

To establish baseline for threat detection

DISSERTATION

Submitted in partial fulfillment of the requirements of the

Degree : MTech in Data Science and Engineering

By

Goyal Taruchit Tarun Chitra
2022DC04496

Under the supervision of

Prathibha Panduranga Rao
Vice President

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE
Pilani (Rajasthan) INDIA

(January, 2025)

BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI
FIRST SEMESTER 2024-25

DSECLZG628T DISSERTATION

Dissertation Title : To establish baseline for threat detection

Name of Supervisor : Prathibha Panduranga Rao

Name of Student : Goyal Taruchit Tarun

Chitra ID No. of Student : 2022DC04496

Courses Relevant for the Project & Corresponding Semester:

1. Introduction to Data Science (Semester 1)
2. Machine Learning (Semester 2)
3. Artificial and Computational Intelligence (Semester 2)
4. Data Visualization and Interpretation (Semester 2)

Abstract

Cybersecurity is key for any organization; attackers keep evolving and learning new ways to evade cyber-attack detection deployed by organizations. By analyzing the events, security operations center (SOC) can detect threats and make existing detections more effective. While analyzing network dataset for detecting cyber-attacks, the volume of records and dimensionality of records generated are very high. As the result, building automated analysis and detection of potential threats can lead to noisy outcomes. In most of the historical research, the power of advanced graph or deep learning models are leveraged for handling high-dimensional dataset. But it comes at the cost of extensive tuning, computation power and time. Thus, the dissertation aims to leverage optimization algorithms for feature selection which enables to handle high dimensional dataset efficiently and effectively, allowing to identify the most optimal set of features for training the models, improving model's overall performance. Most often in network dataset, the features are not linearly correlated, thus, for handling non-linearity of features, optimization algorithms are useful. The features are then used to train two models: the first model performs binary classification to differentiate an attack from a normal event. The second model performs a multi-class classification to identify the type of attack. This enables to handle both the models independently and make choices which allow to get optimal results for the specific objectives of each model. Finally, the project evaluates each model based on the corresponding subset of optimal features obtained from each optimization algorithm and rank the outcomes. Thus, the projects demonstrate mitigating dependence on advance and complex models for higher accuracy, and rather use existing optimization algorithms with Machine Learning algorithms to achieve the same. This also allows to define baseline of results using Machine Learning algorithms, which can be later used as a benchmark for more advanced models.

List of Symbols and Abbreviations

Sr No	Abbreviation	Definition
1	PSO	Particle Swarm Optimization
2	ABC	Artificial Bee Colony
3	AIS	Artificial Immune System
4	SOC	Security Operations Centre
5	IDD	Inadvertent Data Disclosure
6	IOCs	Indicators of Compromise
7	IP	Internet Protocol
8	UEBA	User Entity and Behaviour Analytics
9	SSN	Social Security Number
10	PII	Personally Identifiable Information
11	CIC	Canadian Institute for Cybersecurity
12	p(i, lb) t	Local best of particle i and time t
13	x(i)t	Position of particle i at time t
14	v(i)t	Velocity of particle i at time t
15	gBest	Global best of the swarm
16	pBest	Personal best of the particle
17	T	Number of generations
18	f	Output of objective function
19	fit	Output of fitness function

List of Tables

Table no	Title of table
1	Summarizing analysis of all datasets
2	Comparison of all datasets
3	List of fields in UBR-IDD dataset
4	List of fields in CIC dataset
5	Count and percentage of outliers in each feature
6	Encoded values of ClassLabel
7	Encoded values of ClassLabel after dropping the rows
8	Distribution of records in sampled dataset based on isMalicious
9	Distribution of records in sampled dataset based on attack_id
10	Interpretation of Cohen's Kappa values

List of Figures

Figure number	Figure title
1	Distribution of events in UNR-IDD dataset based on Binary Label
2	Distribution of events in UNR-IDD dataset based on Label
3	Distribution of events in CIC dataset based on Label
4	Distribution of events in CIC dataset based on ClassLabel
5	List of histograms for all independent features plotted using log scale
6	Histogram of Flow duration plotted on log scale
7	Histogram of Total Fwd Packets plotted on log scale
8	Histogram of Total Backward Packets plotted on log scale
9	Histogram of Forward Packets Length Total plotted on log scale
10	Histogram of Bwd Packets Length Total plotted on log scale
11	Histogram of Fwd Packet Length Max plotted on log scale
12	Histogram of Fwd Packet Length Mean plotted on log scale
13	Histogram of Fwd Packet Length Std plotted on log scale
14	Histogram of Bwd Packet Length Max plotted on log scale
15	Histogram of Bwd Packet Length Mean plotted on log scale
16	Histogram of Bwd Packet Length std plotted on log scale
17	Histogram of Flow Bytes/s plotted on log scale
18	Histogram of Flow Packets/s plotted on log scale
19	Histogram of Flow IAT Mean plotted on log scale
20	Histogram of Flow IAT Std plotted on log scale
21	Histogram of Flow IAT Max plotted on log scale
22	Histogram of Flow IAT Min plotted on log scale
23	Histogram of Flow IAT Total plotted on log scale
24	Histogram of Fwd IAT Mean plotted on log scale
25	Histogram of Fwd IAT Std plotted on log scale
26	Histogram of Fwd IAT Max plotted on log scale
27	Histogram of Fwd IAT Min plotted on log scale
28	Histogram of Bwd IAT Total plotted on log scale
29	Histogram of Bwd IAT Mean plotted on log scale
30	Histogram of Bwd IAT Std plotted on log scale
31	Histogram of Bwd IAT Max plotted on log scale
32	Histogram of Bwd IAT Min plotted on log scale
33	Histogram of Fwd PSH Flags plotted on log scale
34	Histogram of Fwd Header Length plotted on log scale
35	Histogram of Bwd Header Length plotted on log scale
36	Histogram of Fwd Packets/s plotted on log scale
37	Histogram of Bwd Packets/s plotted on log scale
38	Histogram of Packet Length Max plotted on log scale
39	Histogram of Packet Length Mean plotted on log scale
40	Histogram of Packet Length Std plotted on log scale

41	Histogram of Packet Length Variance plotted on log scale
42	Histogram of SYN Flag Count plotted on log scale
43	Histogram of URG Flag Count plotted on log scale
44	Histogram of Avg Packet Size plotted on log scale
45	Histogram of Avg Fwd Segment Size plotted on log scale
46	Histogram of Avg Bwd Segment Size plotted on log scale
47	Histogram of Subflow Fwd Packets plotted on log scale
48	Histogram of Subflow Fwd Bytes plotted on log scale
49	Histogram of Subflow Bwd Packets plotted on log scale
50	Histogram of Subflow Bwd Bytes plotted on log scale
51	Histogram of Init Fwd Win Bytes plotted on log scale
52	Histogram of Init Bwd Win Bytes plotted on log scale
53	Histogram of Fwd Act Data Packets plotted on log scale
54	Histogram of Fwd Seg Size Min plotted on log scale
55	Histogram of Active Mean plotted on log scale
56	Histogram of Active Std plotted on log scale
57	Histogram of Active Max plotted on log scale
58	Histogram of Active Min plotted on log scale
59	Histogram of Idle Mean plotted on log scale
60	Histogram of Idle Std plotted on log scale
61	Histogram of Idle Max plotted on log scale
62	Histogram of Idle Min plotted on log scale
63	Bar chart of ClassLabel plotted on log scale
64	Bar chart of Label plotted on log scale
65	Histogram to compare impact of winsorization on Init Fwd Win Bytes
66	Histogram to compare impact of winsorization on Init Bwd Win Bytes
67	Histogram to compare impact of winsorization on Fwd Seg Size Min
68	Histogram to compare impact of winsorization on Bwd IAT Mean
69	Histogram to compare impact of Robust Scaling on Init Fwd Win Bytes
70	Histogram to compare impact of Robust Scaling on Init Bwd Win Bytes
71	Histogram to compare impact of Robust Scaling on Fwd Seg Size Min
72	Histogram to compare impact of Robust Scaling on Bwd IAT Mean
73	Histogram of Flow Duration plotted on log scale after handling negative values and outliers
74	Histogram of Total Fwd Packets plotted on log scale after handling negative values and outliers
75	Histogram of Total Backward Packets plotted on log scale after handling negative values and outliers
76	Histogram of Fwd Packets Length Total plotted on log scale after handling negative values and outliers
77	Histogram of Bwd Packets Length Total plotted on log scale after handling negative values and outliers
78	Histogram of Fwd Packet Length Max plotted on log scale after handling negative values and outliers
79	Histogram of Fwd Packet Length Mean plotted on log scale after handling negative values and outliers

80	Histogram of Fwd Packet Length Std plotted on log scale after handling negative values and outliers
81	Histogram of Bwd Packet Length Max plotted on log scale after handling negative values and outliers
82	Histogram of Bwd Packet Length Mean plotted on log scale after handling negative values and outliers
83	Histogram of Bwd Packet Length Std plotted on log scale after handling negative values and outliers
84	Histogram of Flow Bytes/s plotted on log scale after handling negative values and outliers
85	Histogram of Flow Packets/s plotted on log scale after handling negative values and outliers
86	Histogram of Flow IAT Mean plotted on log scale after handling negative values and outliers
87	Histogram of Flow IAT Std plotted on log scale after handling negative values and outliers
88	Histogram of Flow IAT Max plotted on log scale after handling negative values and outliers
89	Histogram of Flow IAT Min plotted on log scale after handling negative values and outliers
90	Histogram of Flow IAT Total plotted on log scale after handling negative values and outliers
91	Histogram of Fwd IAT Mean plotted on log scale after handling negative values and outliers
92	Histogram of Fwd IAT Std plotted on log scale after handling negative values and outliers
93	Histogram of Fwd IAT Max plotted on log scale after handling negative values and outliers
94	Histogram of Fwd IAT Min plotted on log scale after handling negative values and outliers
95	Histogram of Bwd IAT Total plotted on log scale after handling negative values and outliers
96	Histogram of Bwd IAT Mean plotted on log scale after handling negative values and outliers
97	Histogram of Bwd IAT Std plotted on log scale after handling negative values and outliers
98	Histogram of Bwd IAT Max plotted on log scale after handling negative values and outliers
99	Histogram of Bwd IAT Min plotted on log scale after handling negative values and outliers
100	Histogram of Fwd PSH Flags plotted on log scale after handling negative values and outliers
101	Histogram of Fwd Header Length plotted on log scale after handling negative values and outliers
102	Histogram of Bwd Header Length plotted on log scale after handling negative values and outliers
103	Histogram of Fwd Packets/s plotted on log scale after handling negative values and outliers
104	Histogram of Bwd Packets/s plotted on log scale after handling negative values and outliers

105	Histogram of Packet Length Max plotted on log scale after handling negative values and outliers
106	Histogram of Packet Length Mean plotted on log scale after handling negative values and outliers
107	Histogram of Packet Length Std plotted on log scale after handling negative values and outliers
108	Histogram of Packet Length Variance plotted on log scale after handling negative values and outliers
109	Histogram of SYN Flag Count plotted on log scale after handling negative values and outliers
110	Histogram of URG Flag Count plotted on log scale after handling negative values and outliers
111	Histogram of Avg Packet Size plotted on log scale after handling negative values and outliers
112	Histogram of Avg Fwd Segment Size plotted on log scale after handling negative values and outliers
113	Histogram of Avg Bwd Segment Size plotted on log scale after handling negative values and outliers
114	Histogram of Subflow Fwd Packets plotted on log scale after handling negative values and outliers
115	Histogram of Subflow Fwd Bytes plotted on log scale after handling negative values and outliers
116	Histogram of Subflow Bwd Packets plotted on log scale after handling negative values and outliers
117	Histogram of Subflow Bwd Bytes plotted on log scale after handling negative values and outliers
118	Histogram of Init Fwd Win Bytes plotted on log scale after handling negative values and outliers
119	Histogram of Init Bwd Win Bytes plotted on log scale after handling negative values and outliers
120	Histogram of Fwd Act Data Packets plotted on log scale after handling negative values and outliers
121	Histogram of Fwd Seg Size Min plotted on log scale after handling negative values and outliers
122	Histogram of Active Mean plotted on log scale after handling negative values and outliers
123	Histogram of Active Std plotted on log scale after handling negative values and outliers
124	Histogram of Active Max plotted on log scale after handling negative values and outliers
125	Histogram of Active Min plotted on log scale after handling negative values and outliers
126	Histogram of Idle Mean plotted on log scale after handling negative values and outliers
127	Histogram of Idle Std plotted on log scale after handling negative values and outliers
128	Histogram of Idle Max plotted on log scale after handling negative values and outliers
129	Histogram of Idle Min plotted on log scale after handling negative values and outliers

130	Pyramid chart of Flow Duration w.r.t isMalicious
131	Pyramid chart of Total Fwd Packets w.r.t isMalicious
132	Pyramid chart of Total Backward Packets w.r.t isMalicious
133	Pyramid chart of Fwd Packets Length Total w.r.t isMalicious
134	Pyramid chart of Bwd Packets Length Total w.r.t isMalicious
135	Pyramid chart of Fwd Packet Length Max w.r.t isMalicious
136	Pyramid chart of Fwd Packet Length Mean w.r.t isMalicious
137	Pyramid chart of Fwd Packet Length Std w.r.t isMalicious
138	Pyramid chart of Bwd Packet Length Max w.r.t isMalicious
139	Pyramid chart of Bwd Packet Length Mean w.r.t isMalicious
140	Pyramid chart of Bwd Packet Length Std w.r.t isMalicious
141	Pyramid chart of Flow Bytes/s w.r.t isMalicious
142	Pyramid chart of Flow Packets/s w.r.t isMalicious
143	Pyramid chart of Flow IAT Mean w.r.t isMalicious
144	Pyramid chart of Flow IAT Std w.r.t isMalicious
145	Pyramid chart of Flow IAT Max w.r.t isMalicious
146	Pyramid chart of Flow IAT Min w.r.t isMalicious
147	Pyramid chart of Fwd IAT Total w.r.t isMalicious
148	Pyramid chart of Fwd IAT Mean w.r.t isMalicious
149	Pyramid chart of Fwd IAT Std w.r.t isMalicious
150	Pyramid chart of Fwd IAT Max w.r.t isMalicious
151	Pyramid chart of Fwd IAT Min w.r.t isMalicious
152	Pyramid chart of Bwd IAT Total w.r.t isMalicious
153	Pyramid chart of Bwd IAT Mean w.r.t isMalicious
154	Pyramid chart of Bwd IAT Std w.r.t isMalicious
155	Pyramid chart of Bwd IAT Max w.r.t isMalicious
156	Pyramid chart of Bwd IAT Min w.r.t isMalicious
157	Pyramid chart of Fwd Header Length w.r.t isMalicious
158	Pyramid chart of Bwd Header Length w.r.t isMalicious
159	Pyramid chart of Fwd Packets/s Length w.r.t isMalicious
160	Pyramid chart of Bwd Packets/s Length w.r.t isMalicious
161	Pyramid chart of Packet Length Max w.r.t isMalicious
162	Pyramid chart of Packet Length Mean w.r.t isMalicious
163	Pyramid chart of Packet Length Std w.r.t isMalicious
164	Pyramid chart of Packet Length Variance w.r.t isMalicious
165	Pyramid chart of Avg Packet Size w.r.t isMalicious
166	Pyramid chart of Avg Fwd Segment Size w.r.t isMalicious
167	Pyramid chart of Avg Bwd Segment Size w.r.t isMalicious
168	Pyramid chart of Subflow Fwd Packets w.r.t isMalicious
169	Pyramid chart of Subflow Fwd Bytes w.r.t isMalicious
170	Pyramid chart of Subflow Bwd Packets w.r.t isMalicious
171	Pyramid chart of Subflow Bwd Bytes w.r.t isMalicious
172	Pyramid chart of Init Bwd Win Bytes w.r.t isMalicious
173	Pyramid chart of Fwd Act Data Packets w.r.t isMalicious

174	Correlation matrix based on sub-sampled dataset
175	Stacked bar chart for Bwd Packets Length Total w.r.t isMalicious
176	Stacked bar chart for Fwd Packet Length Std w.r.t isMalicious
177	Stacked bar chart for Bwd Packet Length Max w.r.t isMalicious
178	Stacked bar chart for Bwd Packet Length Mean w.r.t isMalicious
179	Stacked bar chart for Bwd Packet Length Std w.r.t isMalicious
180	Stacked bar chart for Flow IAT Std w.r.t isMalicious
181	Stacked bar chart for Fwd IAT Std w.r.t isMalicious
182	Stacked bar chart for Bwd IAT Total w.r.t isMalicious
183	Stacked bar chart for Bwd IAT Mean w.r.t isMalicious
184	Stacked bar chart for Bwd IAT Std w.r.t isMalicious
185	Stacked bar chart for Bwd IAT Max w.r.t isMalicious
186	Stacked bar chart for Bwd IAT Min w.r.t isMalicious
187	Stacked bar chart for Avg Bwd Segment Size w.r.t isMalicious
188	Stacked bar chart for Subflow Bwd Bytes w.r.t isMalicious
189	Stacked bar chart for Fwd Act Data Packets w.r.t isMalicious
190	Stacked bar chart for Init Fwd Win Bytes w.r.t isMalicious
191	Stacked bar chart for Fwd Seg Size Min w.r.t isMalicious
192	Flowchart for PSO algorithm

Table of contents

Sr No	Title	Page number
1	Chapter 1: List of objectives	1
2	Chapter 2: Research about cybersecurity use cases which align with work of current employer	2 – 4
3	Chapter 3: Searching and analysing open-source datasets for our cybersecurity use case	5 – 13
4	Chapter 4: Data preprocessing, analysis, visualization and feature engineering	14 – 114
5	Chapter 5: Research about optimization algorithms for feature selection	115 – 120
6	Chapter 6: Research about different classification algorithms	121 – 122
7	Chapter 7: Evaluation metrics for classification models	123 – 131
8	Directions for future work after mid semester	132 – 133
9	Bibliography/ References	134

Chapter 1

Following is the list of objectives: -

1. Research about cybersecurity use cases which align with work of current employer
2. Searching and analysing open-source datasets for our cybersecurity use case
3. Data preprocessing, analysis, visualization and feature engineering
4. Research about optimization algorithms for feature selection
5. Research about different classification algorithms
6. Research about different evaluation metrics used to evaluate results of classifier model
7. Feature selection
8. Training the models
9. Evaluation of models
10. Documentation of results
11. Review and corrections

The objectives that we marked with green were undertaken till date.

Objective 6 was not part of the list of objectives submitted during submission of abstract. It was added later based on the feedback received after viva evaluation of the abstract.

Chapter 2

Research about cybersecurity use cases which align with work of current employer

Following are some of the cybersecurity uses cases used in the organization: -

1. Classification of email as spam and non-spam, phishing and non-phishing.
2. Analysing Indicators of Compromise (IOCs) for intrusion detection system.
3. Identifying potential threats in network traffic.
4. UEBA for anomaly detection
5. Inadvertent Data Disclosure (IDD)

Classification email as spam and non-spam, phishing and non-phishing: -

1. All employees in the organization have an option to report a suspicious email received by them.
2. Once an email is reported, it goes to respective cybersecurity team and parsed against different set of rules.
3. Based on the outcome of parsing, the email is classified as spam or non-spam, phishing or non-phishing.
4. If there are no issues observed, the email is classified as clean.
5. In the background, there is an automated email checker which keeps track of all emails received by employees, and validates if it's an authentic email or a suspicious email.
6. For suspicious emails, it checks more of its meta data for further analysis and actions.
7. Most of the times, it checks for credential harvester attack since its one of the most common cyberattacks observed over email.

Analysing Indicators of Compromise for intrusion detection system

1. Matching and fetching details of IOCs is essential to build detection rules and models for intrusion detection system.
2. Some of the examples of types of IOCs: IP address, Domain name, File hash, Email address, URL.
3. Building IOC scanner helps to quickly detect cyber threats and enable SOC team to get details faster for their usage to handle and resolve the issue in less turnaround time.
4. IOC for Incident Response: When a breach is suspected: -
 - a. A list of all relevant IOCs is made.
 - b. All logs are scanned to check for presence of the list of IOCs.
 - c. The IOCs that match in the logs are determined and their details are fetched such as:
 - - i. First seen
 - ii. Last seen
 - iii. Count
 - iv. Number of distinct users against which it was observed
 - v. Number of distinct hosts against which it was observed
 - d. Based on the data, impacted systems are analysed.
 - e. The timeline and scope of breach is determined.
5. IOC for threat hunting: -

- a. An IOC is searched across all logs.
- b. The list of IOCs is made based on historical threats that are previously observed and documented. Thus, it is built based on Advanced Persistent Threats (APTs).
- c. Among the list of known IOCs, the IOCs that match in logs are analysed by fetching the meta data such as: -
 - i. First seen
 - ii. Last seen
 - iii. Count
 - iv. Number of distinct users against which it was observed
 - v. Number of distinct hosts against which it was observed
- d. As per the requirements, more details are fetched
- e. If the events observed in logs are classified as malicious, then actions such as quarantining and isolation are carried out.
- f. Since logs are generated at run time, fetching all the relevant information and computing statistics can be time consuming.
- g. Thus, in order to improve efficiency, summary tables are built and maintained for each type of IOC, and searched are performed on those summary tables. This helps to fasten the searches and reduce turn-around time while searching on vast volume of data or searching data across long time range.

Identifying potential threats in network traffic

1. In network traffic, there are logs generated based on user activities.
2. But sometimes, we observe logs having spikes at unusual hours. For example, transaction activities carried out 2 am.
3. Thus, such events require macro and micro level monitoring and analysis to identify such events.
4. Many times, most of the activities tracked as unusual are normal events, caused due events like: -
 - a. Some batch job which was executed after resolving its error.
 - b. Activities carried out in business hours of another time-zone.
5. Thus, the occurrence of malicious events are rare, but identifying them is extremely critical.

UEBA for anomaly detection

1. Focus on homogenous population in the organization that have similar and repetitive patterns of behaviour.
2. Conduct historical analysis using true positive security analysis.
3. Creating baseline for each user based on their behaviour and behaviour of other similar users.
4. Identify anomalies using statistics and unsupervised learning.
5. Score anomalies using predetermined weights and combine them to reach the threshold for alerting.
6. Generation of alerts for further analysis.
7. For creating baseline of users, we try to find users who are similar and then form a baseline based on their behaviour pattern.
8. Along with the anomaly, its rank in terms of impact will also be computed and used to reduce the alerts, prioritize the most important issues among all the alerts.

Inadvertent Data Disclosure (IDD)

1. It is used for detecting and preventing misdirected emails.
2. Example: Detecting sensitive data such as SSNs.
3. It also used Titus classification for automated identification of sensitive data in email and in attachments attached in the email.
4. The scanning is also carried out for data that is stored in the system of employees. Thus, if some employee has data which contains PII information of users, it detects the file name and file path and generates email to notify the employee and the manager about and ask to take actions such as moving data out from employee's storage or deleting the files if they are not required.

Chapter 3

Searching and analysing open-source datasets for our cybersecurity use case

Table 1: Summarizing analysis of all datasets: -

Sr No	Dataset name	Number of rows	Number of features	Number of duplicates	Null records: Y/N	Number of target features	Binary or Multi-class classification	Comments
1	BETH	763144	16	0	N	2	Binary	3 files: - training data validation data testing data This table has records of training data.
2	BrakTooth	9002	5	1909	N	1	Multi-class	
3	Mil-STD-1553	23000	52	0	Y	2	Multi-class	7 files, 1 is of benign data and 6 are of attacks This table has records of benign file.
4	ServerLogs	172838	16	0	N	1	Binary	
5	UNR-IDD	37411	34	1	N	2	Binary, Multi-class	
6	cic	9167581	59	310	N	2	Multi-class	The file is in .parquet format.
7	KDD cup 1999	494021	42	348435	N	1	Multi-class	Imported from sklearn preloaded datasets.

Table 2: Comparison of all datasets: -

Sr No	Dataset name	Advantages	Disadvantages
1	BETH	1. Large number of records. 2. No duplicates and no null records.	1. Does not allow to build model for multi-class classification.
2	BrakTooth	1. No null values. 2. Moderate number of records 3. Allows to build model for multi-class classification.	1. 21% records are duplicate.
3	Mil-STD-1553	1. Large number of records with very high dimensionality. 2. Allows to build model for multi-class classification.	1. Large number of null records.
4	ServerLogs	1. Large number of records. 2. No duplicates or null records.	1. Does not allow to build model for multi-class classification.
5	UNR-IDB	1. Large number of records. 2. No duplicates or null records. 3. Allows to build model both binary and multi-class classification.	
6	cic	1. Large number of records. 2. Very high dimensionality. 3. Allow to build model for multi-class classification.	
7	KDD cup 1999	1. Large number of records. 2. Very high dimensionality. 3. Allow to build model for multi-class classification.	1. 70.5% records in the dataset are duplicate.

From the initial analysis, we observed two datasets are most suitable for our project: -

1. UNR-IDD
2. cic

Thus, we further analysed the two datasets to understand their properties.

