none",
        axis.title.y = element_text(size = 13))

# Boxplot for flow_bytes_s after log transformation
box2 <- DDoS_dataset0 %>%
  filter(flow_bytes_s > 0) %>%
  mutate(
    Q5 = quantile(flow_bytes_s, 0.05, na.rm = TRUE),
    Q95 = quantile(flow_bytes_s, 0.95, na.rm = TRUE)
  ) %>%
  filter(flow_bytes_s >= Q5 & flow_bytes_s <= Q95) %>%
  ggplot(aes(x = factor(label), y = log(flow_bytes_s), fill = factor(label))) +
  geom_boxplot(alpha = 0.8) +
  labs(title = "",
       x = "Label",
       y = "Log(Flow Bytes/s)") +
  scale_fill_manual(values = c("0" = "blue", "1" = "red"),
                    labels = c("0" = "BENIGN", "1" = "DDoS")) +
  theme_minimal() + theme(legend.position = "none",
        axis.title.y = element_text(size = 13))

# Boxplot for flow_packets_s after log transformation
box3 <- DDoS_dataset0 %>%
  filter(flow_packets_s > 0) %>%
  mutate(
    Q5 = quantile(flow_packets_s, 0.05, na.rm = TRUE),
    Q95 = quantile(flow_packets_s, 0.95, na.rm = TRUE)
  ) %>%
  filter(flow_packets_s >= Q5 & flow_packets_s <= Q95) %>%
  ggplot(aes(x = factor(label), y = log(flow_packets_s), fill = factor(label))) +
  geom_boxplot(alpha = 0.8) +
  labs(title = "",
       x = "Label",
```

```
       y = "Log(Flow Packets/s)") +
  scale_fill_manual(values = c("0" = "blue", "1" = "red"),
                    labels = c("0" = "BENIGN", "1" = "DDoS")) +
  theme_minimal() + theme(legend.position = "none",
        axis.title.y = element_text(size = 13))

dummyplot0 <- DDoS_dataset0 %>%
  filter(flow_packets_s > 0) %>%
  mutate(
    Q5 = quantile(flow_packets_s, 0.05, na.rm = TRUE),
    Q95 = quantile(flow_packets_s, 0.95, na.rm = TRUE)
  ) %>%
  filter(flow_packets_s >= Q5 & flow_packets_s <= Q95) %>%
  ggplot(aes(x = factor(label), y = log(flow_packets_s), fill = factor(label))) +
  geom_boxplot(alpha = 0.8) +
  labs(title = "",
       x = "Label",
       y = "Log(flow_packets_s)") +
  scale_fill_manual(values = c("0" = "blue", "1" = "red"),
                    labels = c("0" = "BENIGN", "1" = "DDoS"),
                    guide = guide_legend(title = "Label",
                                         title.position = "top",
                                         direction = "horizontal")) +
  theme_minimal()

legend0 <- g_legend(dummyplot0)
```

```
# Select only numeric variables + label
cor_matrix <- DDoS_dataset0 %>%
  mutate(label = as.numeric(as.factor(label)) - 1) %>%
  select_if(is.numeric) %>%
  select(1:29, label) %>%
  cor()

# Select full numeric variables + label
cor_matrixFull <- DDoS_dataset0 %>%
  mutate(label = as.numeric(as.factor(label)) - 1) %>%
  select_if(is.numeric) %>%
  cor()
```

```
# highlighting high correlated variables from the 30 variables
high_corr <- cor_matrix %>%
  as.data.frame() %>%
  mutate(row = rownames(cor_matrix)) %>%
  gather(key = "variable", value = "correlation", -row) %>%
  filter(abs(correlation) >= 0.9 & correlation != 1)

# highlighting high correlated variables from the full dataset
high_corrFull <- cor_matrixFull %>%
  as.data.frame() %>%
  mutate(row = rownames(cor_matrixFull)) %>%
  gather(key = "variable", value = "correlation", -row) %>%
  filter(abs(correlation) >= 0.9 & correlation != 1)
```

```r
# Defining summary table containing data summary
summary_table <- data.frame(
  c("Total number of records:",
    "Total number of Variables (columns):",
    "Total number of Continuous variables:",
    "Total number of Categorical variables:",
    "Instances of BENIGN (class 0):",
    "Instances of DDoS (class 1):"),
  c("225,745","85","65","13","97,718","128,027")
)

# print as a table using latex
kable(summary_table, format = "latex", booktabs = TRUE, col.names = NULL,
      caption = "CIC-DDoS2019 Dataset Structure and Class Distribution \\label{tab:summary
      ) %>%
  kable_styling(latex_options = c("striped", "scale_down")) %>%
  kable_styling(position = "center")
```

```r
# Table of label distribution
kable(table(DDoS_dataset$label), format = "latex",
      caption = "Distribution of Attack Label \\label{tab:label_distribution}",
      col.names = c("Label","Instances"), booktabs = TRUE) %>%
  kable_styling(latex_options = c("striped", "scale_down"), position = "center")
```

```r
# Table of label and protocol
kable(table(DDoS_dataset$label, DDoS_dataset$protocol), format = "latex",
      caption = "Distribution of Label and Protocol \\label{tab:label_protocol}",
      col.names = c("Label","Protocol: 0","Protocol: 6","Protocol: 17"),
      booktabs = TRUE) %>%
  kable_styling(latex_options = c("striped", "scale_down"), position = "center")
```