UNR-IDD dataset: -

Table 3: List of fields in UNR-IDD dataset

Sr No	Field name	Description		Type of field
1	Switch ID	12 switches		High-order Nominal
2	Port Number	4 ports		Low-order Nominal
3	Received Packets	Number of packets received by the port	Port statistics	Scale
4	Received Bytes	Number of bytes received by the port		Scale
5	Sent Bytes	Number of bytes sent		Scale
6	Sent Packets	Number of packets sent by the port		Scale
7	Port alive Duration (S)	The time port has been alive in seconds		Scale
8	Packets Rx Dropped	Number of packets dropped by the receiver		Scale
9	Packets Tx Dropped	Number of packets dropped by the sender		Scale
10	Packets Rx Errors	Number of transmit errors		Scale
11	Packets Tx Errors	Number of receive errors		Scale
12	Delta Received Packets	Number of packets received by the port	Delta port statistics	Scale
13	Delta Received Bytes	Number of bytes received by the port		Scale
14	Delta Sent Bytes	Number of packets sent by the port		Scale
15	Delta Sent Packets	Number of bytes sent		Scale
16	Delta Port alive Duration (S)	The time port has been alive in seconds		Scale
17	Delta Packets Rx Dropped	Number of packets dropped by the receiver		Scale
18	Delta Packets Tx Dropped	Number of packets dropped by the sender		Scale
19	Delta Packets Rx Errors	Number of transmit errors		Scale
20	Delta Packets Tx Errors	Number of receive errors		Scale

21	Connection Point	Network connection point expressed as a pair of the network element identifier and port number.	Flow entry and Flow table	Scale
22	Total Load/Rate	Obtain the current observed total load/rate (in bytes/s) on a link		Scale
23	Total Load/Latest	Obtain the latest total load bytes counter viewed on that link.		Scale
24	Unknown Load/Rate	Obtain the current observed unknown-sized load/rate (in bytes/s) on a link.		Scale
25	Unknown Load/Latest	Obtain the latest unknown-sized load bytes counter viewed on that link.		Scale
26	Latest bytes counter			Scale
27	is_valid	Indicates whether this load was built on valid values.		Binary
28	Table ID	Returns the Table ID values.		Scale
29	Active Flow Entries	Returns the number of active flow entries in this table.		Scale
30	Packets Looked Up	Returns the number of packets looked up in the table.		Scale
31	Packets Matched	Returns the number of packets that successfully matched in the table		Scale
32	Max Size	Returns the maximum size of this table.		Scale
33	Label	Normal: Normal network functionality TCP-SYN: TCP-SYN Flood PortScan: Port Scanning Overflow: Flow Table overflow Blackhole: Blackholde attack Diversion: Traffic diversion attack	Multi-class classification	Target: Multi-class
34	Binary Label	Normal: Normal network functionality	Binary classification	Target: Binary

		Attack: Network intrusion		
--	--	---------------------------	--	--

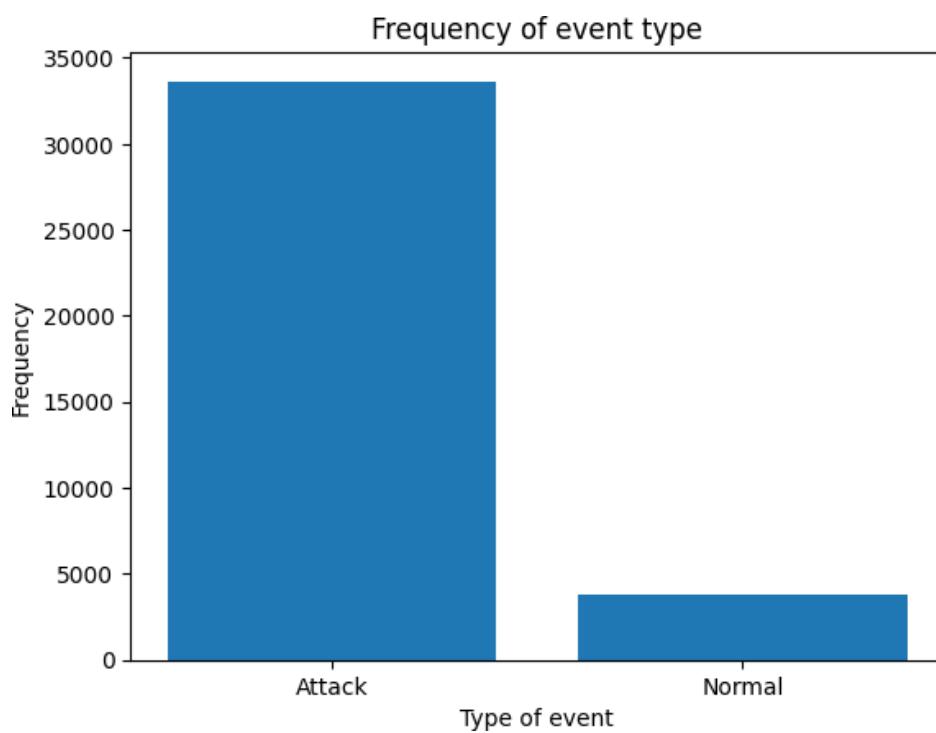


Figure 1: Distribution of events in UNR-IDD dataset based on Binary Label

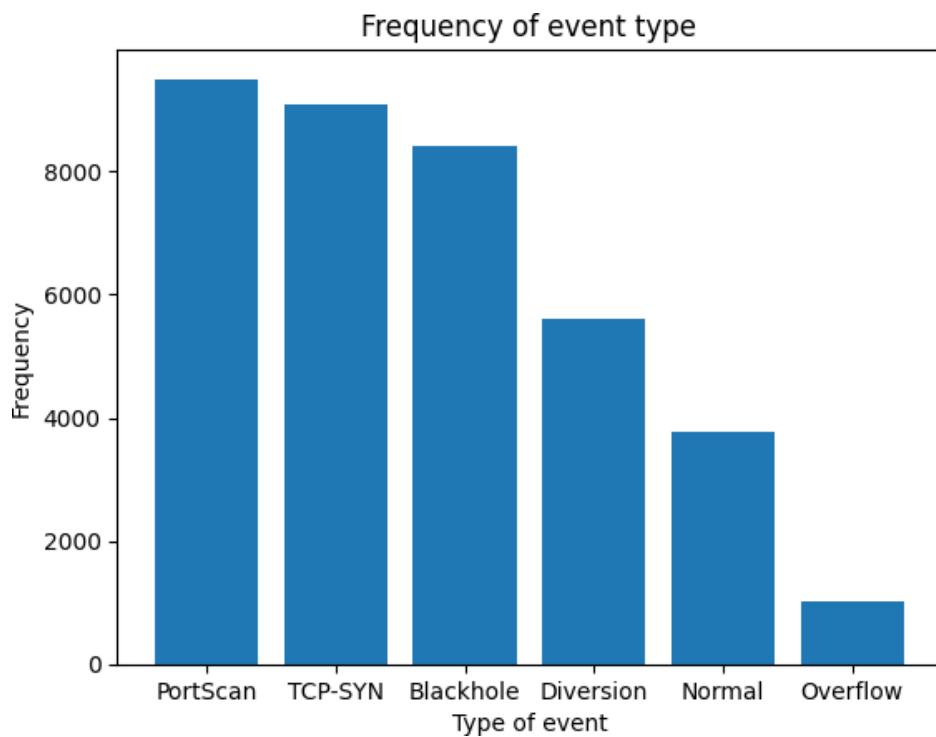


Figure 2: Distribution of events in UNR-IDD dataset based on Label

Observations from above analysis: -

1. We have a high order nominal field: Switch ID with 12 distinct categories, thus, for which we need to identify if one-hot encoding is feasible or if any other better alternative method can be used for training the model.
2. We have a low order nominal field: Port Number, with 4 distinct categories, thus, we can employ one-hot encoding to use the field while training the model.
3. All other fields for training are numeric, and thus, we can normalize them for training the model.
4. For target features, we have two fields: -
 - a. Label: Multi-class classification
 - b. Binary Label: Binary classification
5. The dataset is highly imbalanced for target field: Binary Label. We have a greater number of records of type Attack and lesser number of records of type: Normal.
6. The dataset is imbalanced for target field: Label. There are 3 types of attacks having the greatest number of events in the dataset: -
 - i. PortScan
 - ii. TCP-SYN
 - iii. Blackhole

Foreseen challenges with the dataset: -

1. Imbalanced nature of target features.
2. High order nominal field: Switch ID.

CIC dataset: -

Table 4: List of fields in CIC dataset

Sr No	Field Name	Description	Type of field
1	Flow Duration	The duration of the flow.	Scale
2	Total Fwd Packets	Total number of forward packets	Scale
3	Total Backward Packets	Total number of backward packets	Scale
4	Fwd Packets Length Total	Total length of forward packets	Scale
5	Bwd Packets Length Total	Total length of backward packets	Scale
6	Fwd Packet Length Max	Maximum length of forward packets	Scale
7	Fwd Packet Length Mean	Mean length of forward packets	Scale
8	Fwd Packet Length Std	Standard deviation length of forward packets	Scale
9	Bwd Packet Length Max	Maximum length of backward packets	Scale
10	Bwd Packet Length Mean	Mean length of backward packets	Scale

11	Bwd Packet Length Std	Standard deviation length of backward packets	Scale
12	Flow Bytes/s	Flow bytes per second	Scale
13	Flow Packets/s	Flow packets per second	Scale
14	Flow IAT Mean	Mean time between flows	Scale
15	Flow IAT Std	Standard deviation of time between flows	Scale
16	Flow IAT Max	Maximum time between flows	Scale
17	Flow IAT Min	Minimum time between flows	Scale
18	Fwd IAT Total	Total time between forward packets	Scale
19	Fwd IAT Mean	Mean time between forward packets	Scale
20	Fwd IAT Std	Standard deviation of time between forward packets	Scale
21	Fwd IAT Max	Maximum time between forward packets	Scale
22	Fwd IAT Min	Minimum time between forward packets	Scale
23	Bwd IAT Total	Total time between backward packets	Scale
24	Bwd IAT Mean	Mean time between backward packets	Scale
25	Bwd IAT Std	Standard deviation of time between backward packets	Scale
26	Bwd IAT Max	Maximum time between backward packets	Scale
27	Bwd IAT Min	Minimum time between backward packets	Scale
28	Fwd PSH Flags	Forward packets with PUSH flags	Scale
29	Fwd Header Length	Length of header in forward packets	Scale
30	Bwd Header Length	Length of header in backward packets	Scale
31	Fwd Packets/s	Forward packets per second	Scale
32	Bwd Packets/s	Backward packets per second	Scale
33	Packet Length Max	Maximum length of packets	Scale
34	Packet Length Mean	Mean length of packets	Scale
35	Packet Length Std	Standard deviation length of packets	Scale
36	Packet Length Variance	Variance of length of packets	Scale
37	SYN Flag Count	Number of SYN flags	Scale
38	URG Flag Count	Number of URG flags	Scale
39	Avg Packet Size	Average packet size	Scale
40	Avg Fwd Segment Size	Average forward segment size	Scale
41	Avg Bwd Segment Size	Average backward segment size	Scale
42	Subflow Fwd Packets	Subflow forward packets	Scale
43	Subflow Fwd Bytes	Subflow forward bytes	Scale
44	Subflow Bwd Packets	Subflow backward packets	Scale
45	Subflow Bwd Bytes	Subflow backward bytes	Scale
46	Init Fwd Win Bytes	Initial forward window size	Scale
47	Init Bwd Win Bytes	Initial backward window size	Scale
48	Fwd Act Data Packets	Forward packets with actual data	Scale

49	Fwd Seg Size Min	Minimum segment size in forward packets	Scale
50	Active Mean	Mean active time	Scale
51	Active Std	Standard deviation of active time	Scale
52	Active Max	Maximum active time	Scale
53	Active Min	Minimum active time	Scale
54	Idle Mean	Mean idle time	Scale
55	Idle Std	Standard deviation of idle time	Scale
56	Idle Max	Maximum idle time	Scale
57	Idle Min	Minimum idle time	Scale
58	Label	The intrusion type	Target class: Multi-class classification
59	ClassLabel	Subtype of intrusion	Target class: Multi-class classification

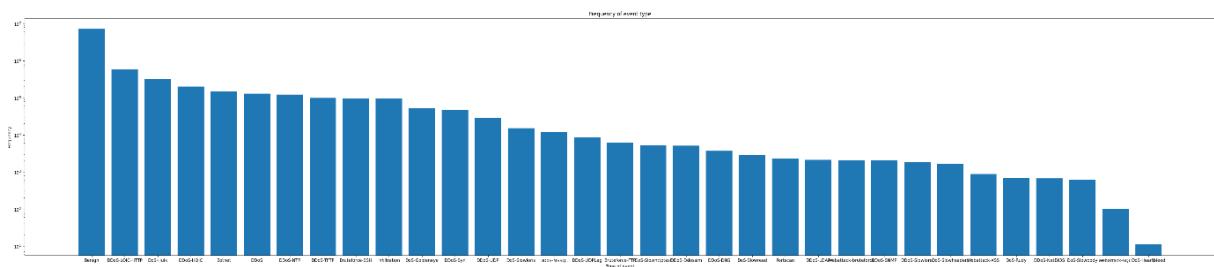


Figure 3: Distribution of events in CIC dataset based on Label

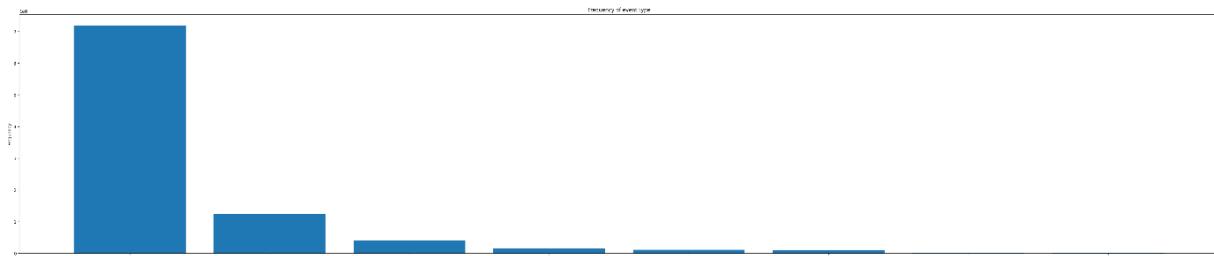


Figure 4: Distribution of events in CIC dataset based on ClassLabel

Observations from above analysis:-

1. We have high dimensional and large volume dataset.
 2. All independent features are of type scale. Thus, we need to normalize them prior training the model.
 3. The target features: Label and ClassLabel are highly imbalanced, the greatest number of events are of type Benign.
 4. In target feature: Label, there are 7186189 records of type Benign ~ 78% of the total records.
 5. Thus, for binary classification, we need to create a new field to differentiate between Benign and Malicious events.

Foreseen challenges with the dataset: -

- ### 1. Imbalanced nature of target features.

Selection of dataset: -

We will build the project using **CIC dataset**.

Reason for selection:-

1. All independent features are of type scale. Thus, we can use normalization operation to handle the data.
2. There are no independent features of type: ordinal or nominal, thus, we will not have more than the original number of features to analyse.
3. We will be able to handle high dimensionality of the dataset using optimization approaches during feature selection.

Chapter 4

Data preprocessing, analysis, visualization and feature engineering

Original shape of the dataset: - (9167581, 59)

Checking and removing duplicate records: -

310 duplicate records were fetched, which were removed. Thus, the new shape of the dataset is (9167271, 59).

0.0042% of duplicate records for Label=Benign were removed.

0.0016% of duplicate records for Label=DDOS-NTP were removed.

0.00016% of duplicate records for ClassLabel=DDOS were removed.

As the result, it was observed that very small proportion of records were removed from the above category of records in the dataset. Thus, the overall distribution of records with respect to Label and ClassLabel have remained the same.

Distribution of data in each independent feature: -

Matplotlib library was used to plot the distribution of all features with chart type: histogram. But, due to large difference in scale, patterns were not observed.

Thus, again the histograms were plotted using log scale which helped to find pattern of distribution for each feature in the dataset.

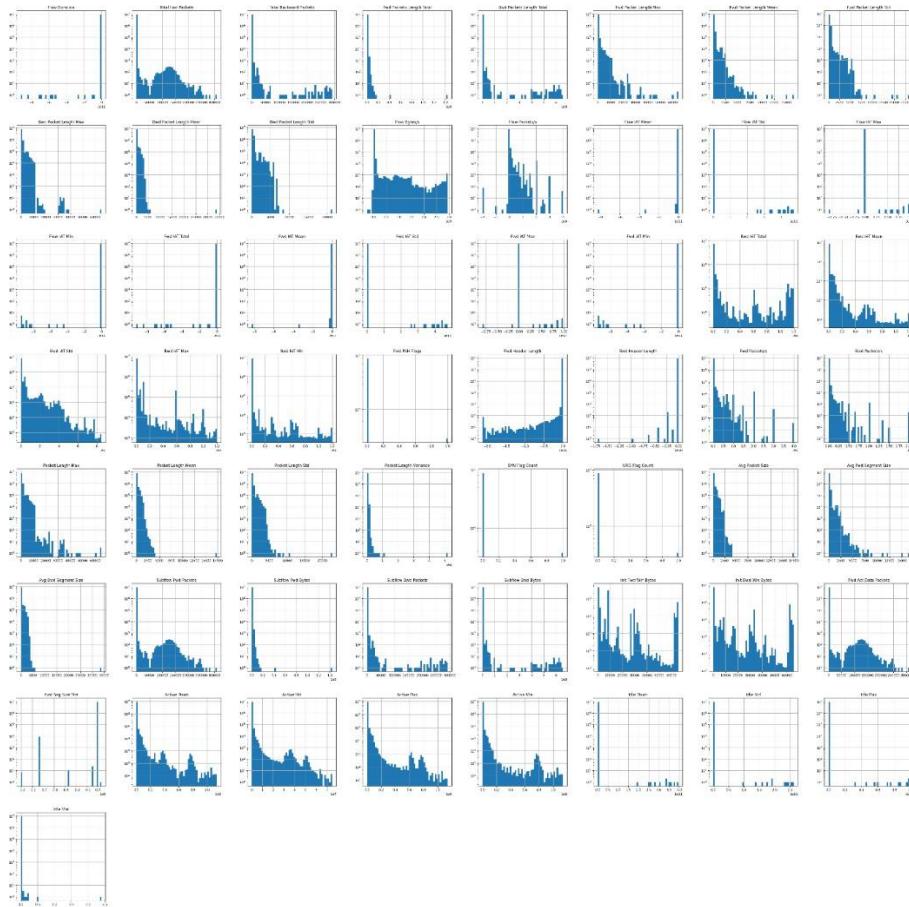


Figure 5: List of histograms for all independent features plotted using log scale

Flow Duration: The duration of the flow

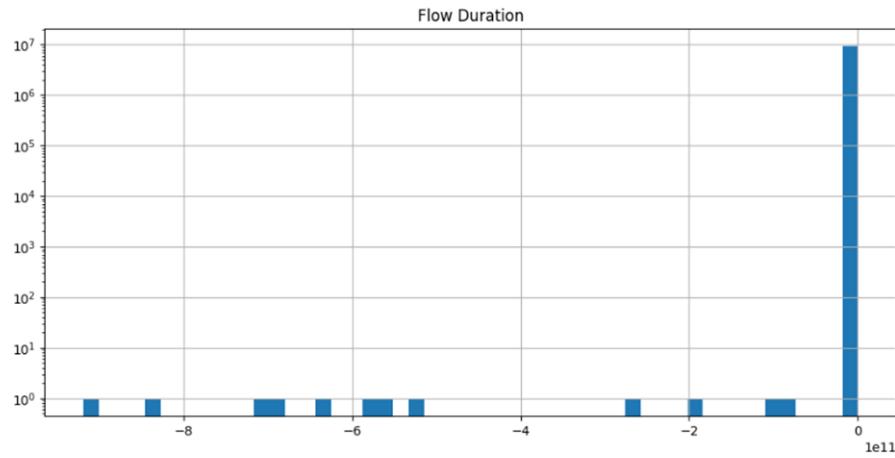


Figure 6: Histogram of Flow duration plotted on log scale

We observed negative values on X-axis, thus, we need to check the actual values under the column to determine if data is accurate or invalid.

Peak was observed at extreme right, Flow Duration=0.

There are some scattered bins of count=1

Total Fwd Packets: Total number of forward packets

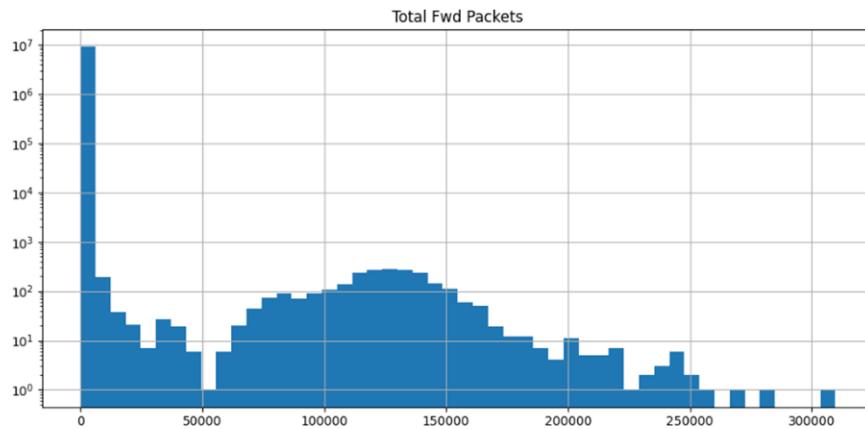


Figure 7: Histogram of Total Fwd Packets plotted on log scale

The distribution is skewed towards right: Positively skewed.

Peak was observed on first bin from the left, after which we saw sharp decline.

There is another small peak around Total Fwd Packets=125000, but it is in plateau shape. Thus, we see many values around 125000.

The first bin (Peak) is in the range around 0 to 6250.

After the second bin , there is consistent decline.

Since there are two peaks at significant distance apart, we can also call the graph bi-modal.

We observed value for Total Fwd Packets>300000. This may indicate outlier in the data.

Total Backward Packets: Total number of backward packets

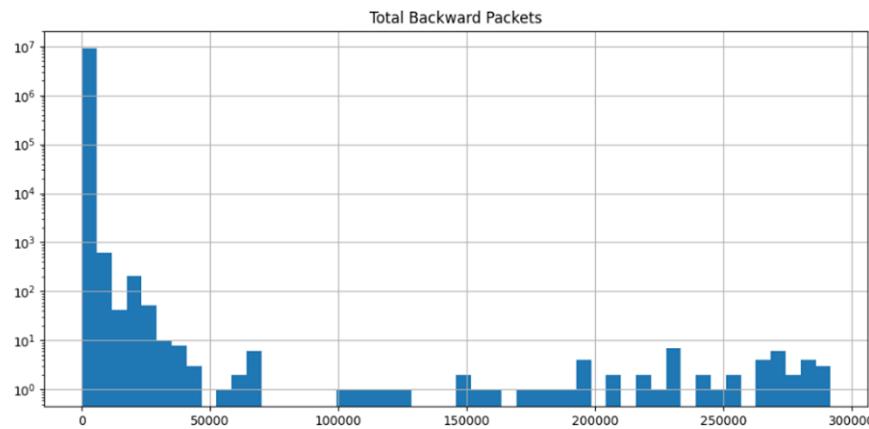


Figure 8: Histogram of Total Backward Packets plotted on log scale

The distribution is skewed towards right: Positively skewed.

Peak was observed on first bin from left, Total Backward Packets: 0 to 6250.

After the peak, there is significant decline in results.

Some records were observed at regular intervals but with very less frequency.

Fwd Packets Length Total: Total length of forward packets

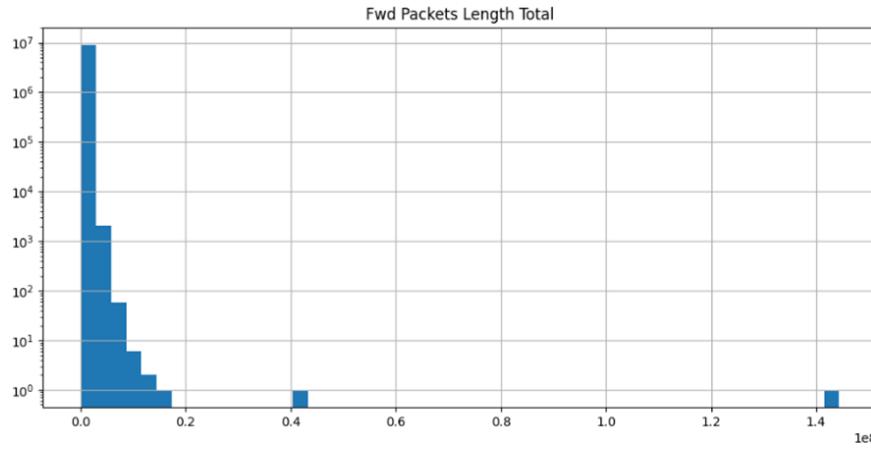


Figure 9: Histogram of Forward Packets Length Total plotted on log scale

Peak was observed on first bin from left.

Most values are stacked on the left side of X-axis and they continuously decline as we move towards right hand side of X-axis.

There are a couple of observations at a distance on right hand side after long gap. They may indicate outliers in the data.

Bwd Packets Length Total: Total length of backward packets

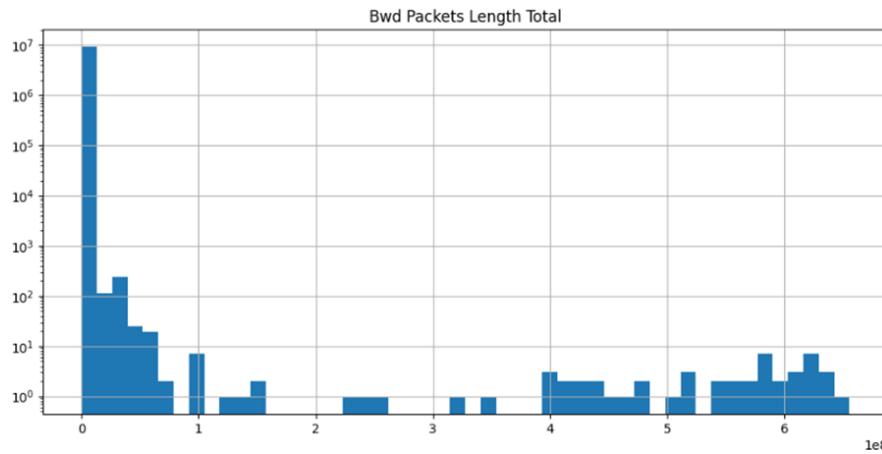


Figure 10: Histogram of Bwd Packets Length Total plotted on log scale

The distribution is skewed towards right: Positively skewed.

Peak was observed on first bin from left.

After the peak, there is significant decline in results.

There are some observations spread out on X-axis, but all have frequency less than 10.

Fwd Packet Length Max: Maximum length of forward packets

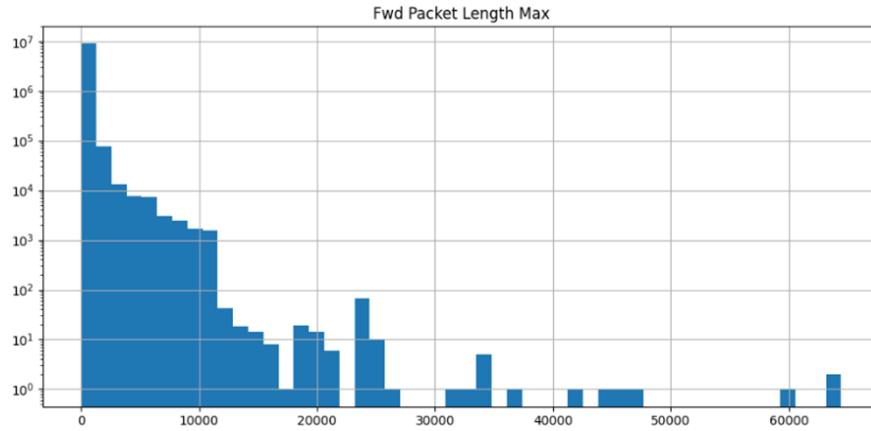


Figure 11: Histogram of Fwd Packet Length Max plotted on log scale

The distribution is skewed towards right: Positively skewed.

Peak was observed on first bin from left.

The first bin (Peak) is in the range around 0 to 1250.

Most number of observations lie between Fwd Packet Length Max>0 and Fwd Packet Length Max<10000.

A small peak was observed around Fwd Packet Length Max>20000 and Fwd Packet Length Max<300000. However the frequency is relatively very less compared to the peak observed in first bin.

There are some observations around Fwd Packet Length Max=60000. This may indicate outliers in the data.

Fwd Packet Length Mean: Mean length of forward packets

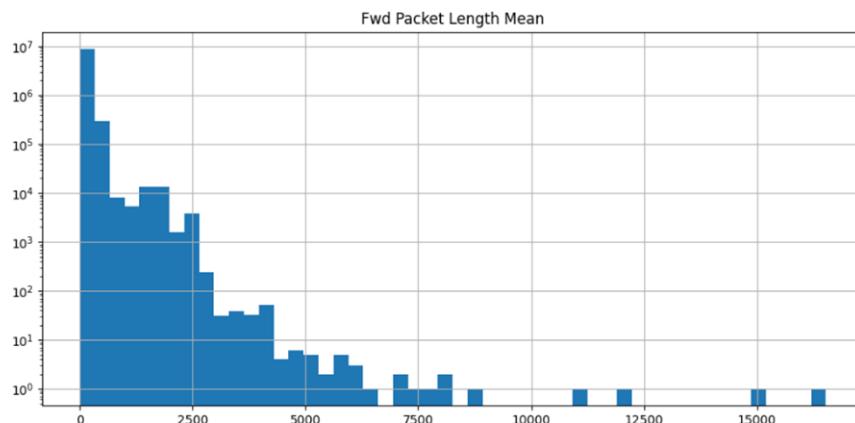


Figure 12: Histogram of Fwd Packet Length Mean plotted on log scale

The distribution is skewed towards right: Positively skewed.

Peak was observed on first bin from left.

Most number of observations lie between Fwd Packet Length Mean \geq 0 and Fwd Packet Length Mean \leq 2500.

There are some small number of observations around Fwd Packet Length Mean=15000 and above. This may indicate outliers in the data.

Fwd Packet Length Std: Standard deviation length of forward packets

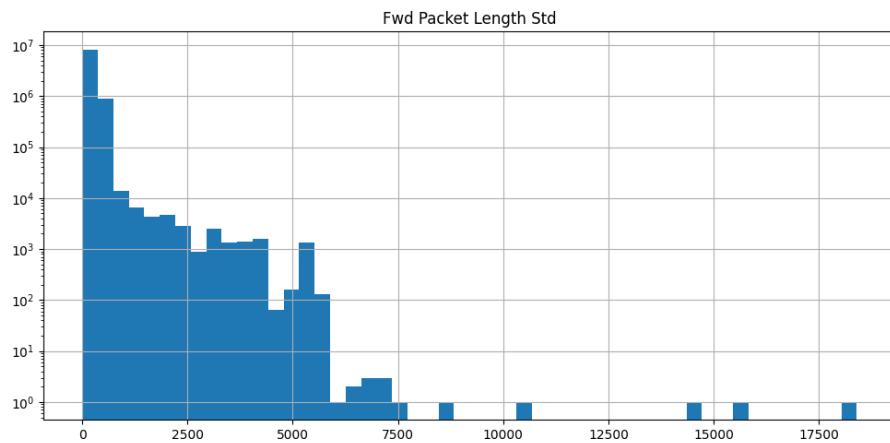


Figure 13: Histogram of Fwd Packet Length Std plotted on log scale

The distribution is skewed towards right: Positively skewed.

Peak was observed on first bin from left.

Most number of observations lie between Fwd Packet Length Std \geq 0 and Fwd Packet Length Std \leq 5000.

There are some very small number of observations at Fwd Packet Length Std $>$ 7500. This may indicate outliers in the data.

Bwd Packet Length Max: Maximum length of backward packets

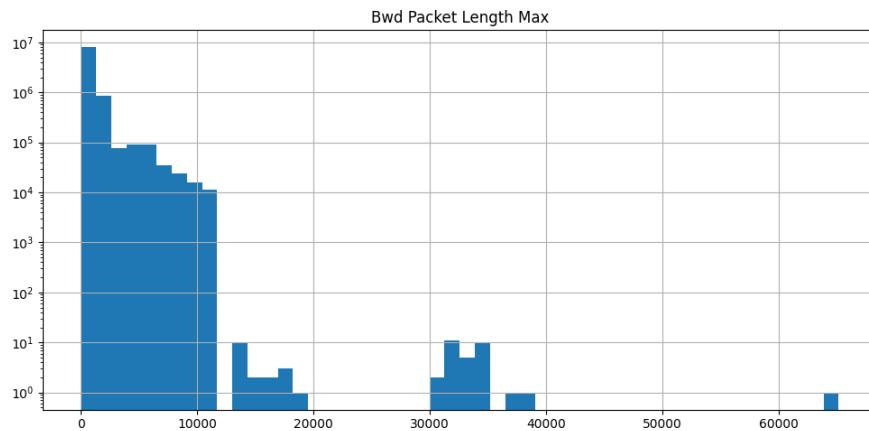


Figure 14: Histogram of Bwd Packet Length Max plotted on log scale

The distribution is skewed towards right: Positively skewed.

Peak was observed on first bin from left. Peak lies around $\text{Bwd Packet Length Max} \geq 0$ and $\text{Bwd Packet Length Max} \leq 1250$.

Most number of observations lie between $\text{Bwd Packet Length Max} \geq 0$ and $\text{Bwd Packet Length Max} \leq 10000$.

There are few observations in the range: - $\text{Bwd Packet Length Max} \geq 11250$ and $\text{Bwd Packet Length Max} \leq 20000$, $\text{Bwd Packet Length Max} \geq 30000$ and $\text{Bwd Packet Length Max} \leq 35000$.

There is an observation at $\text{Bwd Packet Length Max} > 60000$. This may indicate outliers in the data.

Bwd Packet Length Mean: Mean length of backward packet

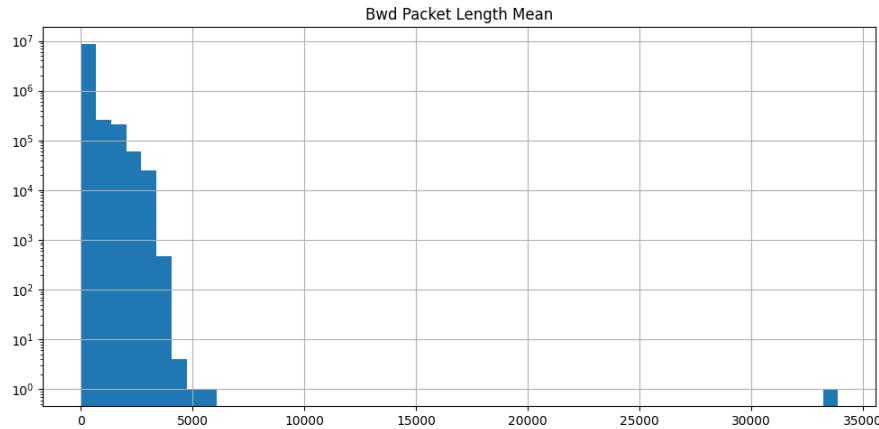


Figure 15: Histogram of Bwd Packet Length Mean plotted on log scale

The distribution is skewed towards right: Positively skewed.

Peak was observed on first bin from left. Peak lies around $\text{Bwd Packet Length Mean} \geq 0$ and $\text{Bwd Packet Length Mean} \leq 666.67$.

After the peak, there is significant decline in results.

Between Bwd Packet Length Mean=0 and Bwd Packet Length Mean=5000, we observed J-shaped graph.

There is an observation at Bwd Packet Length Mean=35000. This may indicate outliers in the data.

Bwd Packet Length Std: Standard deviation length of backward packets

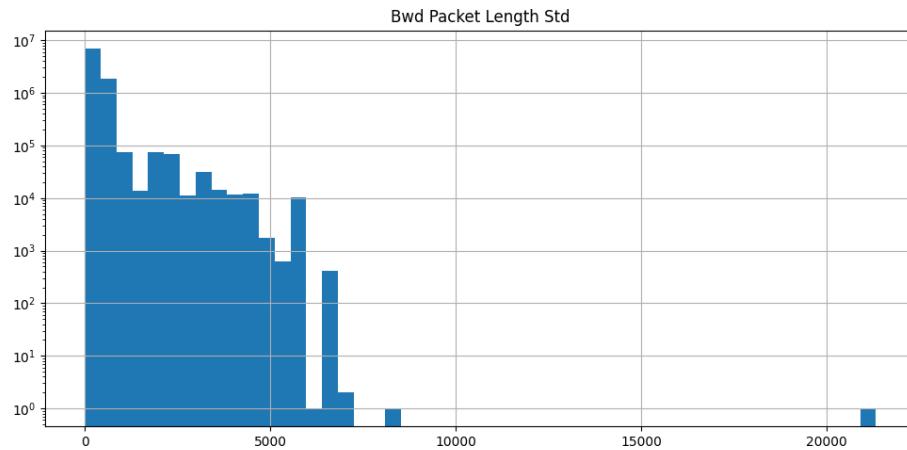


Figure 16: Histogram of Bwd Packet Length std plotted on log scale

The distribution is skewed towards right: Positively skewed.

Peak was observed on first bin from left. Peak lies around $\text{Bwd Packet Length Std} \geq 0$ and $\text{Bwd Packet Length Std} \leq 416.67$.

After the peak, there is significant decline in results.

There is plateau region observed around $\text{Bwd Packet Length Std} \geq 2083$ and $\text{Bwd Packet Length Std} \leq 2500$.

There is another plateau region observed (smaller than the above) around $\text{Bwd Packet Length Std} \geq 3750$ and $\text{Bwd Packet Length Std} \leq 4166$.

There is an observation at $\text{Bwd Packet Length Std} > 20000$. This may indicate outliers in the data.

Flow Bytes/s: Flow bytes per second

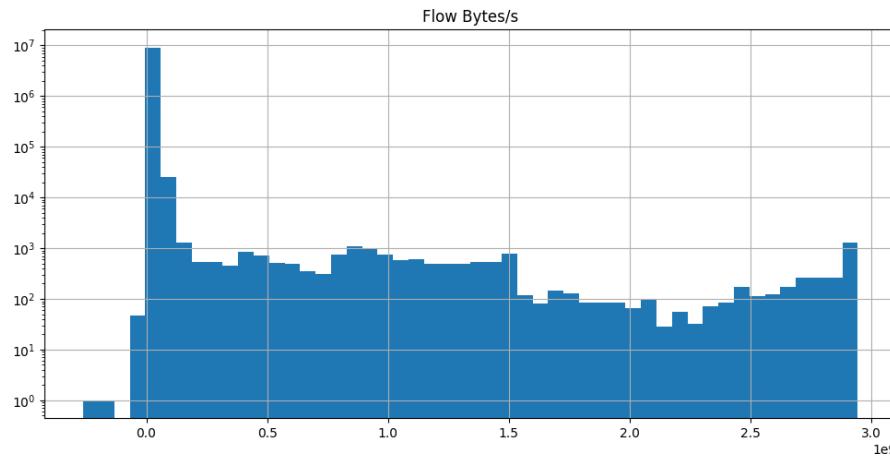


Figure 17: Histogram of Flow Bytes/s plotted on log scale

The distribution is skewed towards right: Positively skewed.

Peak was observed around Flow Bytes/s=0.

After the peak, there is consistent decline in results.

Towards right hand side of the graph, there is increase in number of observations compared to other bins prior to it excluding the peak.

Between the two extremes of the graph there were some plateau regions.

We observed negative values on X-axis, thus, we need to check the actual values under the column to determine if data is accurate or invalid.

Flow Packets/s: Flow packets per second

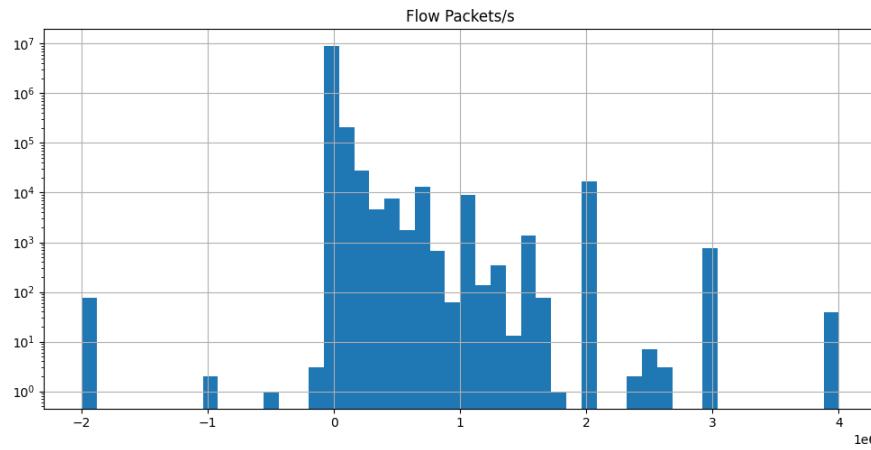


Figure 18: Histogram of Flow Packets/s plotted on log scale

The distribution is skewed towards right: Positively skewed.

Peak was observed around Flow Packets/s=0

After the peak, there is consistent decline in results.

At Flow Packets/s=2 and Flow Packets/s=3, there relatively small peaks.

We observed negative values on X-axis, thus, we need to check the actual values under the column to determine if data is accurate or invalid.

Flow IAT Mean: Mean time between flows

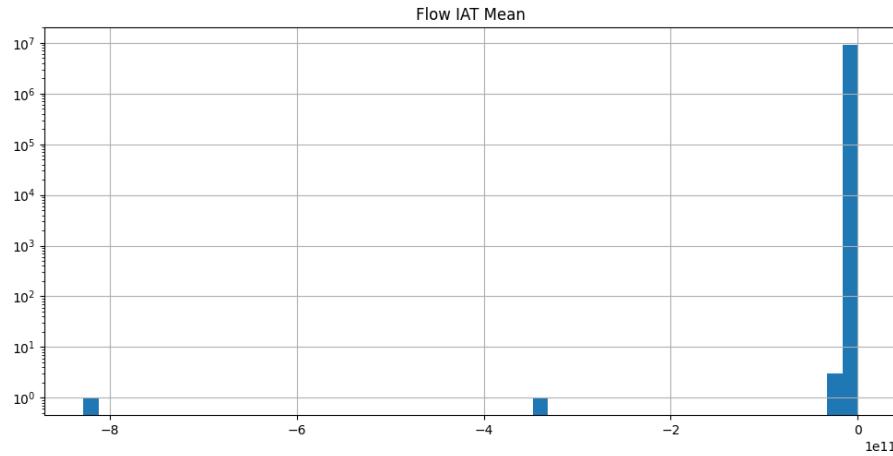


Figure 19: Histogram of Flow IAT Mean plotted on log scale

Peak was observed at Flow IAT Mean=0.

Most values are concentrated in bin represented by the peak.

We observed negative values on X-axis, thus, we need to check the actual values under the column to determine if data is accurate or invalid.

Flow IAT Std: Standard deviation of time between flows

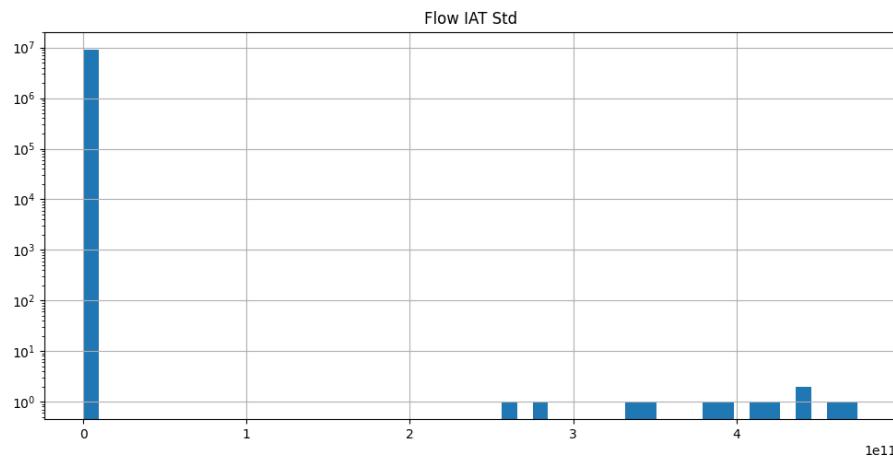


Figure 20: Histogram of Flow IAT Std plotted on log scale

The distribution is skewed towards right: Positively skewed.

Peak was observed around Flow IAT Std=0.

Most values are concentrated in bin represented by the peak.

There are a few observations in the range: - Flow IAT Std>=2 and Flow IAT Std<=3, Flow IAT Std>=3 and Flow IAT Std<=4 and Flow IAT Std>4.

Flow IAT Max: Maximum time between flows

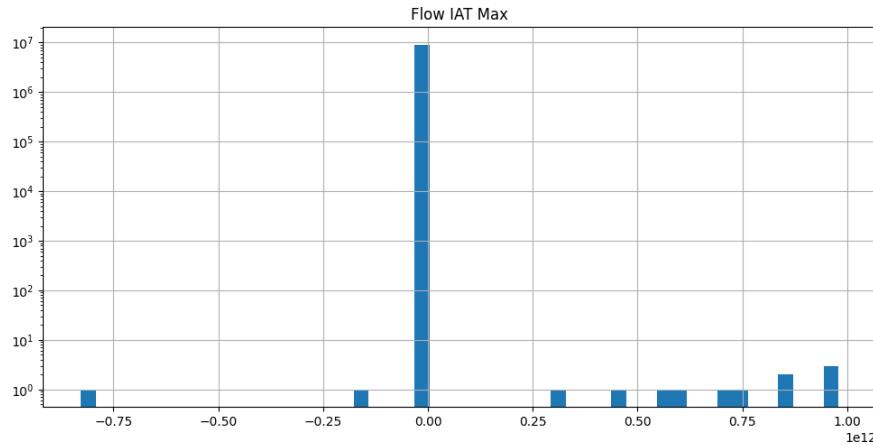


Figure 21: Histogram of Flow IAT Max plotted on log scale

Peak was observed around Flow IAT Max=0.

We observed negative values on X-axis, thus, we need to check the actual values under the column to determine if data is accurate or invalid.

On X-axis values lie in the range -1.0 to +1.0

Flow IAT Min: Minimum time between flows

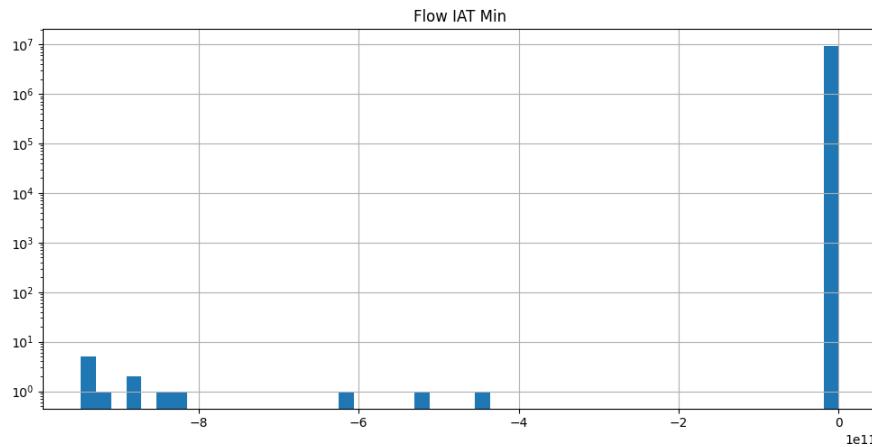


Figure 22: Histogram of Flow IAT Min plotted on log scale

Peak was observed around Flow IAT Min=0.

We observed negative values on X-axis, thus, we need to check the actual values under the column to determine if data is accurate or invalid.

Fwd IAT Total: Total time between forward packets

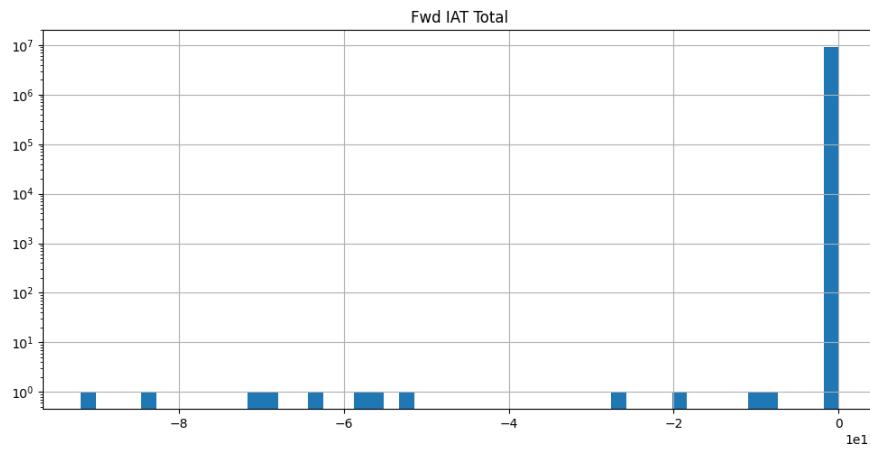


Figure 23: Histogram of Flow IAT Total plotted on log scale

Peak was observed around Fwd IAT Total=0.

We observed negative values on X-axis, thus, we need to check the actual values under the column to determine if data is accurate or invalid.

Fwd IAT Mean: Mean time between forward packets

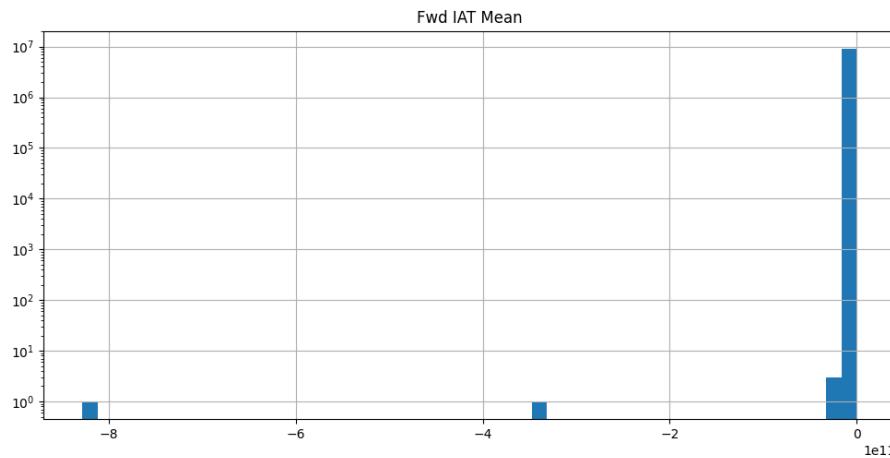


Figure 24: Histogram of Fwd IAT Mean plotted on log scale

Peak was observed around Fwd IAT Mean=0.

We observed negative values on X-axis, thus, we need to check the actual values under the column to determine if data is accurate or invalid.

Fwd IAT Std: Standard deviation of time between forward packets

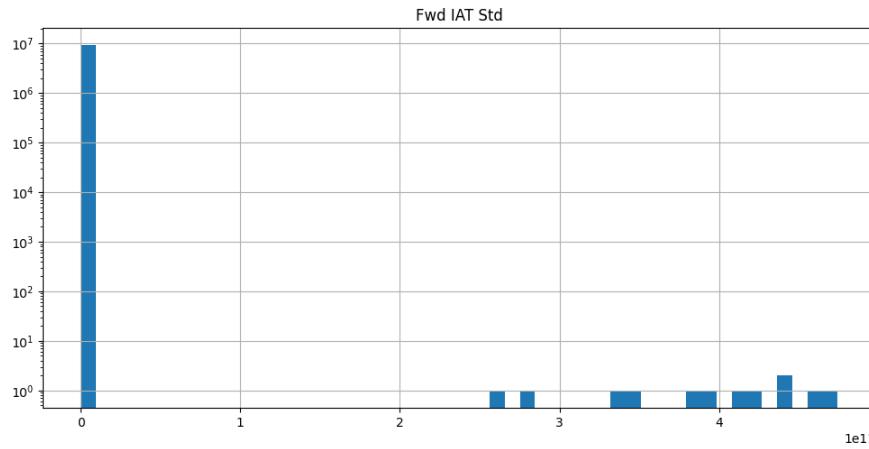


Figure 25: Histogram of Fwd IAT Std plotted on log scale

Peak was observed around Fwd IAT Std=0.

There are small number of observations in the range: Fwd IAT Std \geq 2 and Fwd IAT Std \leq 3, Fwd IAT Std \geq 3 and Fwd IAT Std \leq 4, Fwd IAT Std $>$ 4.

Fwd IAT Max: Maximum time between forward packets

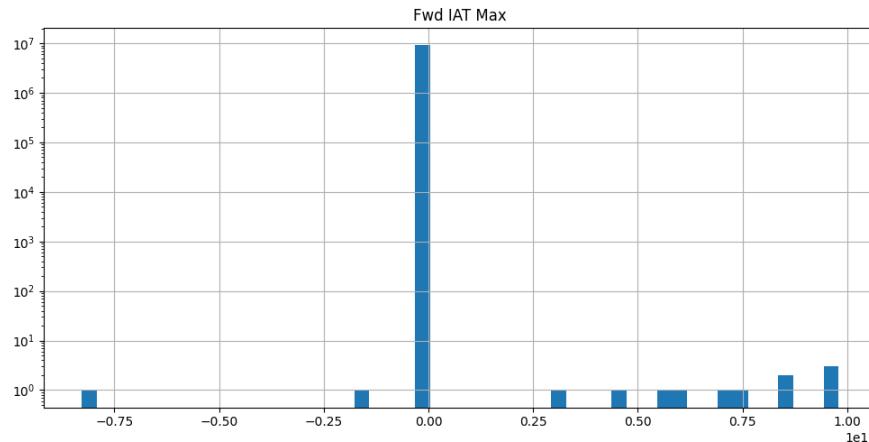


Figure 26: Histogram of Fwd IAT Max plotted on log scale

Peak was observed around Fwd IAT Max=0.