```r
# Print top portion of high correlation table
head(high_corr) %>%
  kable(format = "latex",
        caption = "Printed Head Portion of High Correlation Variables (|Correlation| >= 0.9
        booktabs = TRUE) %>%
  kable_styling(latex_options = c("striped", "scale_down"), position = "center")
```

```r
# bar plot of target distribution
ggplot(DDoS_dataset, aes(x = as.factor(label))) +
  geom_bar(aes(fill = as.factor(label)), color = "black", alpha = 0.8, linewidth = 0.3) +
  labs(title = "Figure 1:  Distribution of DDoS Attack Labels", x = "", y = "Count") +
  theme_minimal() +
  scale_fill_manual(values = c("0" = "blue", "1" = "red"),
                    labels = c("0" = "BENIGN", "1" = "DDoS"),
                    guide = guide_legend(title = NULL)) +
  scale_y_continuous(labels = scales::comma)
```

```r
# bar plot of target and protocol distribution
ggplot(DDoS_dataset, aes(x = as.factor(protocol), fill = as.factor(label))) +
  geom_bar(color = "black", alpha = 0.7, linewidth = 0.3) +
  labs(title = "Figure 2: Distribution of DDoS Attack Labels by Protocol", x = "Protocol",
```

```
  theme_minimal() +
  scale_fill_manual(values = c("0" = "blue", "1" = "red"),
                    labels = c("0" = "BENIGN", "1" = "DDoS"),
                    guide = guide_legend(title = NULL)) +
  scale_y_continuous(labels = scales::comma)

# Issues protocol variable
# When "0", BENIGN is 54, while DDoS is 0.
# When "6", BENIGN is 64793, while DDoS is 128027.
# When "17", BENIGN is 32871, while DDoS is 0.
```

```
# Using grid.arrange for plots for flag counts
grid.arrange(
  arrangeGrob(plot1, plot2, plot3, plot4, plot5, plot6, plot7, plot8, ncol = 2),
  legend,
  ncol = 1,
  heights = c(10, 1),
  top = textGrob("Figure 3: Distribution of Flag Counts By Label", gp = gpar(fontsize = 10,
)
```

```
# Using grid.arrange for plotsgrid.arrange for flags
grid.arrange(
  arrangeGrob(plot_1, plot_2, plot_3, plot_4,
  ncol = 2,
  top = textGrob("Figure 4: Distribution of PSH/URG Flags in the FWD & BWD",
                 gp = gpar(fontsize = 10, fontface = "bold")),
  bottom = legend))
```

```
# Boxplots arranged with dummy legend
grid.arrange(
  box1, box2, box3,
  ncol = 1,
  top = textGrob("Figure 5: Boxplots of Selected Variables After Transformation",
                 gp = gpar(fontsize = 15, fontface = "bold")),
  bottom = legend0 # Adjust the height of the legend to make it smaller
)
```

```
# Using ggcorrplot to plot correlation matrix of subset 30 variables
ggcorrplot(cor_matrix, method = "square",
           type = "full",
           lab = TRUE,
           lab_size = 1.8,
           tl.srt = 50,
           tl.cex = 9,
           title = "Figure 6: Correlation Matrix Heat Map Of 30 Variables",
           ggtheme = theme_gray(),
           colors = c("blue", "lightgrey", "red"))
```

```
# scatter plot of flow_duration and flow_iat_max facet wrapped by label
ggplot(DDoS_dataset, aes(x = flow_duration, y = flow_iat_max, color = factor(label))) +
  geom_point(alpha=0.4, size =0.7) +  # Line plot
  labs(title = "Figure 7:  Scatter Plot of Flow duration and Flow Iat Max by label",
```

```
       subtitle = "Correlation: 0.92",
       x = "Flow Duration",
       y = "Flow IAT MAX") +
  scale_color_manual(
    values = c("0" = "blue", "1" = "red"),
    labels = c("0" = "BENIGN", "1" = "DDoS"),
    name = "Label") +
  guides(color = guide_legend(override.aes = list(size = 4))) +
  facet_wrap(~ label, nrow = length(unique(DDoS_dataset$label))) +
  scale_x_continuous(labels = scales::comma) +
  scale_y_continuous(labels = scales::comma) +
  theme_minimal()
```

```
# scatter plot of flow_duration and fwd_iat_total facet wrapped by label
ggplot(DDoS_dataset, aes(x = flow_duration, y = fwd_iat_total, color = factor(label))) +
  geom_point(alpha=0.4, size =0.7) +  # Line plot
  labs(title = "Figure 8:  Scatter Plot of Flow duration and Flow Iat Total by label",
       subtitle = "Correlation: 0.99",
       x = "Flow Duration",
       y = "Flow IAT Total") +
  scale_color_manual(
    values = c("0" = "blue", "1" = "red"),
    labels = c("0" = "BENIGN", "1" = "DDoS"),
    name = "Label") +
  guides(color = guide_legend(override.aes = list(size = 4))) +
  facet_wrap(~ label, nrow = length(unique(DDoS_dataset$label))) +
  scale_x_continuous(labels = scales::comma) +
  scale_y_continuous(labels = scales::comma) +
  theme_minimal()
```