We observed negative values on X-axis, thus, we need to check the actual values under the column to determine if data is accurate or invalid.

There are scattered but very small number of observations between Fwd IAT Max=0.0 and Fwd IAT Max=1.0

Fwd IAT Min: Minimum time between forward packets

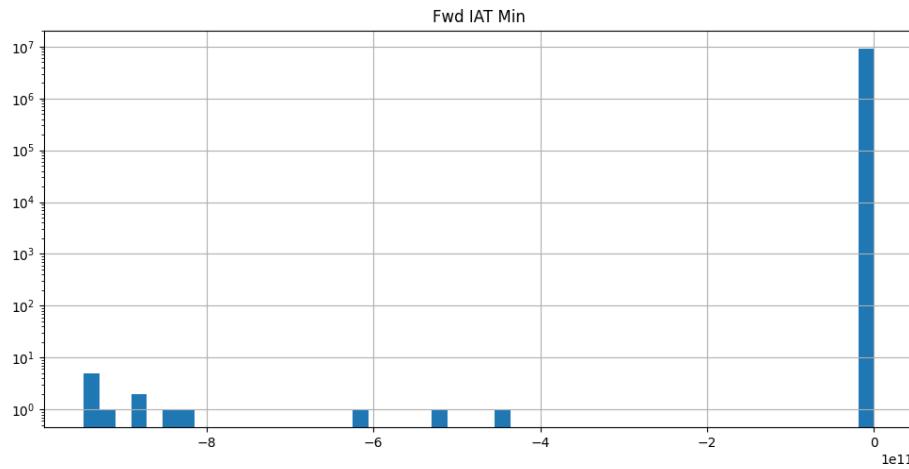


Figure 27: Histogram of Fwd IAT Min plotted on log scale

Peak was observed around Fed IAT Min=0.

We observed negative values on X-axis, thus, we need to check the actual values under the column to determine if data is accurate or invalid.

Bwd IAT Total: Total time between backward packets

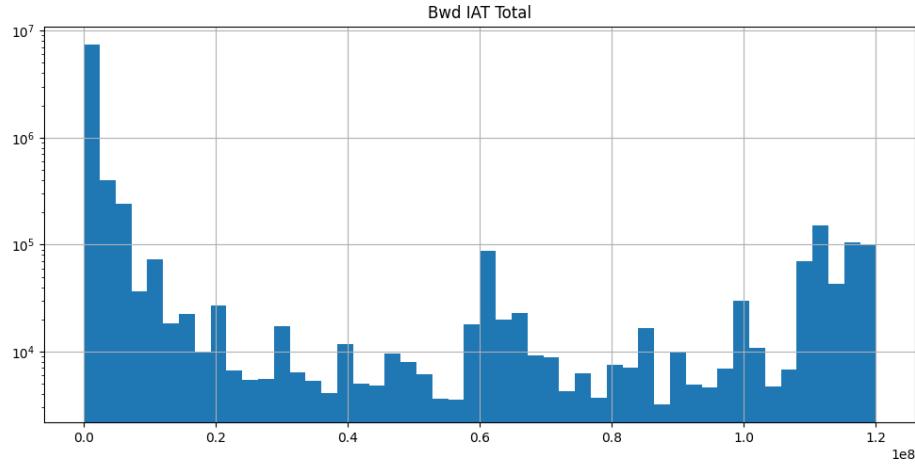


Figure 28: Histogram of Bwd IAT Total plotted on log scale

Peak was observed around Bwd IAT Total=0.

After the peak, there is consistent decline in results.

There are relatively smaller peaks at Bwd IAT Total=0.6 and Bwd IAT Total=1.125

There was a plateau region observed between Bwd IAT Total>=0.625 and Bwd IAT Total<=0.675

Bwd IAT Mean: Mean time between backward packets

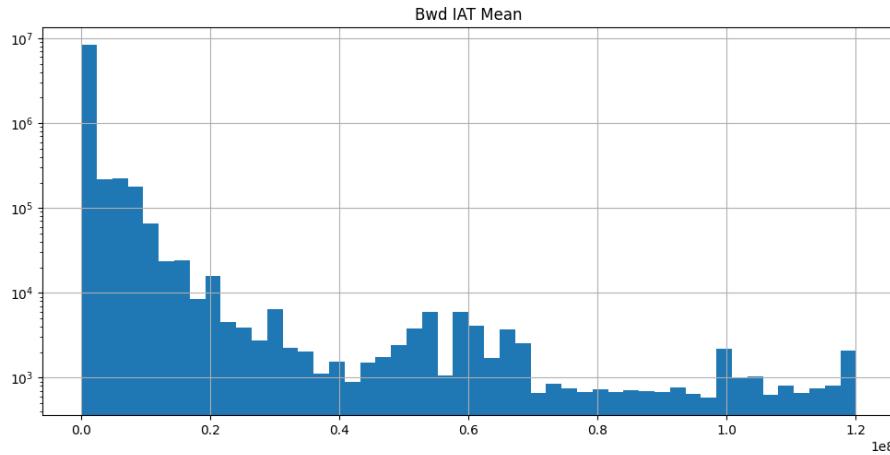


Figure 29: Histogram of Bwd IAT Mean plotted on log scale

Peak was observed around Bwd IAT Mean=0.

After the peak, there is consistent decline in results.

Most observations are stacked on left side of the graph, near the peak.

On X-axis values lie in the range 0.0 to +1.2

Bwd IAT Std: Standard deviation of time between packets

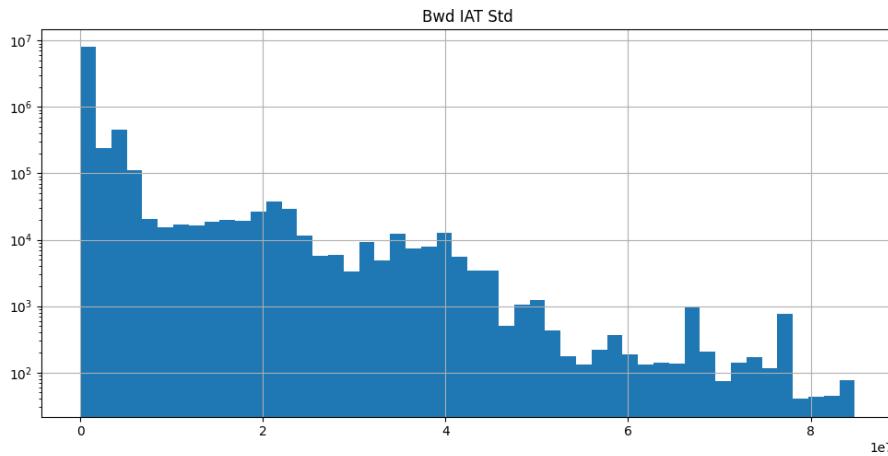


Figure 30: Histogram of Bwd IAT Std plotted on log scale

The distribution is skewed towards right: Positively skewed.

Peak was observed around Bwd IAT Std=0

After the peak, there is consistent decline in results.

There was plateau region observed between Bwd IAT Std \geq 1.169 and Bwd IAT Std=2.

As the value of Bwd IAT Std increases, the size of bins decreases. In between there are a few exceptions where size of bin is greater than their neighbours.

Bwd IAT Max: Maximum time between packets

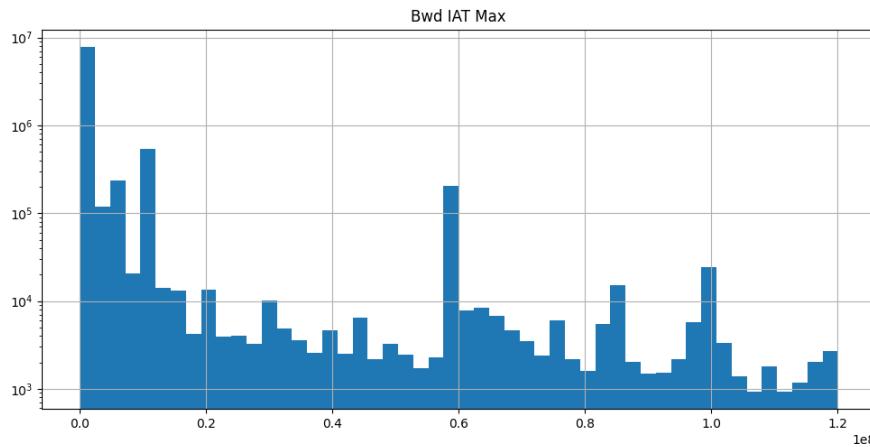


Figure 31: Histogram of Bwd IAT Max plotted on log scale

The distribution is skewed towards right: Positively skewed.

Peak was observed around Bwd IAT Max=0.0

After the peak, there is consistent decline in results.

There are relatively smaller peaks at Bwd IAT Max=0.125 and Bwd IAT Max=0.575

Since there are multiple peaks at significant distance apart, we can also call the graph multi-modal.

The bins prior and after all three peaks are very small.

Bwd IAT Min: Minimum time between packets

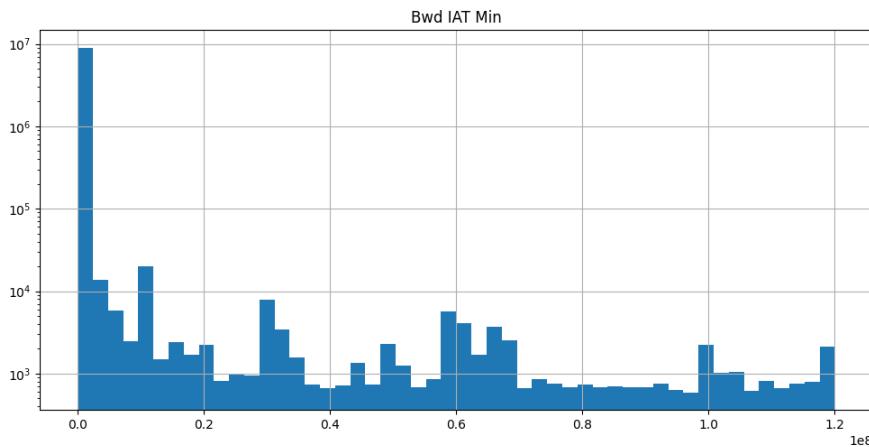


Figure 32: Histogram of Bwd IAT Min plotted on log scale

The distribution is skewed towards right: Positively skewed.

Peak was observed around Bwd IAT Min=0

After the peak, there is significant decline in results.

On X-axis values lie in the range 0.0 to 1.2

Fwd PSH Flags: Forward packets with PUSH flags

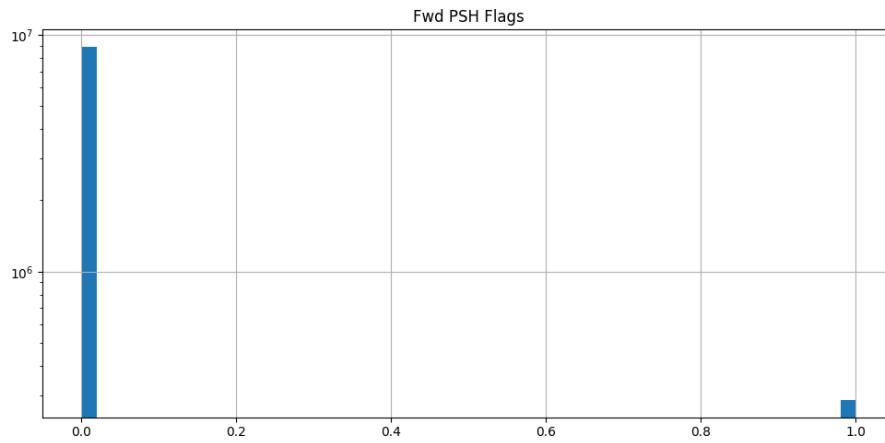


Figure 33: Histogram of Fwd PSH Flags plotted on log scale

Most of the values are concentrated in the first bin at Fwd PSH Flags=0.0

There were few observations at Fwd PSH Flags=1.0. This may indicate outlier in the data.

There were no results between Fwd PSH Flags=0.0 and Fwd PSH Flags=1.0

Fwd Header Length: Length of header in forward packets

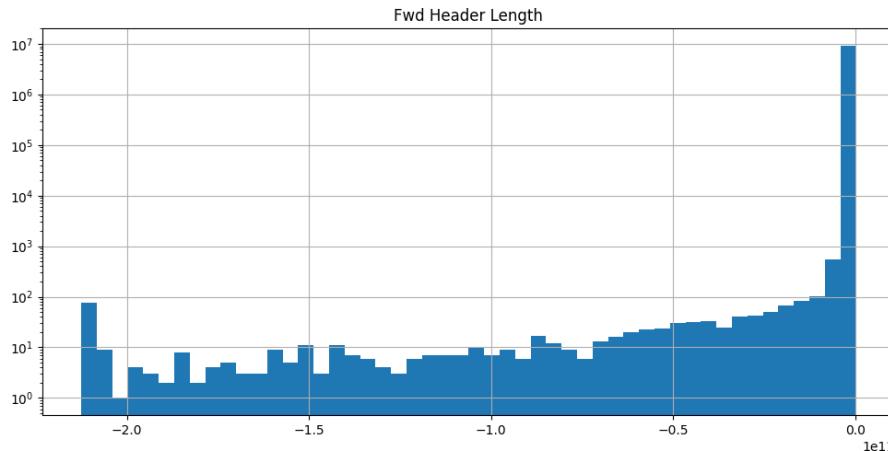


Figure 34: Histogram of Fwd Header Length plotted on log scale

The distribution is skewed towards left: Negatively skewed.

Peak was observed around Fwd Header Length=0.0

There were no results for Fwd Header Length>0.0

There are relatively smaller size bins of left hand side of the peak.

On X-axis values lie in the range -2.0 to 0.0

Bwd Header Length: Length of header in backward packets

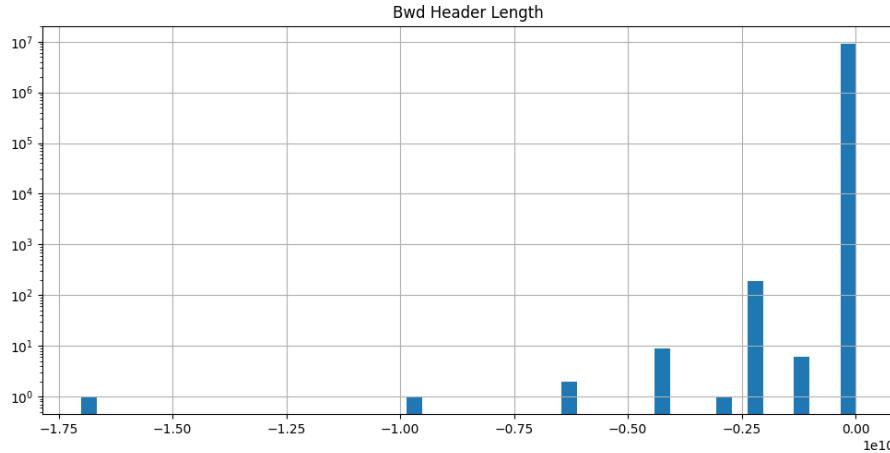


Figure 35: Histogram of Bwd Header Length plotted on log scale

Peak was observed around Bwd Header Length=0.0

Most values are concentrated at the peak.

There were few observations at Bwd Header Length=-1.75, -1, -0.6, -0.30

There were no results for Bwd Header Length>0.0

On X-axis values lie in the range -1.75 to 0.0

Fwd Packets/s: Forward packets per second

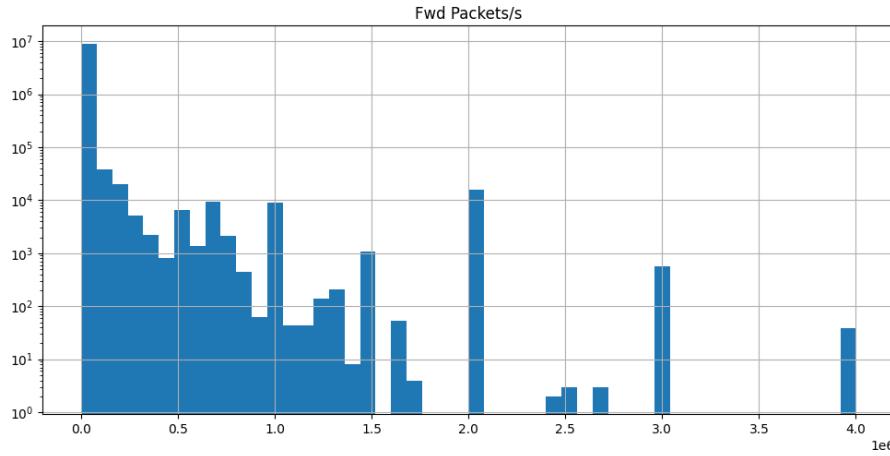


Figure 36: Histogram of Fwd Packets/s plotted on log scale

The distribution is skewed towards right: Positively skewed.

Peak was observed around Fwd Packets/s=0

From Fwd Packets/s=0.0 to 1.5, the values are stacked to the right hand side of peak.

There are relatively smaller peaks at Fwd Packets/s= 2.0, 3.0, 4.0

There is a wide gap (no results) between Fwd Packets/s=3.0 and Fwd Packets/s=4.0

Most values are concentrated between Fwd Packets/s=0.0 and Fwd Packets/s=1.5. Between this range the graph also resembles to J-shaped graph.

Bwd Packets/s: Backward packets per second

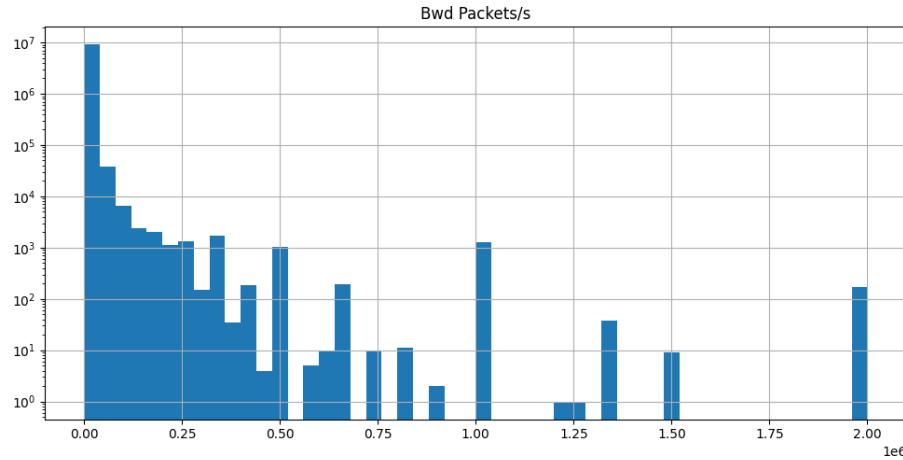


Figure 37: Histogram of Bwd Packets/s plotted on log scale

The distribution is skewed towards right: Positively skewed.

Peak was observed around Bwd Packets/s=0.0

Most values are concentrated between Bwd Packets.s=0.0 and Bwd Packets/s=0.5. Between this range the graph also resembles to J-shaped graph.

There are relatively smaller peaks at Bwd Packets/s=0.5, 1.0 and 2.0

After Bwd Packets/s>=1.0, the bins are scattered and gaps were observed at irregular intervals on the x-axis.

Packet Length Max: Maximum length of packets

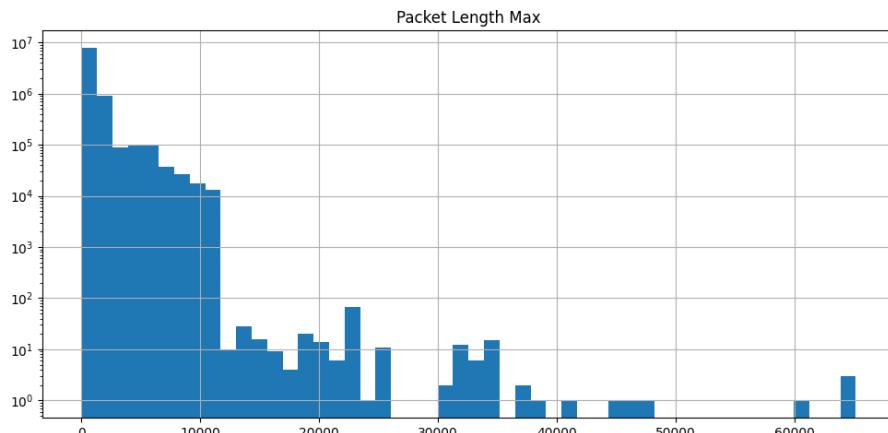


Figure 38: Histogram of Packet Length Max plotted on log scale

The distribution is skewed towards right: Positively skewed.

Peak was observed around Packet Length Max=0

After the peak, there is significant decline in results between Packet Length Max \geq 0 and Packet Length Max \leq 10000.

Between Packet Length Max=0 and Packet Length Max=10000, the graph also resemble to J-shaped graph.

Between Packet Length Max=10000 to 26000, the results are significantly lower than Packet Length Max=0 to 10000.

There were no results observed between Packet Length Max=26000 to 30000, 50000 to 60000.

There are some results observed between Packet Length Max=30000 to 50000.

There are small number of results observed for Packet Length Max $>$ 60000. This may indicate outlier in the data.

Packet Length Mean: Mean length of packets

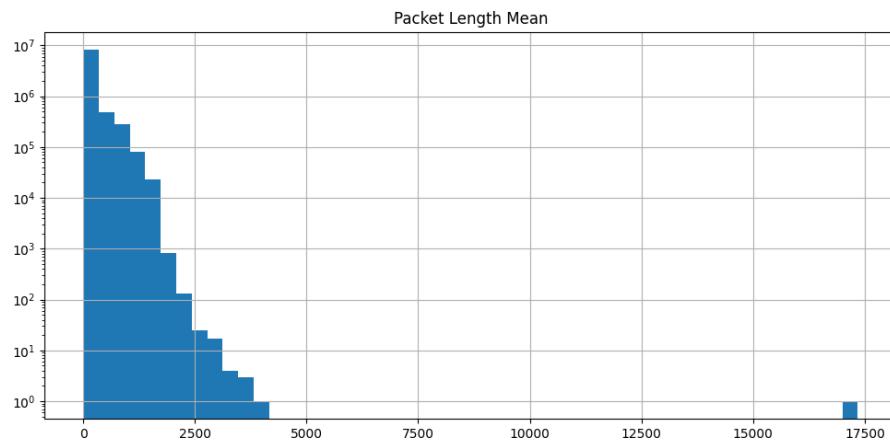


Figure 39: Histogram of Packet Length Mean plotted on log scale

On X-axis, values lie in the range 0 to 17500.

The distribution is a J-shaped graph.

Peak was observed around Packet Length Mean=0

All other bins are stacked against the peak on its right hand side.

There is a constant decline of results as we move towards right side of the graph.

The results are concentrated between Packet Length Mean \geq 0 and Packet Length Mean $<$ 5000.

There is a small observation at Packet Length Mean=17500. This may indicate an outlier in the data.

Packet Length Std: Standard deviation length of packets

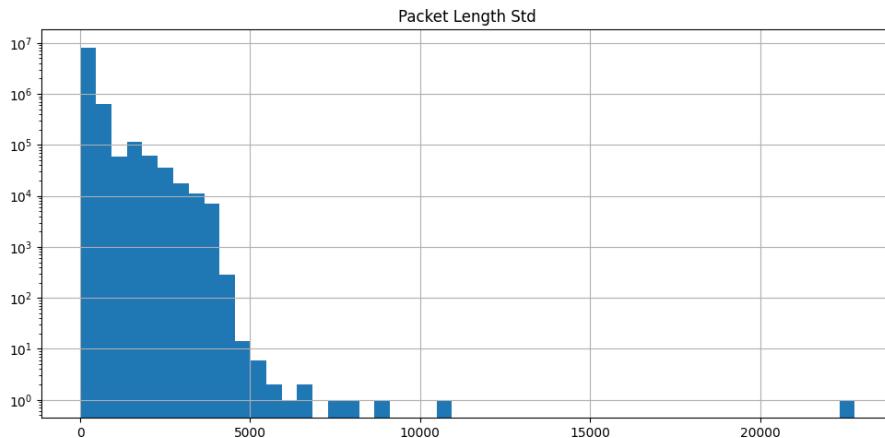


Figure 40: Histogram of Packet Length Std plotted on log scale

The distribution is a J-shaped graph.

Peak was observed around Packet Length Std=0

All other bins are stacked against the peak on its right hand side.

Most values are concentrated between Packet Length Std ≥ 0 and Packet Length Std ≤ 5000 .

There is an observation at Packet Length Std > 20000 . This may indicate an outlier in the data.

On X-axis, values lie in the range 0 to 20000.

Packet Length Variance: Variance of length of packets

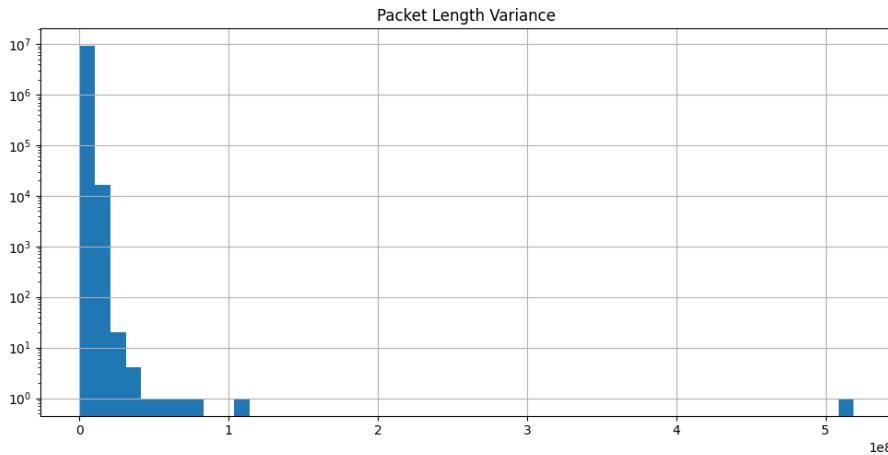


Figure 41: Histogram of Packet Length Variance plotted on log scale

The distribution is skewed towards right: Positively skewed.

Peak was observed around Packet Length Variance=0

Most values are concentrated between Packet Length Variance ≥ 0 and Packet Length Variance ≤ 1 .

There is an observation after long gap at Packet Length Variance > 5 . This may indicate an outlier in the data.

SYN Flag Count: Number of SYN flags

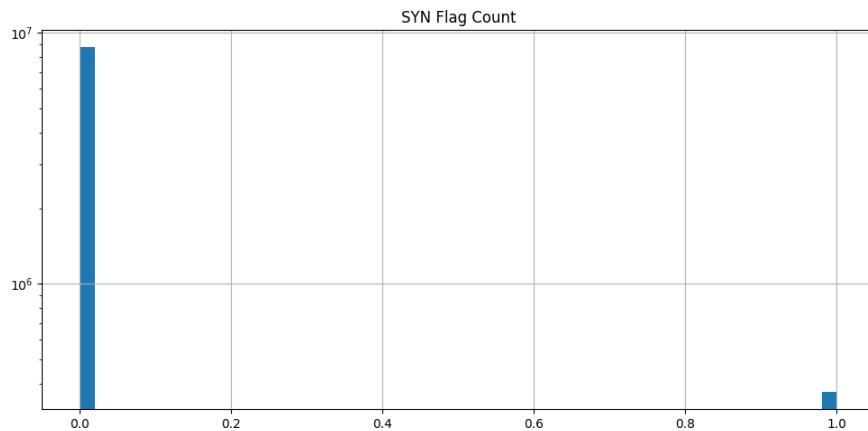


Figure 42: Histogram of SYN Flag Count plotted on log scale

Peak was observed at SYN Flag Count=0.

Most values are concentrated at the peak.

There are a few observations at SYN Flag Count=1.0. This may indicate outlier in the data.

URG Flag Count: Number of URG flags

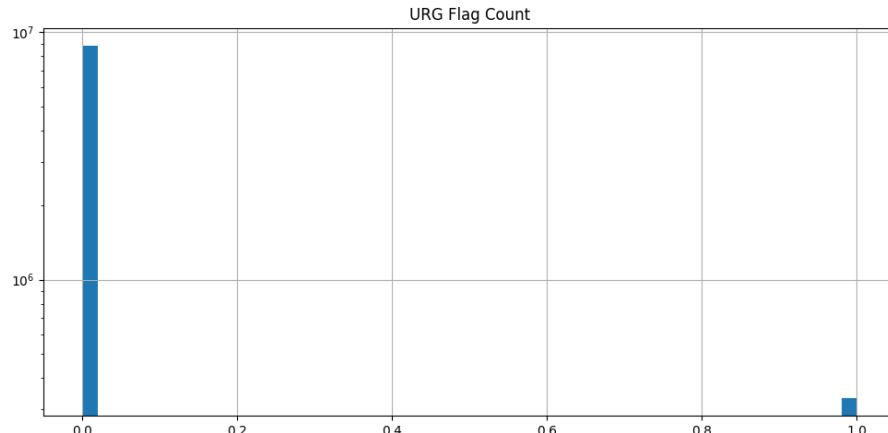


Figure 43: Histogram of URG Flag Count plotted on log scale

Peak was observed at URG Flag Count=0.

Most values are concentrated at the peak.

There are a few observations at URG Flag Count=1.0. This may indicate outlier in the data.

Avg Packet Size: Average packet size

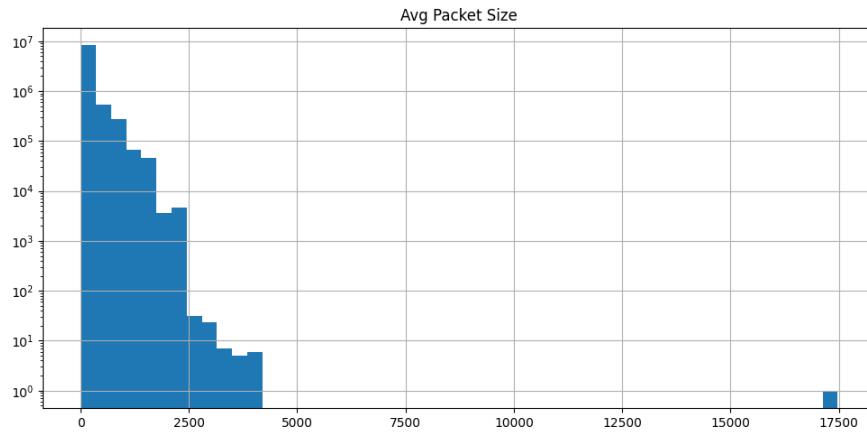


Figure 44: Histogram of Avg Packet Size plotted on log scale

The distribution is J-shaped graph.

Most of the values are stacked at left end and then it continuously declines as we move towards right hand side of the x-axis.

Peak was observed at Avg Packet Size=0.

Most values are concentrated between Avg Packet Size>=0 and Avg Packet Size<5000.

There were some values after a long gap between Avg Packet Size>5000 and Avg Packet Size <=17500. This may indicate outlier in the data.

Avg Fwd Segment Size: Average forward segment size

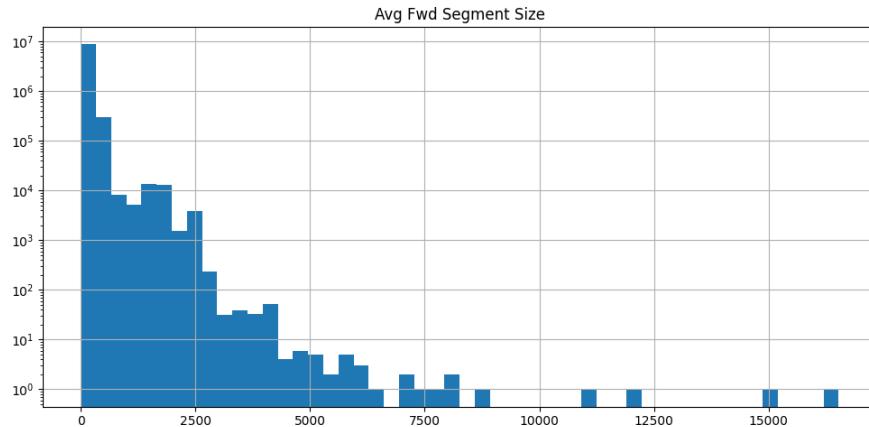


Figure 45: Histogram of Avg Fwd Segment Size plotted on log scale

The distribution is skewed towards right: Positively skewed.

Peak was observed at Avg Fwd Segment Size=0.

After the peak, there is consistent decline in results.

Most values are concentrated between Avg Fwd Segment Size>=0 and Avg Fwd Segment Size<=5000.

There were some values around Avg Fwd Segment Size=7500.

There were couple of values observed in range Avg Fwd Segment Size>10000 and Avg Fwd Segment Size<12500, Avg Fwd Segemnt Size>=15000. This may indicate outlier in the data.

Avg Bwd Segment Size: Average backward segment size

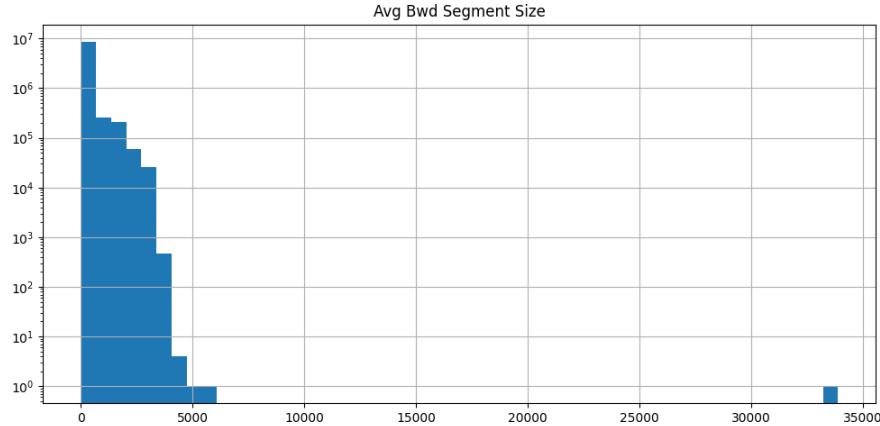


Figure 46: Histogram of Avg Bwd Segment Size plotted on log scale

The distribution is J-shaped graph.

Most of the values are stacked at left end and then it continuously declines as we move towards right hand side of the x-axis.

Peak was observed at Avg Bwd Segment Size=0.

Most values are concentrated between Avg Bwd Segment Size>=0 and Avg Bwd Segment Size<=5000.

There is a long gap observed after Avg Bwd Segment Size>5000.

On extreme right end side of the graph, between Avg Bwd Segment Size>=30000 and Avg Bwd Segment Size<=35000, few values were observed. This may indicate outlier in the data.

Subflow Fwd Packets: Subflow forward packets

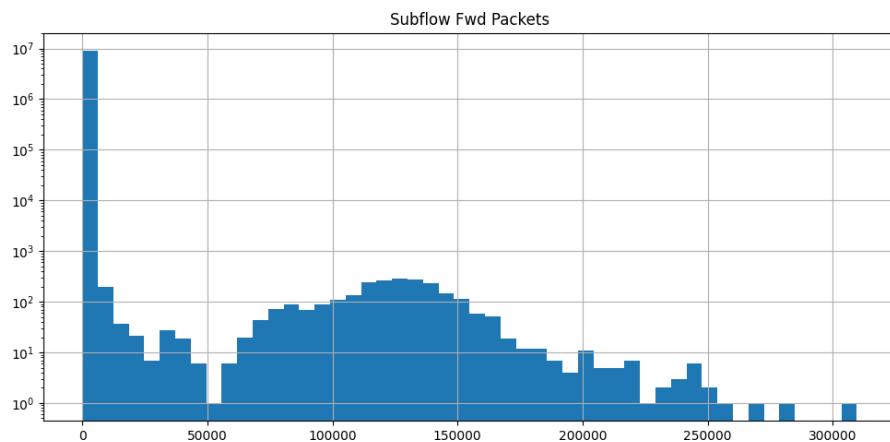


Figure 47: Histogram of Subflow Fwd Packets plotted on log scale

Peak was observed at Subflow Fwd Packets=0.

After the peak, there is significant decline in results up to Subflow Fwd Packets=50000.

There is a plateau region observed between Subflow Fwd Packets \geq 100000 and Subflow Fwd Packets \leq 150000.

There were decline in the number of results observed after Subflow Fwd Packets \geq 150000.

There are many values between Subflow Fwd Packets \geq 50000 and Subflow Fwd Packets \leq 150000.

There is a value after Subflow Fwd Packets $>$ 300000. This may indicate outlier in the data.

Subflow Fwd Bytes: Subflow forward bytes

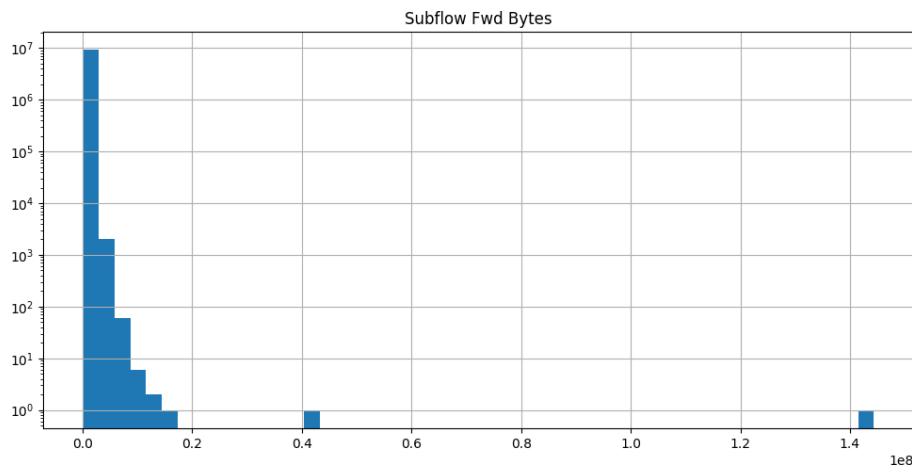


Figure 48: Histogram of Subflow Fwd Bytes plotted on log scale

The distribution is J-shaped graph.

Most of the values are stacked at left end and then it continuously declines as we move towards right hand side of the x-axis.

Most values are concentrated between Subflow Fwd Bytes \geq 0 and Subflow Fwd Bytes <0.2 .

There were couple of values observed around Subflow Fwd Bytes=0.4 and Subflow Fwd Bytes>1.4. This may indicate outlier in the data.

Subflow Bwd Packets: Subflow backward packets

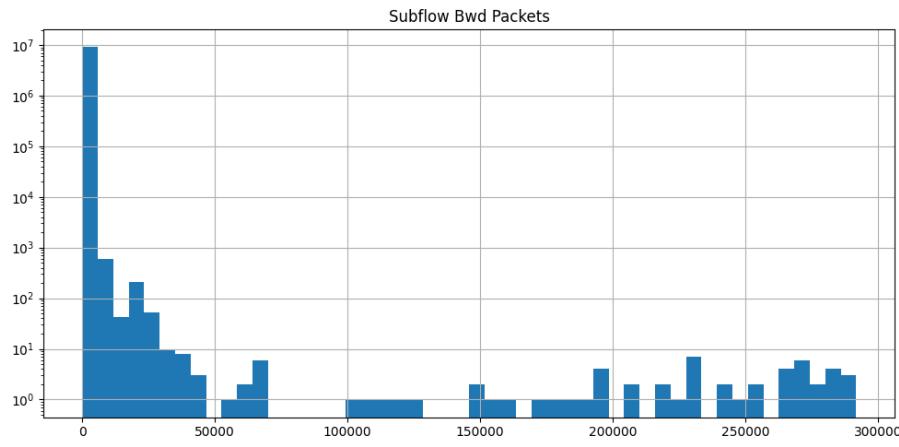


Figure 49: Histogram of Subflow Bwd Packets plotted on log scale

The distribution is skewed towards right: Positively skewed.

Peak was observed at Subflow Bwd Packets=0.

After the peak, there is consistent decline in results.

Most values are concentrated between Subflow Bwd Packets ≥ 0 and Subflow Bwd Packets ≤ 50000 .

After Subflow Bwd Packets >50000 , there are many small plateau regions at irregular gaps up to Subflow Bwd Packets <300000 .

Subflow Bwd Bytes: Subflow backward bytes

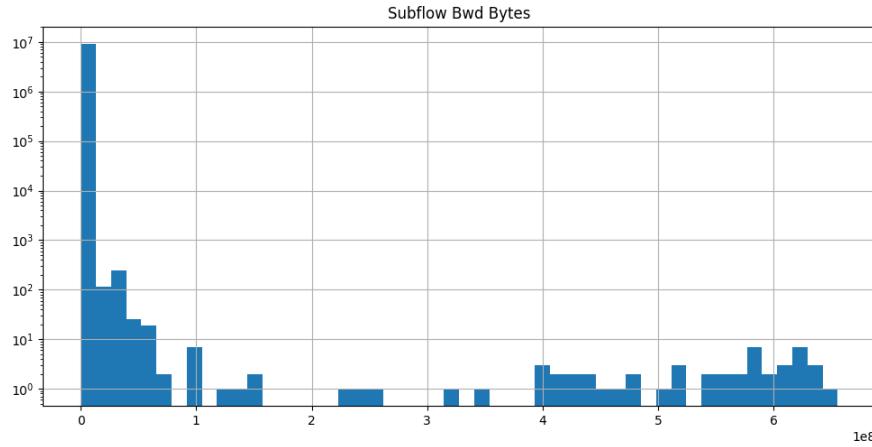


Figure 50: Histogram of Subflow Bwd Bytes plotted on log scale

The distribution is skewed towards right: Positively skewed.

Peak was observed at Subflow Bwd Bytes=0.

Between Subflow Bwd Bytes ≥ 0 and Subflow Bwd Bytes ≤ 1 , the graph appeared similar to J-shaped graph.

Most values are concentrated between Subflow Bwd Bytes ≥ 0 and Subflow Bwd Bytes ≤ 1 .

There are some plateau regions on right hand side of the peak at irregular gaps.

On the X-axis values lie in the range 0 to 7.

Init Fwd Win Bytes: Initial forward window size

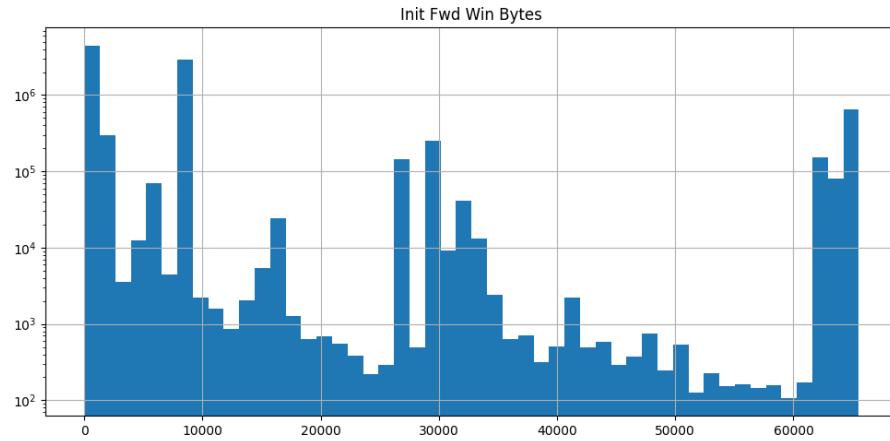


Figure 51: Histogram of Init Fwd Win Bytes plotted on log scale

There are two large peaks at Init Fwd Win Bytes=0 and Init Fwd Win Bytes=10000.

There are smaller peaks at Init Fwd Win Bytes=30000 and Init Fwd Win Bytes>60000.

Between the peaks, the frequency of bins is relatively very less.

There are no gaps in the results observed on X-axis of the graph.

Since the graph has multiple peaks, we can also call it multi-modal.

From broad overview, as we move from left to right hand side of the graph, the results decrease. But, due to tall peaks observed in between, we cannot conclude consistent decline of results.

Init Bwd Win Bytes: Initial backward window size

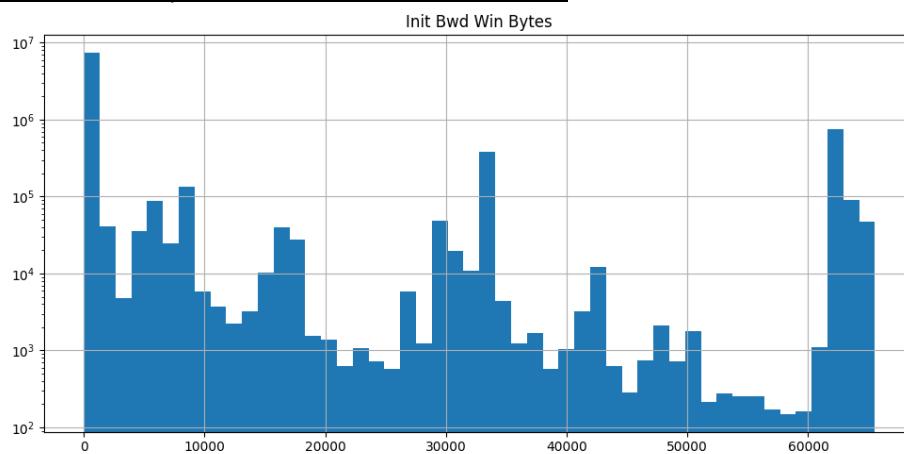


Figure 52: Histogram of Init Bwd Win Bytes plotted on log scale

There are three main peaks from overall observation of the graph: Init Bwd Win Bytes=0, 30000, 60000.

The tallest peak was observed at Init Bwd Win Bytes=0, the second tallest was at Init Bwd Win Bytes=60000 and the smallest peak among the three was observed at Init Bwd Win Bytes=30000.

Between the peaks, the frequency of bins is relatively very less.

There are no gaps in the results observed on X-axis of the graph.

Since the graph has multiple peaks, we can also call it multi-modal.

Fwd Act Data Packets: Forward packets with actual data

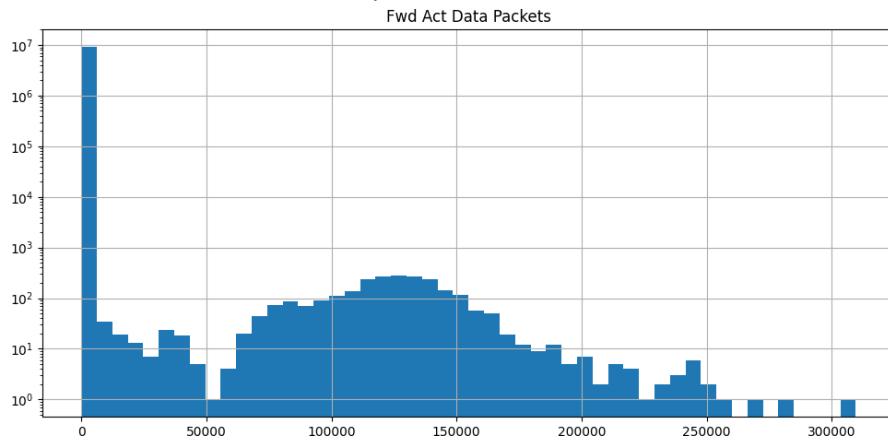


Figure 53: Histogram of Fwd Act Data Packets plotted on log scale

Peak was observed at Fwd Act Data Packets=0.

After the peak, there is significant decline in results up to Fwd Act Data Packets=50000.

There is a plateau region observed between Fwd Act Data Packets \geq 100000 and Fwd Act Data Packets \leq 150000.

There are some values observed after Fwd Act Data Packets>300000. This may indicate outlier in the data.

Fwd Seg Size Min: Minimum segment size in forward packets

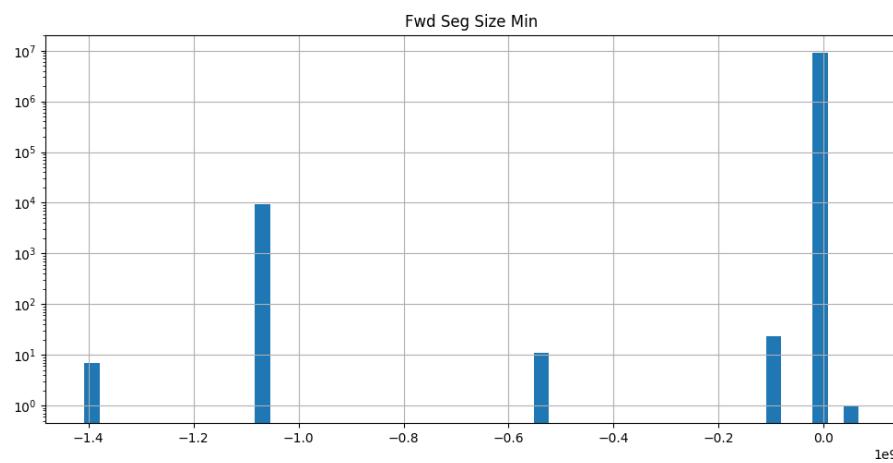


Figure 54: Histogram of Fwd Seg Size Min plotted on log scale

Peak was observed at Fwd Seg Size Min=0.0

On the X-axis value lie in the range -1.4 to 0.0. Thus, the values on X-axis are all negative, we need to check the actual values under the column to determine if data is accurate or invalid.

There are some values observed at Fwd Seg Size Min=-1.4, Fwd Seg Size Min>-1.2 and Fwd Seg Size Min<-1.0, Fwd Seg Size Min>-0.6 and Fwd Seg Size Min<-0.4

Active Mean: Mean active time

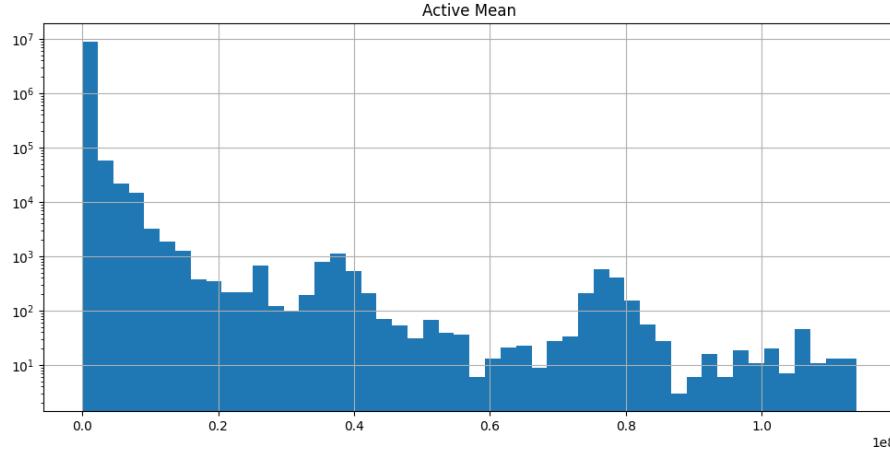


Figure 55: Histogram of Active Mean plotted on log scale

The distribution is skewed towards right: Positively skewed.

Peak was observed at Active Mean=0.

After the peak, there is significant decline in results.

There are two plateau regions observed at Active Mean=0.4 and Active Mean=0.6

There are no gaps in the results observed on X-axis of the graph.

Active Std: Standard deviation of active time

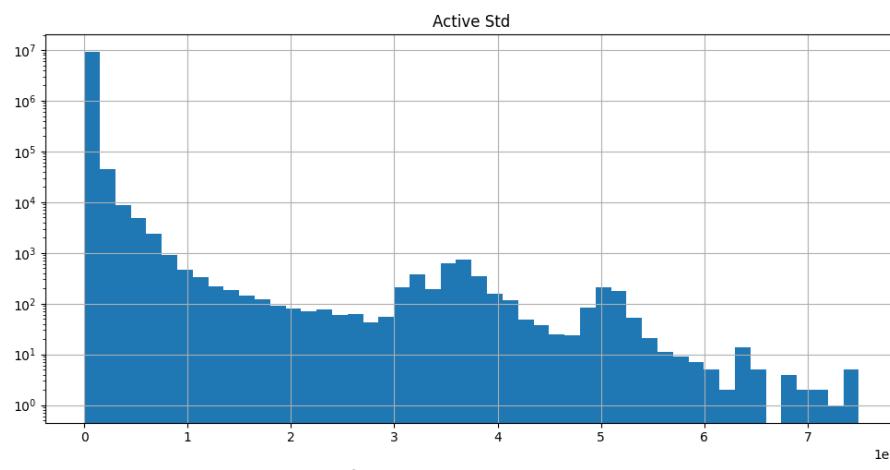


Figure 56: Histogram of Active Std plotted on log scale

The distribution is skewed towards right: Positively skewed.

Peak was observed at Active Std=0.

After the peak, there is consistent decline in results up to Active Std=3.

There are two plateau regions observed between Active Std>=3 and Active Std<=4.

There is decline in the results between Active Std>=4 and Active Std<=5,

There is second plateau in the graph observed near Active Std=5.

There is decline in the results after Active Std>5.

Active Max: Maximum active time

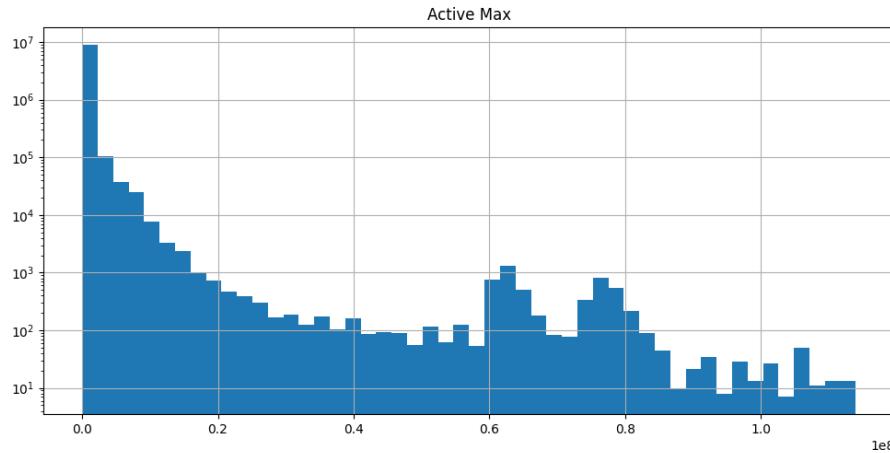


Figure 57: Histogram of Active Max plotted on log scale

The distribution is skewed towards right: Positively skewed.

Peak was observed at Active Max=0.

After the peak, there is consistent decline in results up to Active Max=0.6

Around Active Max=0.6, there is a relatively smaller peak compared to main peak, and a plateau region of 2 bins around it.

Similarly, around Active Max=0.8, there is a relatively smaller peak compared to main peak, and a plateau region of 2 bins around it.

On X-axis values lie in the range 0 to 1.2

There are no gaps in the results observed on X-axis of the graph.

Active Min: Minimum active time

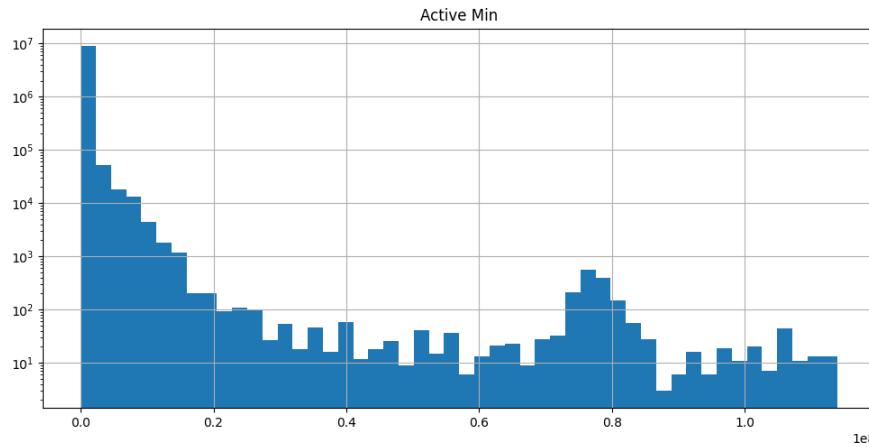


Figure 58: Histogram of Active Min plotted on log scale

The distribution is skewed towards right: Positively skewed.

Peak was observed at Active Min=0.

There is relatively smaller peak at Active Min=0.8 and a plateau region around it.

On X-axis value lie in the range 0 to 1.2

There are no gaps in the results observed on X-axis of the graph.

Idle Mean: Mean idle time

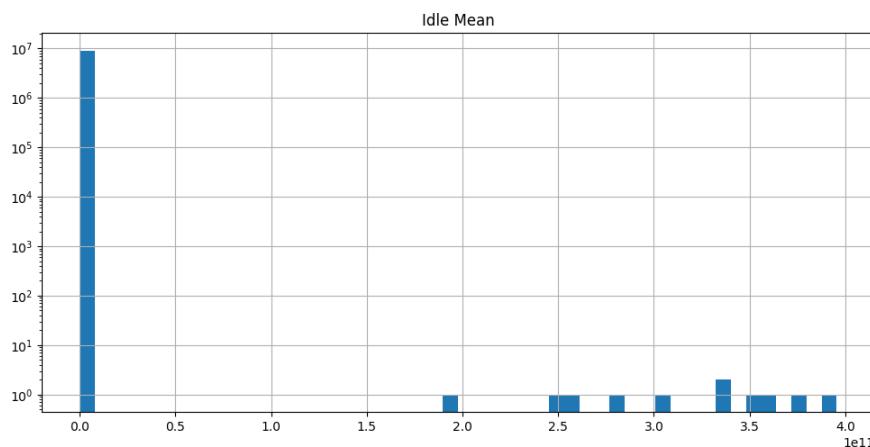


Figure 59: Histogram of Idle Mean plotted on log scale

Peak was observed at Idle Mean=0.

Most values are concentrated in bin represented by the peak.

There are some values observed at Idle Mean=2.0, 3.0, 3.5, 4.0

There are large gaps observed on X-axis of the graph after the peak.

Idle Std: Standard deviation of idle time

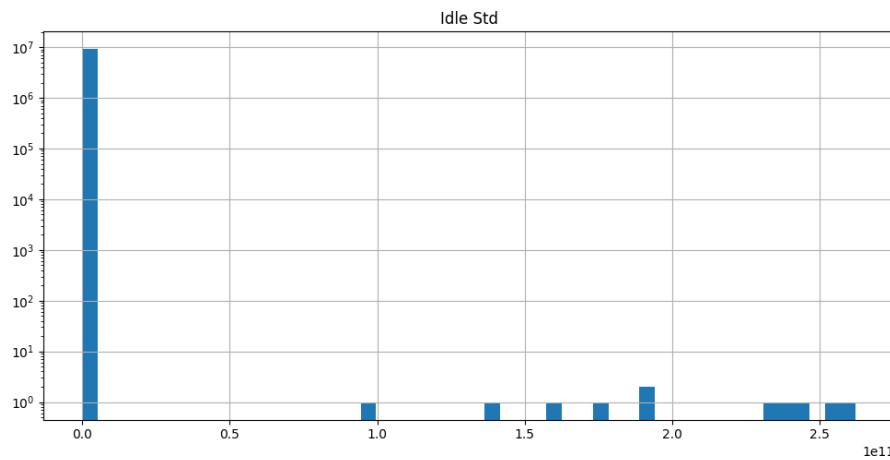


Figure 60: Histogram of Idle Std plotted on log scale

Peak was observed at Idle Std=0.0

Most values are concentrated in bin represented by the peak.

There are some values observed at Idle Std=1.0, 1.5, 2.0 and 2.5

There are large gaps observed on X-axis of the graph after the peak.

Idle Max: Maximum idle time

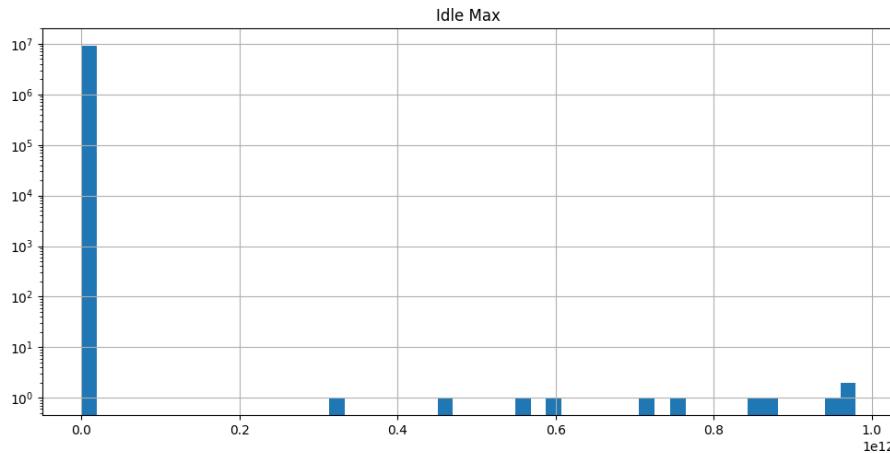


Figure 61: Histogram of Idle Max plotted on log scale

Peak was observed at Idle Max=0.0

Most values are concentrated in bin represented by the peak.

There are some values observed at Idle Max=0.4, 0.6, 0.8, 1.0.

There are large gaps observed on X-axis of the graph after the peak.

Idle Min: Minimum idle time

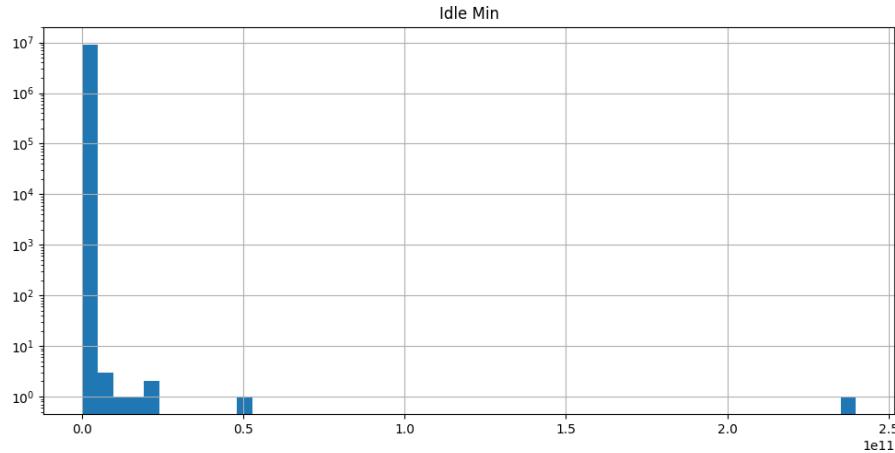


Figure 62: Histogram of Idle Min plotted on log scale

Peak was observed at Idle Min=0.0

Most values are concentrated in bin represented by the peak.

There is a value observed after at Idle Min=2.5, which is after a large gap on X-axis. This may indicate outlier in the data.

Distribution of target features using Bar chart: -

Bar chart for all category of records under target feature: Label and ClassLabel were plotted.

The bar charts strongly indicated the imbalanced nature of the dataset in the direction of Benign records.

The dataset has 78% records classified as Benign and 22% records classified as Malicious.

Thus, it can be classified under Long-Tailed distribution, because lesser category of records (Benign events) has highest frequency, and more category of records (Malicious events) have lower frequency in the dataset.

As the result, we will need to address these issues while training the model to avoid bias for classifying an unknown event as Benign and also reasonably distinguish a Malicious event from Benign event.

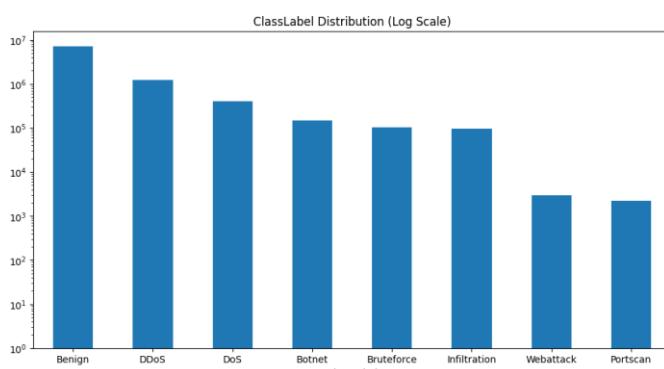


Figure 63: Bar chart of ClassLabel plotted on log scale

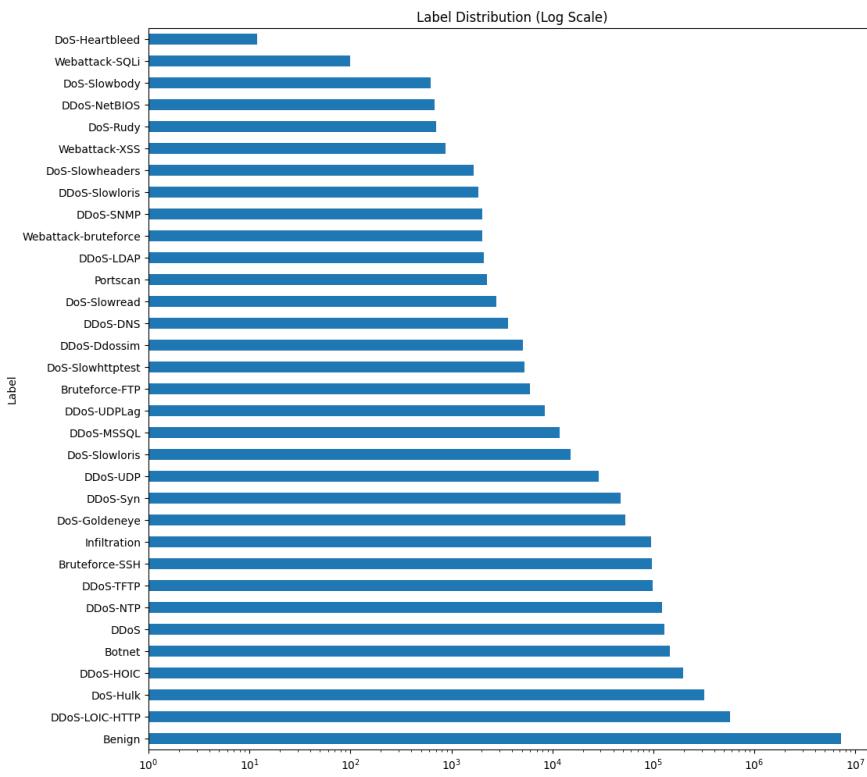


Figure 64: Bar chart of Label plotted on log scale

Analysing and handling negative values: -

Among the features where the negative values were observed, the proportion of negative values were computed.

List of features where negative values were observed are: -

1. Flow Duration = 0.001047%
2. Flow Bytes/s = 0.000578%
3. Flow Packets/s = 0.001047%
4. Flow IAT Mean = 0.001047%
5. Flow IAT Max = 0.000927%
6. Flow IAT Min = 0.030718%
7. Fwd IAT Total = 0.000153%
8. Fwd IAT Mean = 0.000153%
9. Fwd IAT Max = 0.000033%
10. Fwd IAT Min = 0.000349%
11. Fwd Header Length = 0.555312%
12. Bwd Header Length = 0.002803%
13. Init Fwd Win Bytes = 29.002950%
14. Init Bwd Win Bytes = 41.084015%
15. Fwd Seg Size Min = 0.808768%

Among the above 15 features, 2 features have relatively higher percentage of negative values: -

- Init Fwd Win Bytes: 29%
- Init Bwd Win Bytes: 41%

For remaining 13 features, the negative values were imputed with respective median value. This led to increase in concentration of values among those 13 features in their mid-range (at median), but percentage of negative values per feature is less than 1%, thus, the impact was very small.

If the rows having negative values for ‘Init Fwd Win Bytes’ and ‘Init Bwd Win Bytes’, we will lose massive volume of information for all features.

If the two columns were dropped, then the valid datapoints from those two columns will also be lost which may later play important role.

Init Fwd Win Bytes: - Among the negative values, 88% records are Benign and 12% records are Malicious.

Init Bwd Win Bytes: - Among the negative values, 78% records are Benign and 22% records are Malicious.

Thus, the negative values for the two features do not give any different characteristic of events when compared with characteristics of the complete dataset.

As the result, it indicates data quality issues.

We can perform prediction of data points by taking negative values as the unknown data and positive values as training and test data to build a regression model. But, due to time constraints, this approach was not adopted.

As the result, imputation with respective median values were performed against negative values. This led to large hump of values at median, thus the concentration of values around mid-range has increased.

Defining a new feature in the dataset: -

A new feature was defined and added in the dataset: isMalicious, this will enable later to perform binary classification and differentiate between Malicious and Benign events.

Definition of isMalicious: -

If ClassLabel=Benign -> isMalicious=0

If ClassLabel!=Benign -> isMalicious=1

Thus, number of records with isMalicious=0 : 7185881 and isMalicious=1 : 1981390

Dropping an existing feature: -

‘Label’ was dropped from the dataset because it gives further sub-type of the attack, which will not be in scope of the project.

As the result, at this stage, the two target features in the dataset are: -

isMalicious: For binary classification

ClassLabel: For multi-class classification

Handling large size dataset: -

The dataset was too large to carryout analysis and make updates as required, leading to over utilization of system's memory and notebook getting stalled.

Thus, a sample of the dataset was taken by taking 20% of records as the sample size.

Number of records with isMalicious=0 : 1437467 and isMalicious=1 : 395987

It was observed that the sampled dataset has similar imbalanced nature as the original dataset. And all the categories under ClassLabel are observed in the same as original dataset with similar proportions.

Analysing and handling outliers: -

The count and percentage of outliers in each independent feature of sampled dataset were computed.

Definition of outlier: - Given a datapoint x , if $x < (Q1 - 1.5 * IQR)$ or $x > (Q3 + 1.5 * IQR)$, then x is an outlier.

Where $Q1 = 25^{\text{th}}$ percentile

$Q3 = 75^{\text{th}}$ percentile

$IQR = \text{Inter-Quartile Range} = Q3 - Q1$

Table 5: Count and percentage of outliers for each feature

Field name	Number of outliers	Percentage of outliers
Flow Duration	362139	19.75
Total Fwd Packets	167485	9.13
Total Backward Packets	176328	9.62
Fwd Packets Length Total	71763	3.91
Bwd Packets Length Total	265389	14.47
Fwd Packet Length Max	24476	1.33
Fwd Packet Length Mean	74245	4.05
Fwd Packet Length Std	21018	1.15
Bwd Packet Length Max	69888	3.81
Bwd Packet Length Mean	140674	7.67
Bwd Packet Length Std	56299	3.07
Flow Bytes/s	377550	20.59
Flow Packets/s	380170	20.74
Flow IAT Mean	346826	18.92
Flow IAT Std	284585	15.52
Flow IAT Max	255816	13.95
Flow IAT Min	404158	22.04
Fwd IAT Total	352629	19.23
Fwd IAT Mean	355920	19.41
Fwd IAT Std	395445	21.57

Fwd IAT Max	252256	13.76
Fwd IAT Min	420002	22.91
Bwd IAT Total	316674	17.27
Bwd IAT Mean	257401	14.04
Bwd IAT Std	291284	15.89
Bwd IAT Max	266733	14.55
Bwd IAT Min	389592	21.25
Fwd PSH Flags	57754	3.15
Fwd Header Length	127832	6.97
Bwd Header Length	147846	8.06
Fwd Packets/s	379750	20.71
Bwd Packets/s	381136	20.79
Packet Length Max	77407	4.22
Packet Length Mean	179993	9.82
Packet Length Std	68317	3.73
Packet Length Variance	160010	8.73
SYN Flag Count	74451	4.06
URG Flag Count	66218	3.61
Avg Packet Size	177462	9.68
Avg Fwd Segment Size	74245	4.05
Avg Bwd Segment Size	140674	7.67
Subflow Fwd Packets	167485	9.13
Subflow Fwd Bytes	71763	3.91
Subflow Bwd Packets	176328	9.62
Subflow Bwd Bytes	265389	14.47
Init Fwd Win Bytes	719371	39.24
Init Bwd Win Bytes	684281	37.32
Fwd Act Data Packets	122997	6.71
Fwd Seg Size Min	678664	37.02
Active Mean	266956	14.56
Active Std	150242	8.19
Active Max	266956	14.56
Active Min	266956	14.56
Idle Mean	385990	21.05
Idle Std	173591	9.47
Idle Max	385990	21.05
Idle Min	385990	21.05

Among the 57 independent features: -

1. 12 features have outliers whose percentage of difference between Malicious and Benign events is greater than or equal to 10%.
2. Remaining 45 features have nearly equal percentage of outliers labelled as Malicious and Benign.

Out of the 12 features, 4 features have relatively higher percentage of outliers.

1. Init Fwd Win Bytes: 39.24%
2. Init Bwd Win Bytes: 37.32%
3. Fwd Seg Size Min: 37.02%
4. Bwd IAT Mean: 14.04%

Among the above 4 features, there were a greater number of records classified as Benign than Malicious. Thus, the features with relatively higher number of outliers do not indicate any anomaly or provide differentiation to detect Malicious events.

Remaining 8 features have lesser than or equal to 6% of records as outliers.

1. Fwd Packets Length Total
2. Bwd Packet Length Max
3. Bwd Packet Length Std
4. Fwd Header Length
5. Packet Length Max
6. Packet Length Std
7. Avg Fwd Segment Size
8. Subflow Fwd Bytes

All above 8 features have a greater number of outliers classified as Malicious than Benign. Thus, the features with relatively lesser number of outliers help to provide small differentiation of Malicious events over Benign events.

To handle outliers, two approaches were applied and tested: -

1. Winsorization
2. Robust Scaling

The tests were performed on 4 features: -

1. Init Fwd Win Bytes
2. Init Bwd Win Bytes
3. Fwd Seg Size Min
4. Bwd IAT Mean

Winsorization: -

Winsorization was performed by replacing each feature's lower range outliers with the 5th percentile value and higher range outliers with the 95th percentile value.

Histograms for each feature prior and post winsorization were plotted to observe the change in pattern of distribution.

Along with histogram, statistical computations were also done to compare the results for each feature and analyse the impact of the process.

Init Fwd Win Bytes: -

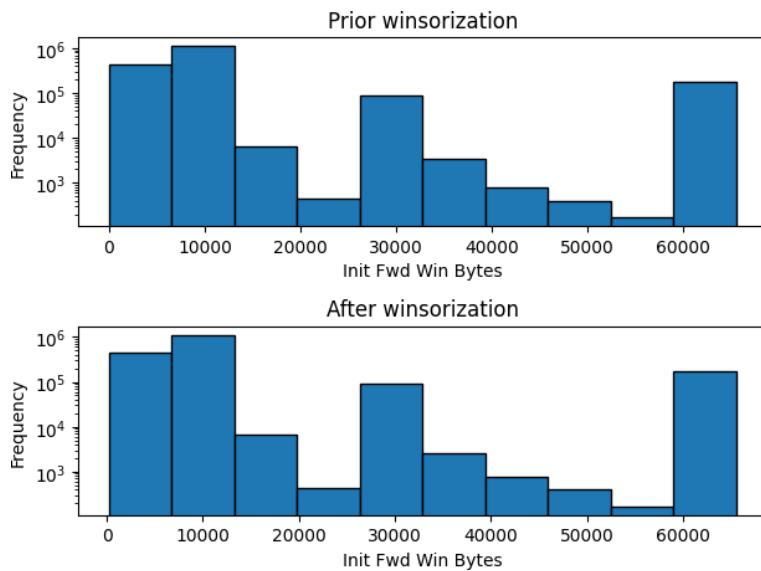


Figure 65: Histogram to compare impact of winsorization on Init Fwd Win Bytes

- The distribution of the feature pre and post winsorization is similar.
- Median value has remained constant=8192.0
- Standard deviation value reduced from 17920 to 17917.

Init Bwd Win Bytes: -

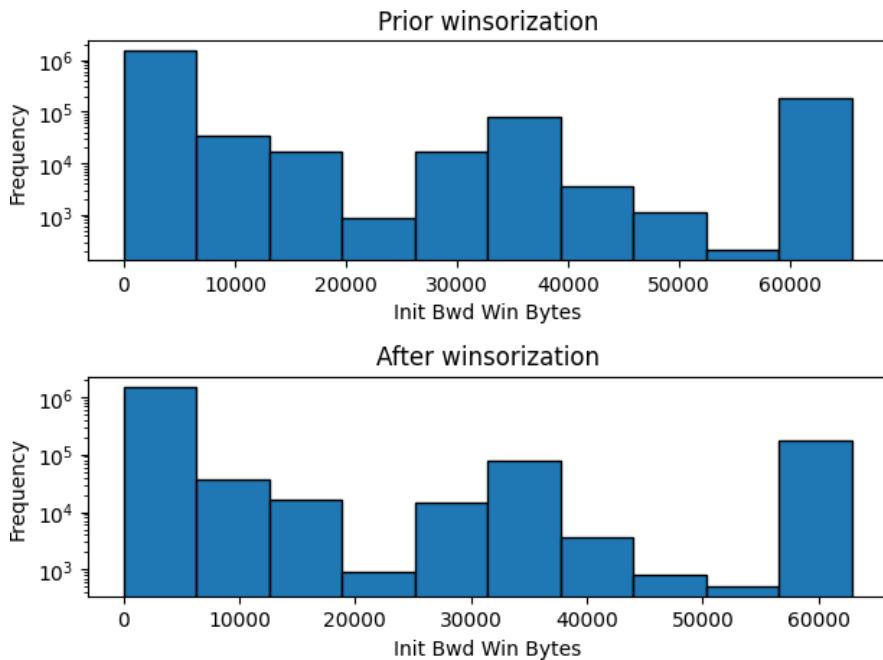


Figure 66: Histogram to compare impact of winsorization on Init Bwd Win Bytes

- The distribution of the feature pre and post winsorization is similar.
- Median value remained constant=235.0
- Standard deviation value reduced from 19414 to 19358.

Fwd Seg Size Min: -

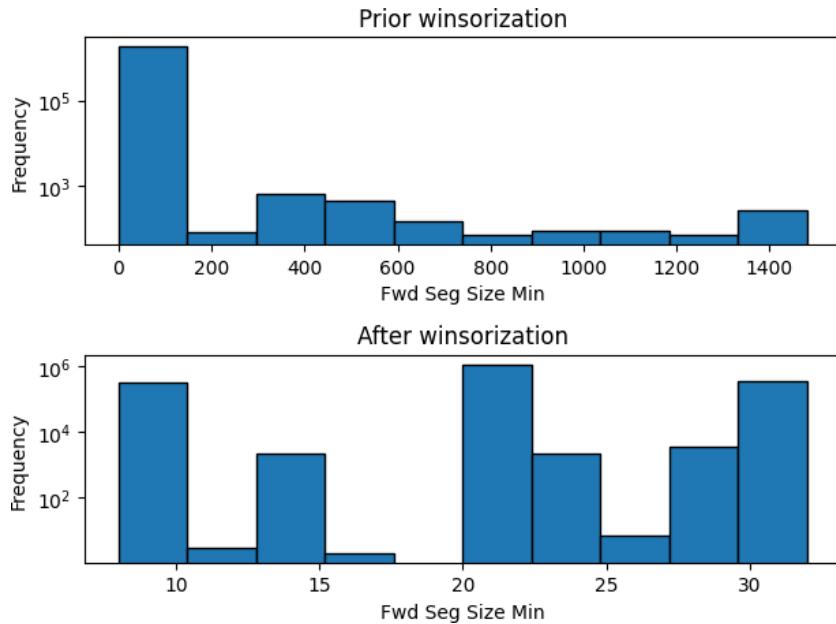


Figure 67: Histogram to compare impact of winsorization on Fwd Seg Size Min

- Based on visual comparison of the two histograms, the distribution of data has changed drastically after winsorization.
- Median value remained constant=20.0
- Standard deviation value reduced from 25.97 to 7.26
- Maximum value prior winsorization was 1480 and after winsorization was 32. Number of values in the sampled dataset prior winsorization between 32 and 1480 = 17305. Thus, many values were impacted due to the process.

Bwd IAT Mean: -

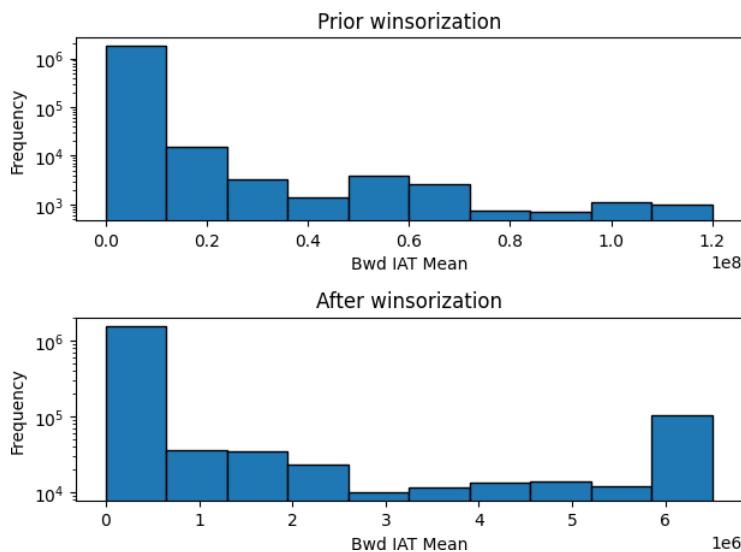


Figure 68: Histogram to compare impact of winsorization on Bwd IAT Mean

- The distribution of data changed after performing winsorization on the feature.
- The main peak on the first bin (left hand side) has remained unchanged.
- There is a new peak observed towards the right hand side of the histogram plotted after winsorization. This may have occurred due to the outlier values that have got replaced by 95th percentile and thus, the frequency of last bin increased.
- Median value remained constant=647.
- Standard deviation value reduced from 6192044.5 to 1657361.2
- Maximum value in the sampled dataset prior winsorization was 120000000.0 and maximum value in the sampled dataset post winsorization was 6501929. Number of values in the sampled dataset prior winsorization between 6501929 and 120000000.0 = 91673. Thus, many values were impacted due to the process.

Robust Scaling: -

Robust Scaling was performed as the second option to handle outliers. If the given data point is x , then its value after Robust Scaling = $x - \text{Median}/\text{IQR}$.

Robust Scaling uses Median and IQR value to transform the data. Median and IQR value are mostly resistant to outliers. Thus, Robust Scaling is also resilient to outliers in data.

Init Fwd Win Bytes: -

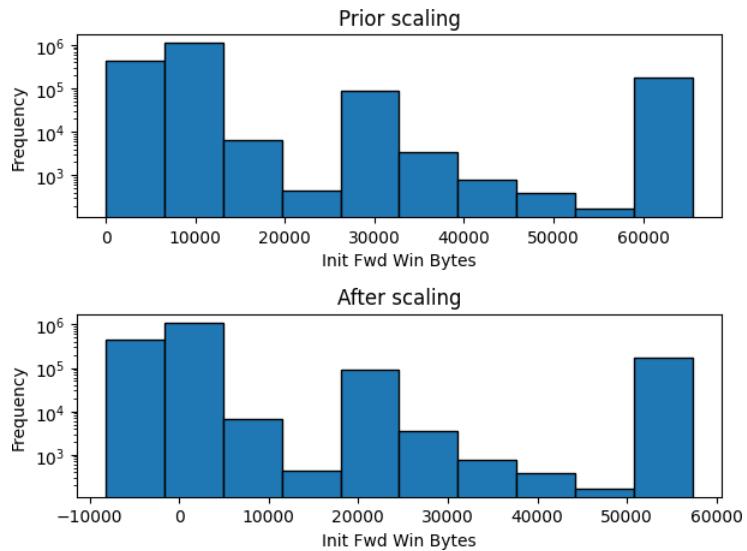


Figure 69: Histogram to compare impact of Robust Scaling on Init Fwd Win Bytes

Init Bwd Win Bytes: -

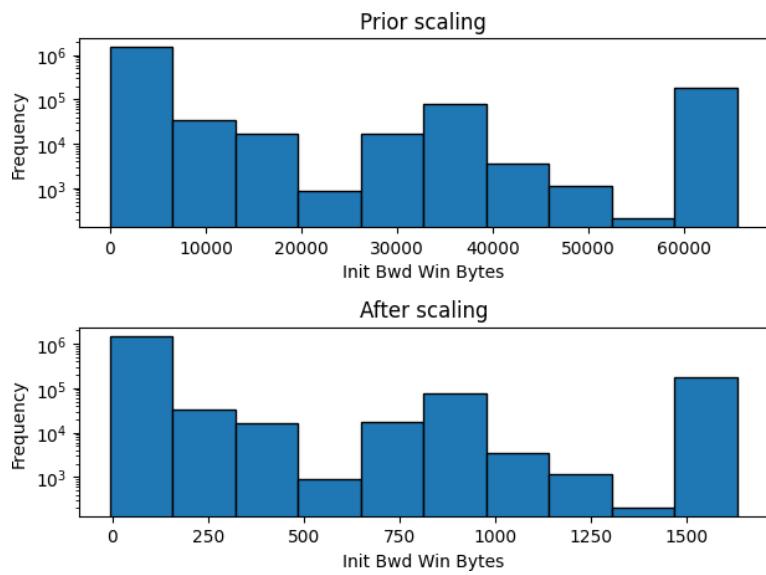


Figure 70: Histogram to compare impact of Robust Scaling on Init Bwd Win Bytes

Fwd Seg Size Min: -

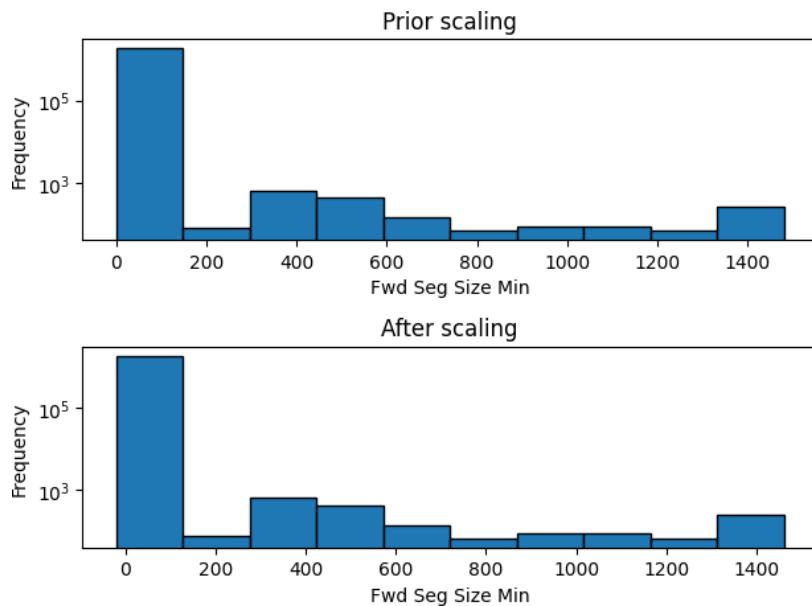


Figure 71: Histogram to compare impact of Robust Scaling on Fwd Seg Size Min

Bwd IAT Mean: -

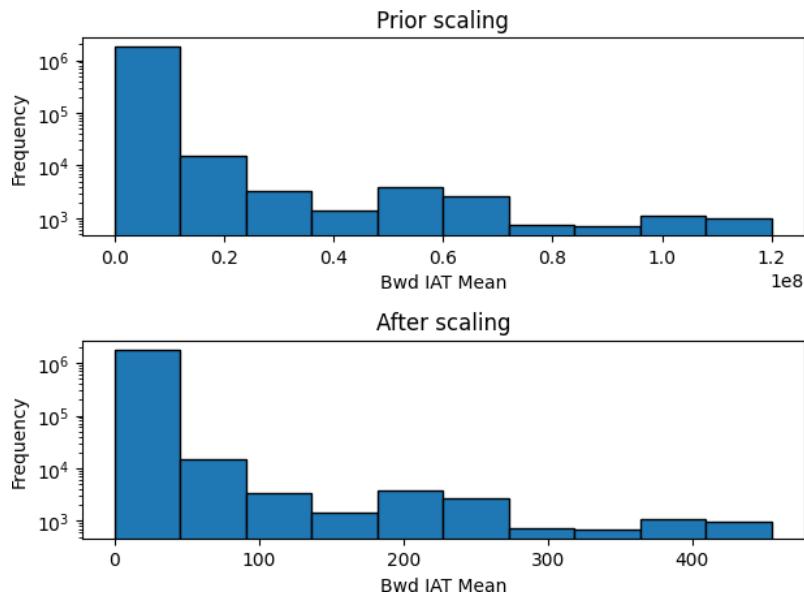


Figure 72: Histogram to compare impact of Robust Scaling on Bwd IAT Mean

After performing Robust Scaling on the four features, it was observed that although the distribution of data remained similar, the values on X-axis changed and it also led to transformation of existing data into negative values.

Summary of the two tests performed for handling outliers: -

- Winsorization impacts large number of values which are outliers and brings them closer to the normal range of data. As the result, the distribution pattern of the features changed by different magnitudes.

- Winsorization helped to reduce the influence of outliers by handling the extreme values in each of the four features.
- Robust Scaling kept the distribution pattern same pre and post operation. However, it led to negative values. This may be due to right skewed nature of the dataset. Since we subtract a datapoint with median, in right skewed dataset, many data points are on the left hand side, that is closer to zero. Thus, subtraction of median from datapoints closer to zero led to generation of negative values.

Approach adopted for handling of outliers: -

- In order to prevent generation of negative values in the dataset, Winsorization was opted for handling outliers among the four features which have relatively large number of outliers.
- Since the four features have a greater number of outliers, they also have higher likelihood of having noisy data. As the result, Winsorization will help to reduce the impact of noise among the four features.
- For the remaining features, outliers were handled by performing imputation with median values. Reason for this approach: -
 - Since the number of outliers among these features were very less, imputing them with respective median value will help to approximate the entries having outliers.
 - Most of the features are skewed, thus, the imputation of outliers was performed with respective median values.

Thus, all outliers among the independent features were handled by combining the approach of Winsorization and Imputation with Median, and mitigated loss of data by preventing deletion of records having outliers.

After handling outliers on the original dataset, again the sample size of 20% was selected as sampled dataset.

Based on the sampled dataset, histograms on log scale were plotted for all independent features, to compare how the distribution of each feature changed prior and after handling of negative values and outliers in the dataset.

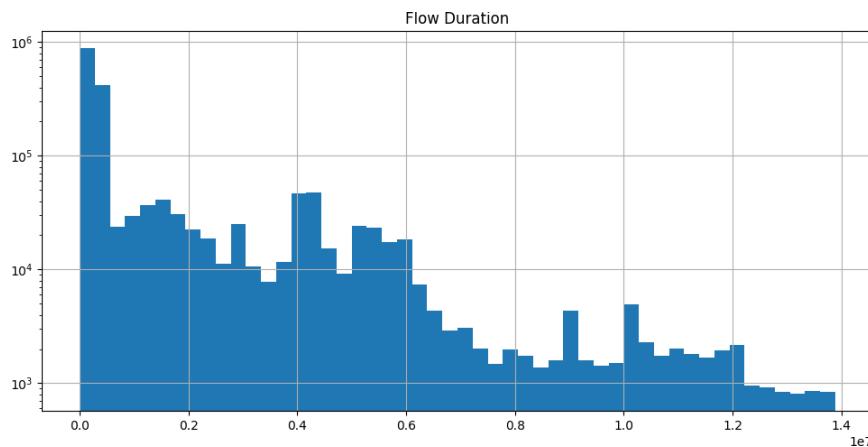


Figure 73: Histogram of Flow Duration plotted on log scale after handling negative values and outliers

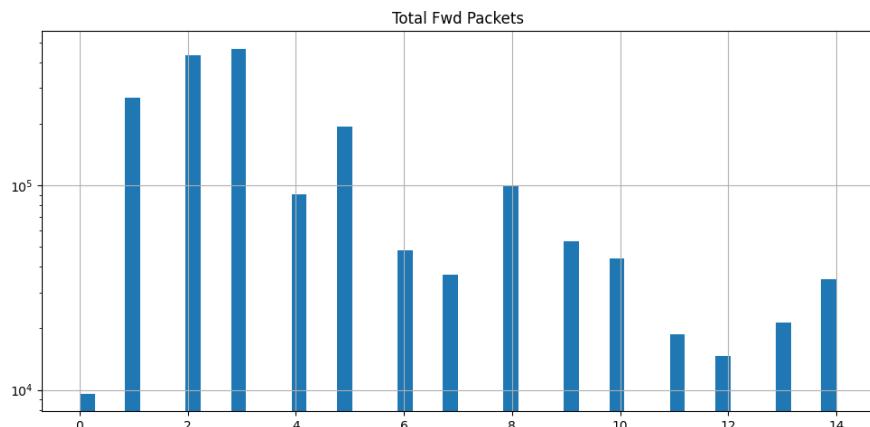


Figure 74: Histogram of Total Fwd Packets plotted on log scale after handling negative values and outliers

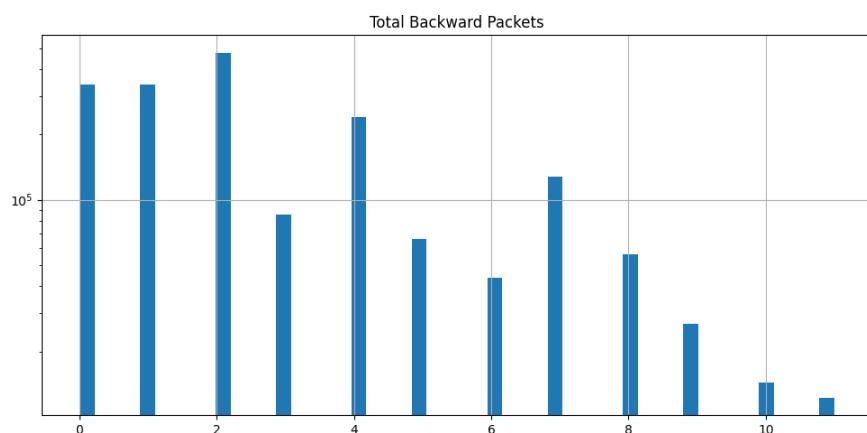


Figure 75: Histogram of Total Backward Packets plotted on log scale after handling negative values and outliers

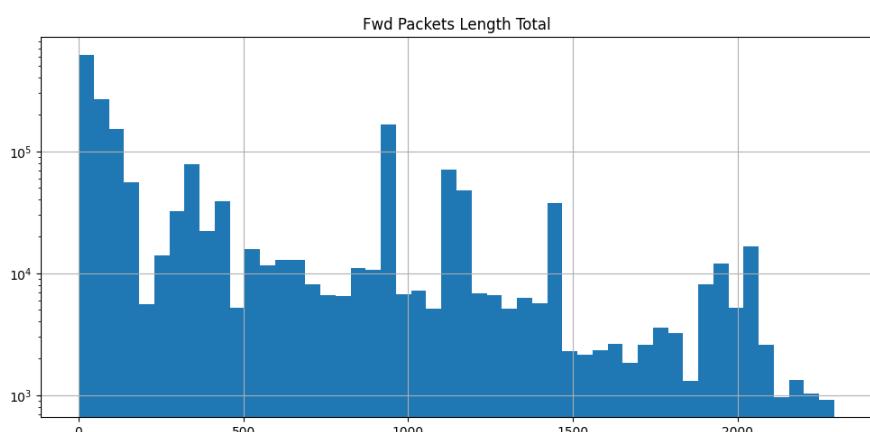


Figure 76: Histogram of Fwd Packets Length Total plotted on log scale after handling negative values and outliers

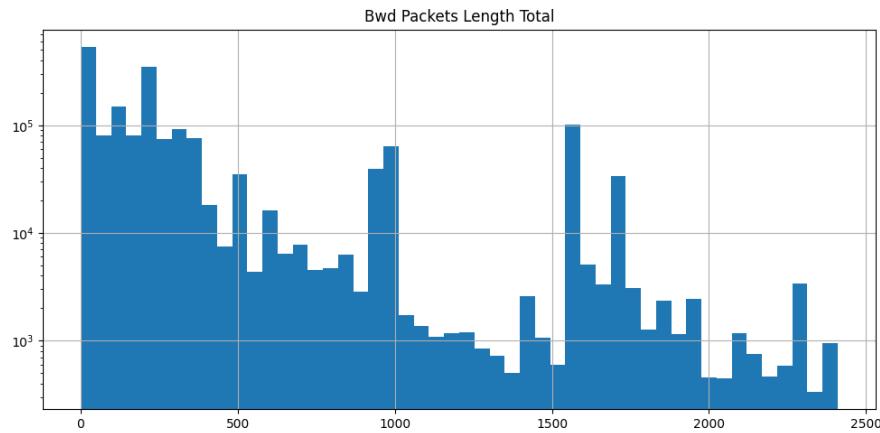


Figure 77: Histogram of Bwd Packets Length Total plotted on log scale after handling negative values and outliers

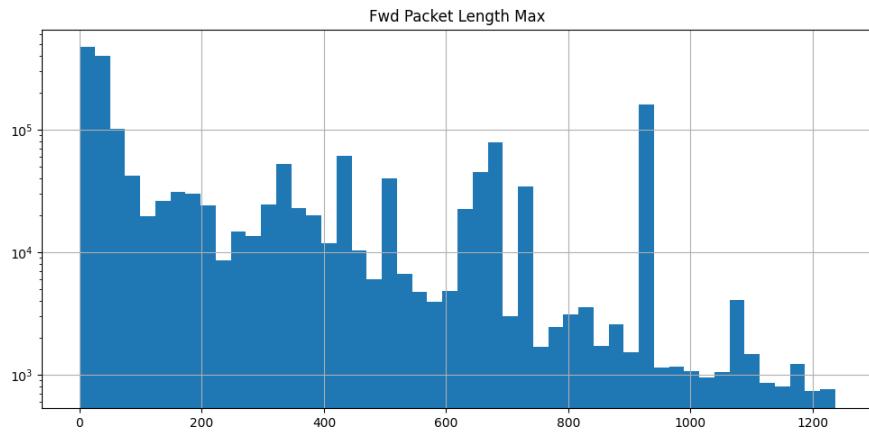


Figure 78: Histogram of Fwd Packet Length Max plotted on log scale after handling negative values and outliers

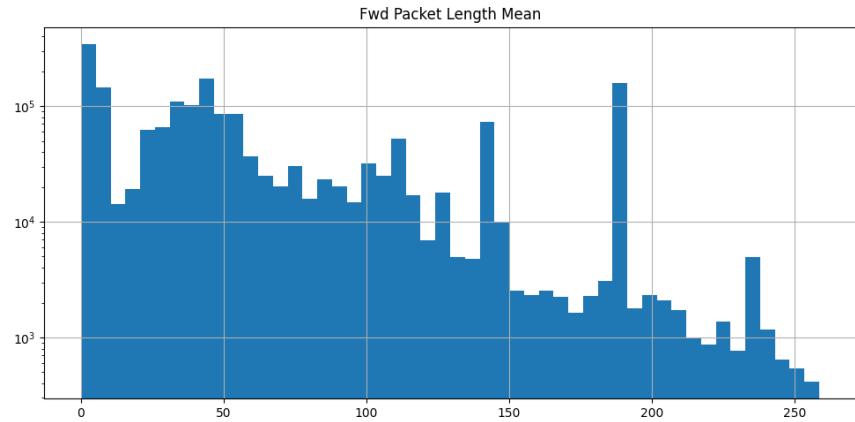


Figure 79: Histogram of Fwd Packet Length Mean plotted on log scale after handling negative values and outliers

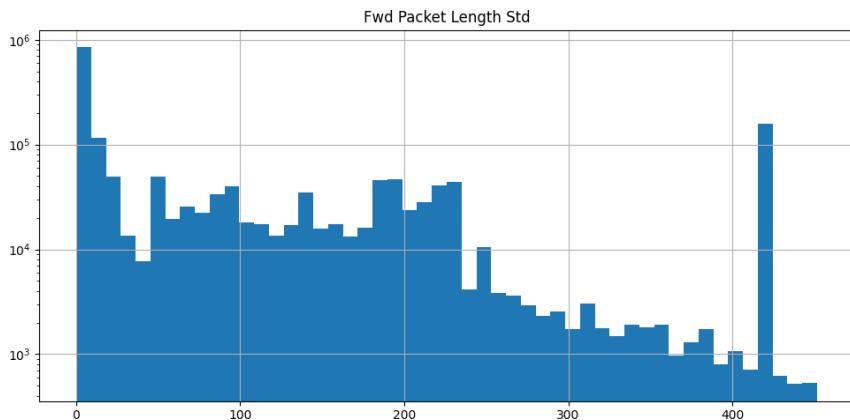


Figure 80: Histogram of Fwd Packet Length Std plotted on log scale after handling negative values and outliers

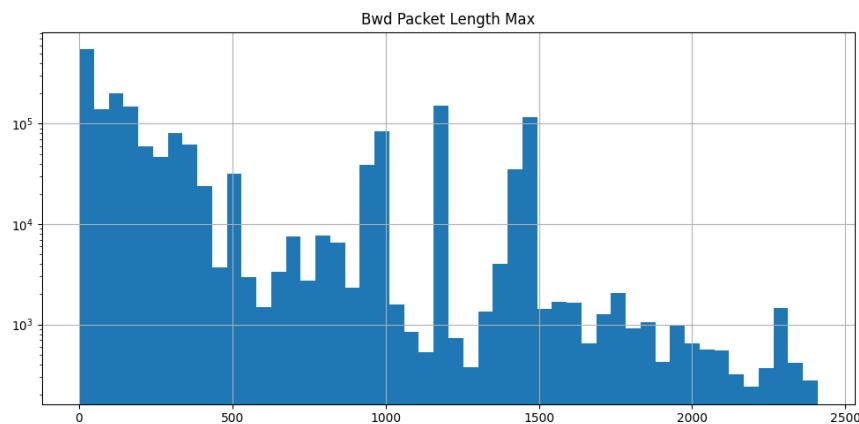


Figure 81: Histogram of Bwd Packet Length Max plotted on log scale after handling negative values and outliers

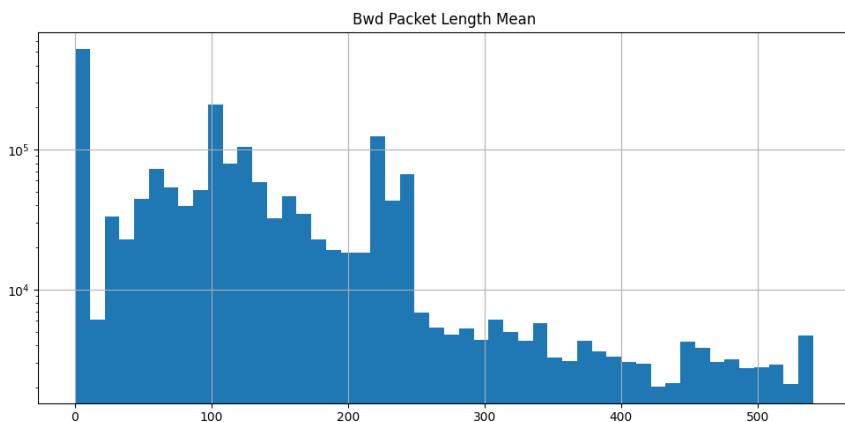


Figure 82: Histogram of Bwd Packet Length Mean plotted on log scale after handling negative values and outliers

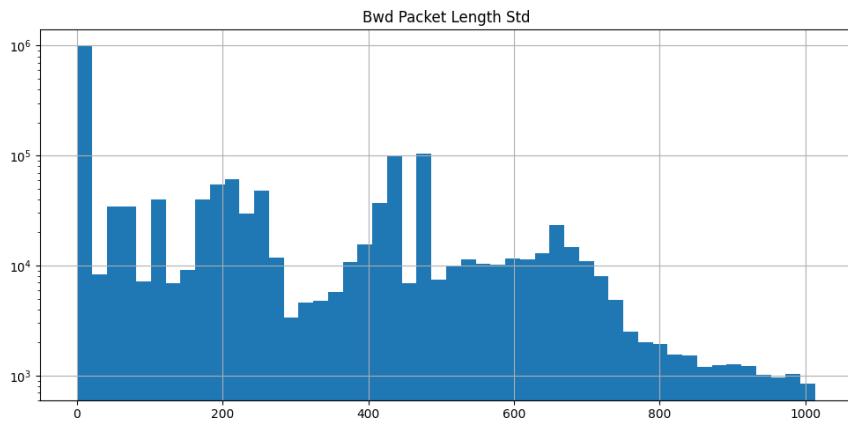


Figure 83: Histogram of Bwd Packet Length Std plotted on log scale after handling negative values and outliers

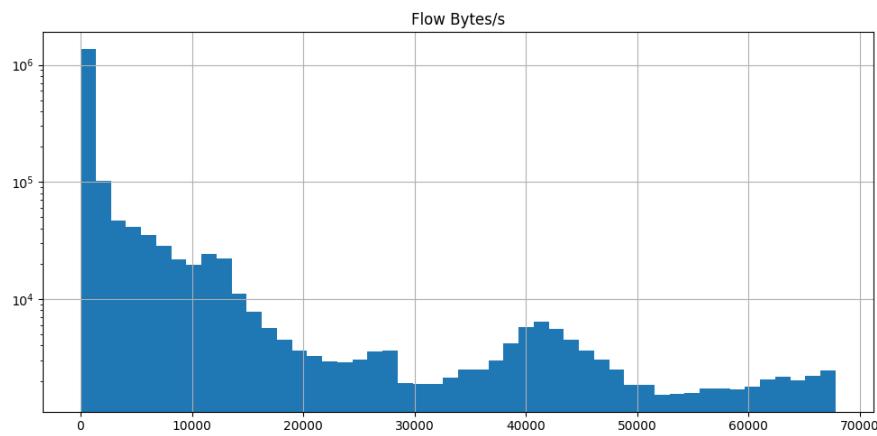


Figure 84: Histogram of Flow Bytes/s plotted on log scale after handling negative values and outliers

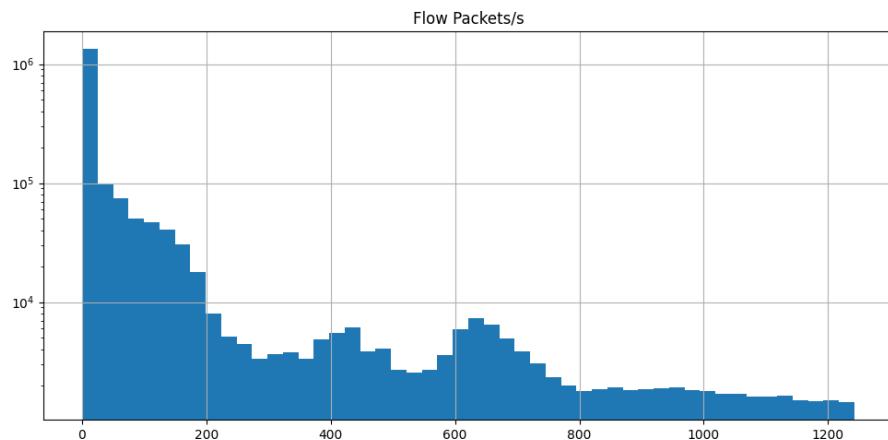


Figure 85: Histogram of Flow Packets/s plotted on log scale after handling negative values and outliers

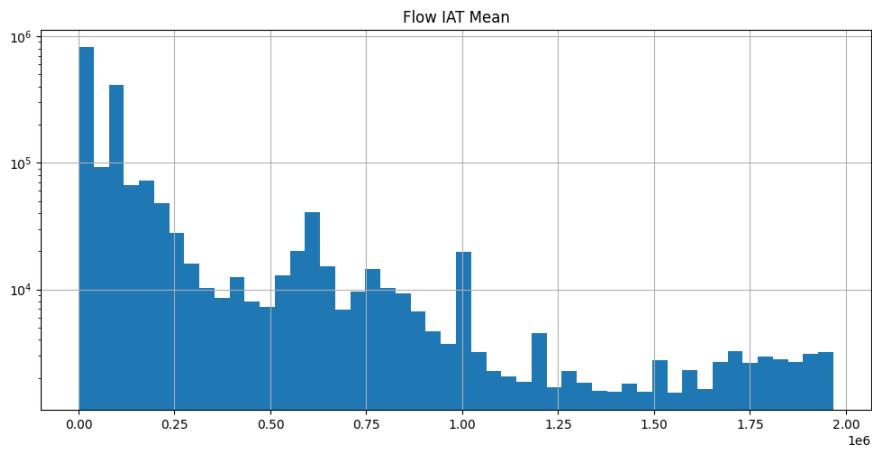


Figure 86: Histogram of Flow IAT Mean plotted on log scale after handling negative values and outliers

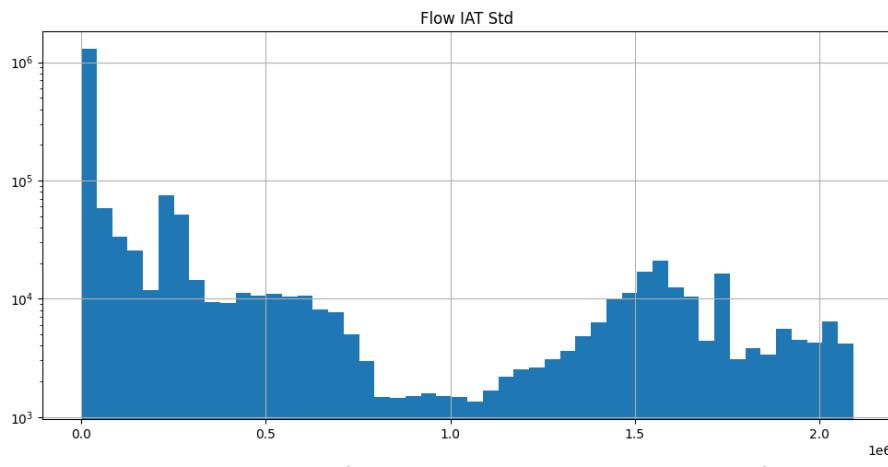


Figure 87: Histogram of Flow IAT Std plotted on log scale after handling negative values and outliers

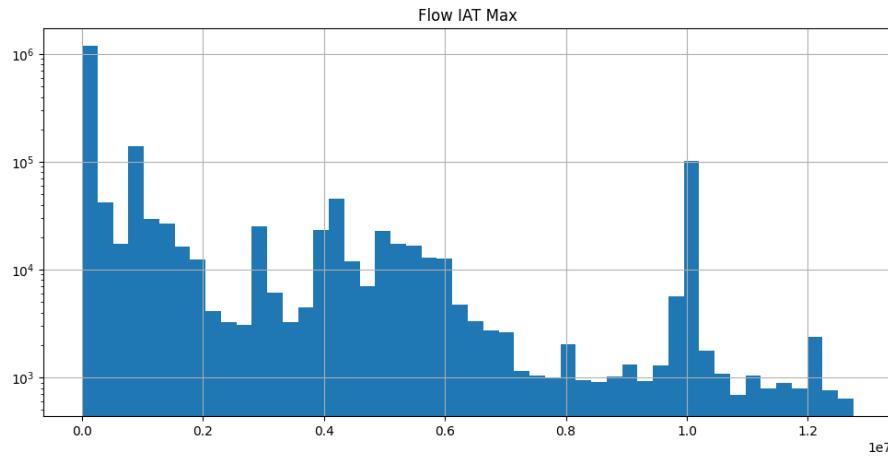


Figure 88: Histogram of Flow IAT Max plotted on log scale after handling negative values and outliers

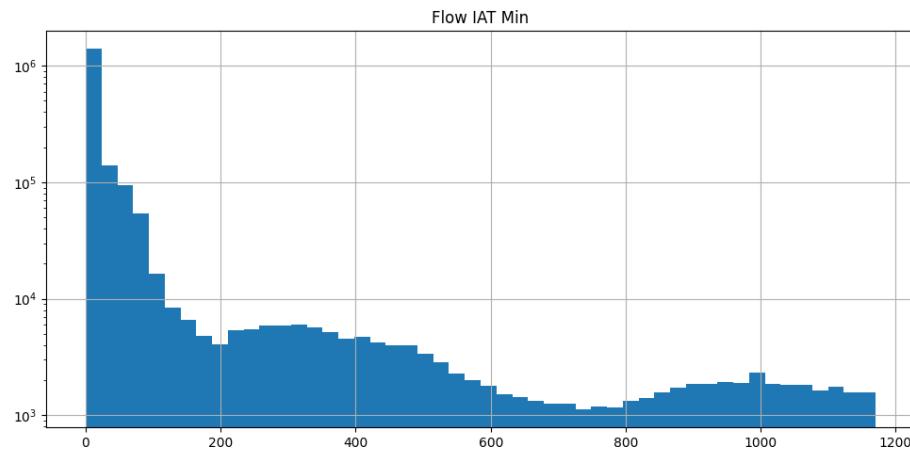


Figure 89: Histogram of Flow IAT Min plotted on log scale after handling negative values and outliers

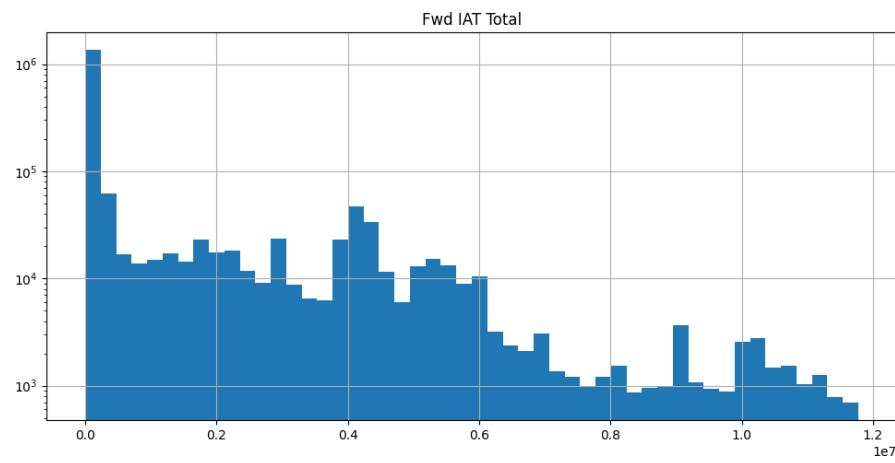


Figure 90: Histogram of Flow IAT Total plotted on log scale after handling negative values and outliers

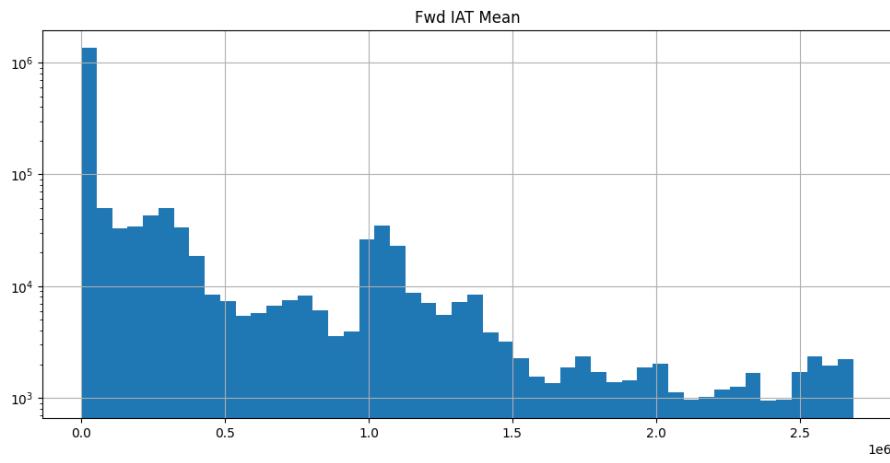


Figure 91: Histogram of Fwd IAT Mean plotted on log scale after handling negative values and outliers

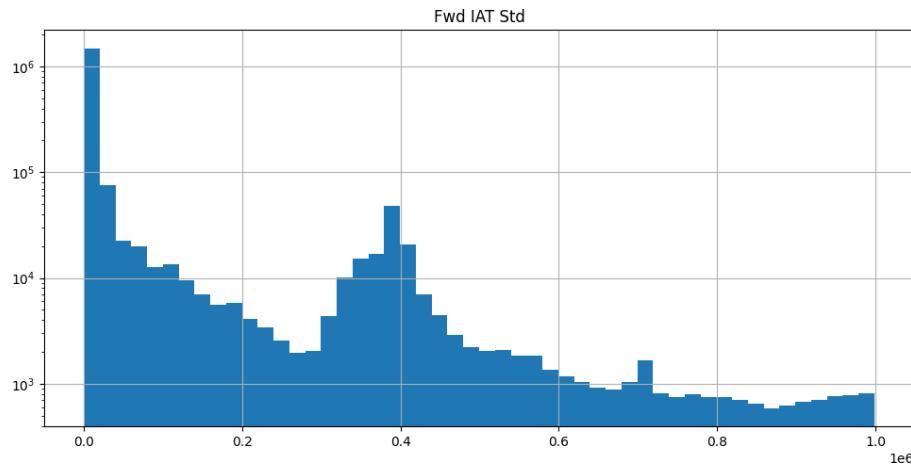


Figure 92: Histogram of Fwd IAT Std plotted on log scale after handling negative values and outliers

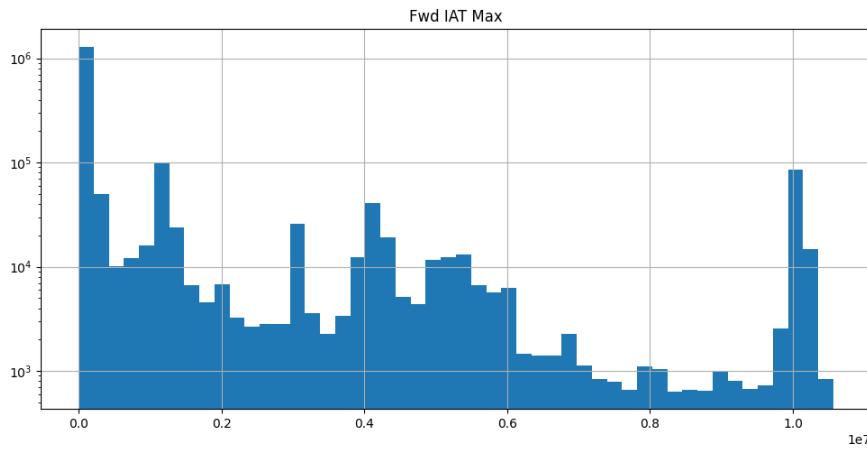


Figure 93: Histogram of Fwd IAT Max plotted on log scale after handling negative values and outliers

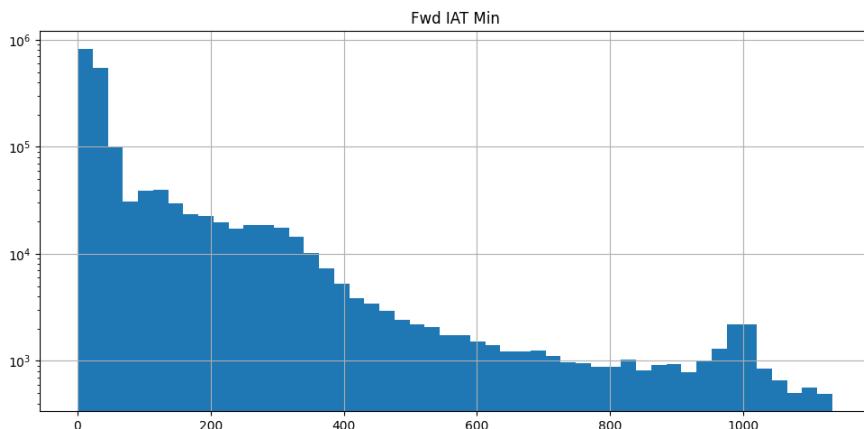


Figure 94: Histogram of Fwd IAT Min plotted on log scale after handling negative values and outliers

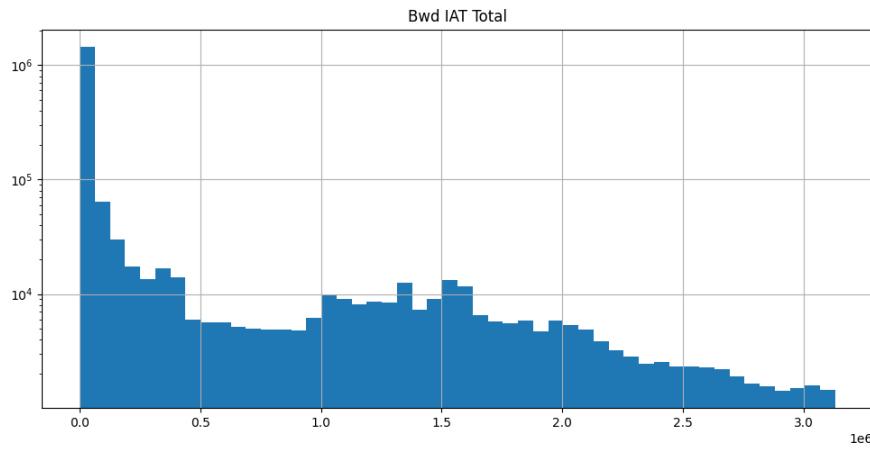


Figure 95: Histogram of Bwd IAT Total plotted on log scale after handling negative values and outliers

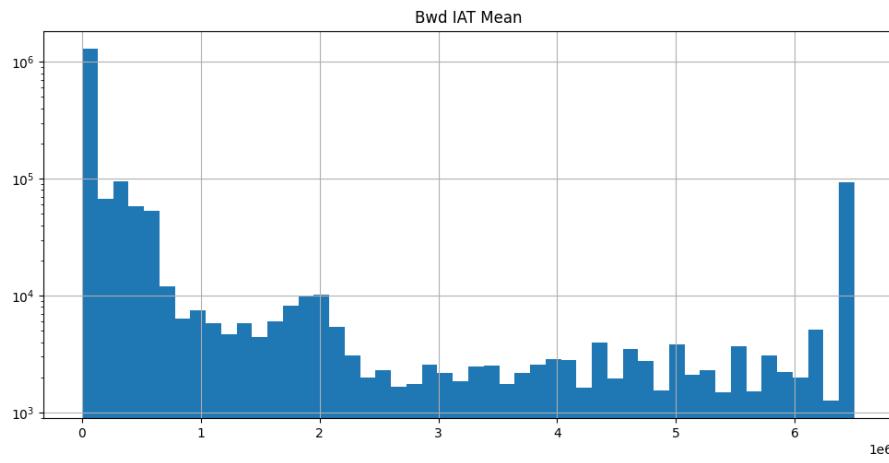


Figure 96: Histogram of Bwd IAT Mean plotted on log scale after handling negative values and outliers

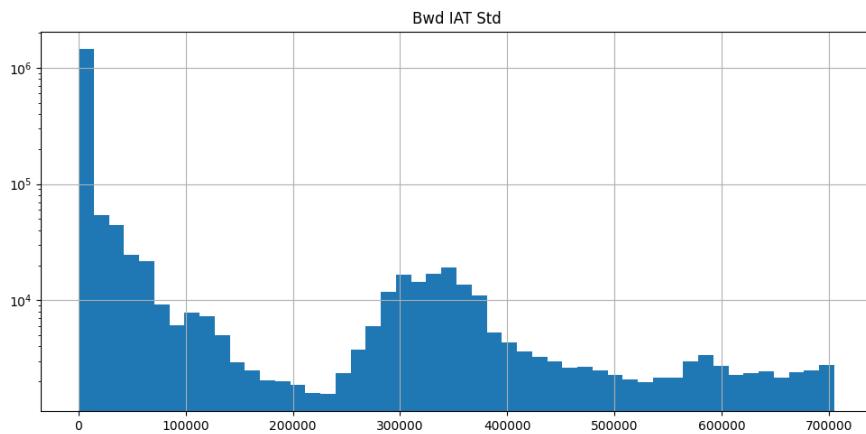


Figure 97: Histogram of Bwd IAT Std plotted on log scale after handling negative values and outliers

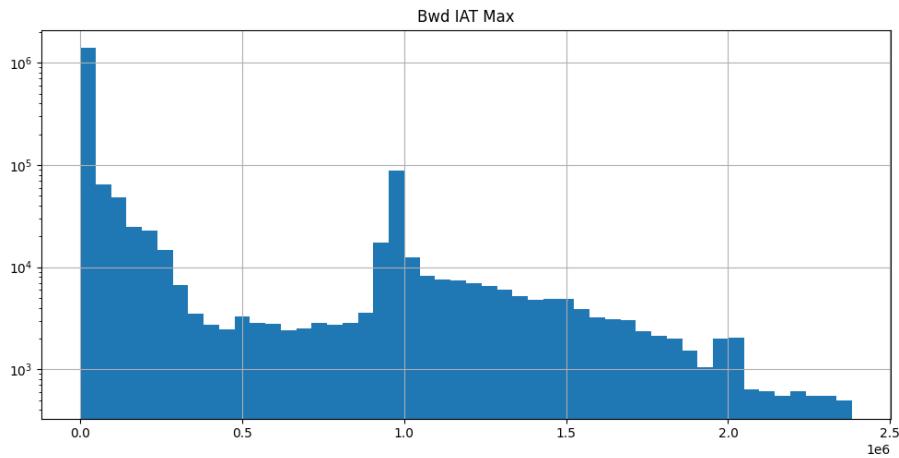


Figure 98: Histogram of Bwd IAT Max plotted on log scale after handling negative values and outliers

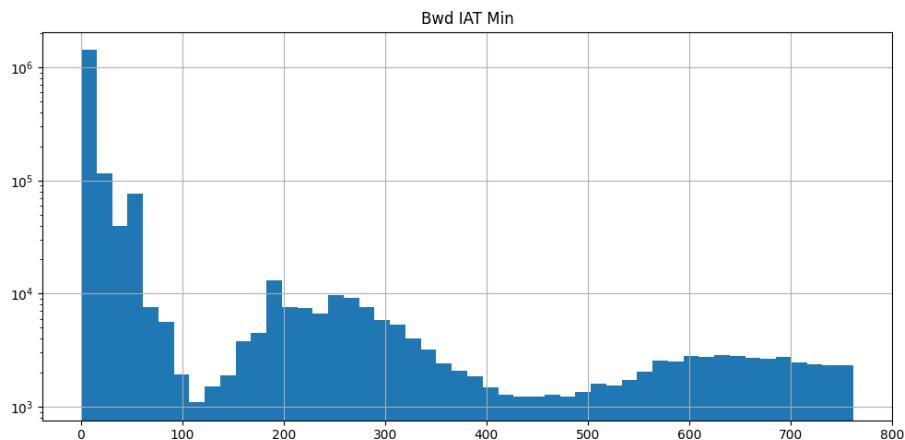


Figure 99: Histogram of Bwd IAT Min plotted on log scale after handling negative values and outliers

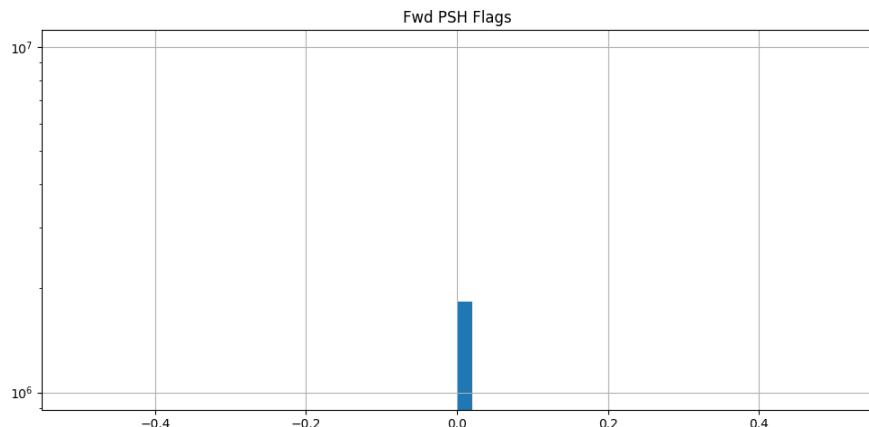


Figure 100: Histogram of Fwd PSH Flags plotted on log scale after handling negative values and outliers

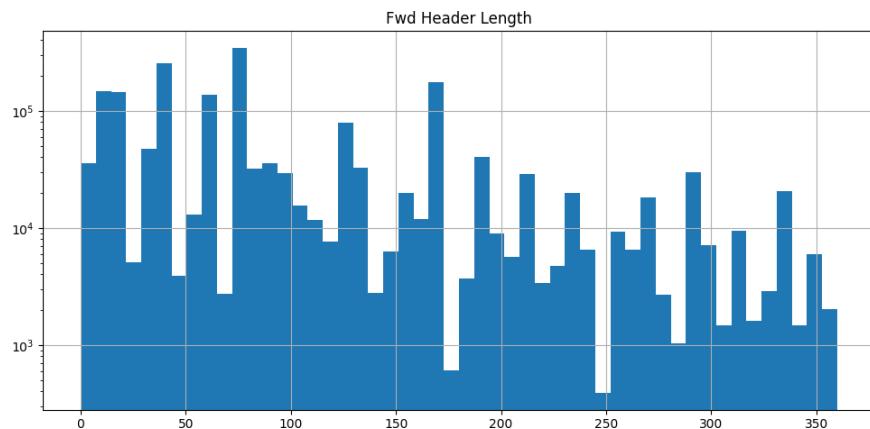


Figure 101: Histogram of Fwd Header Length plotted on log scale after handling negative values and outliers

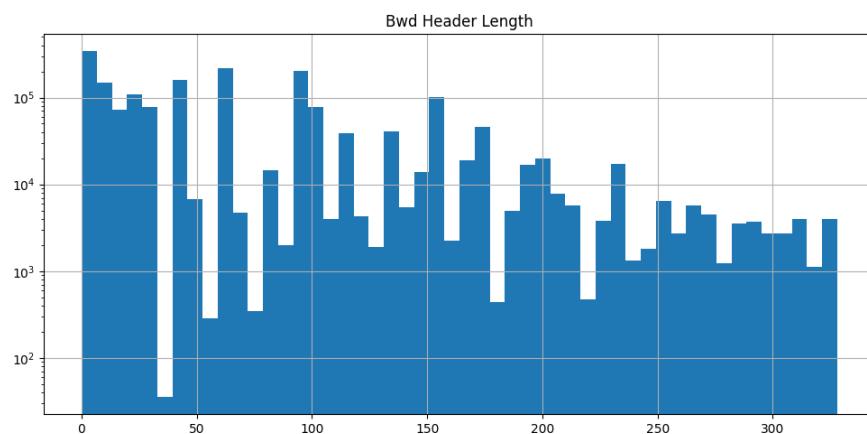


Figure 102: Histogram of Bwd Header Length plotted on log scale after handling negative values and outliers

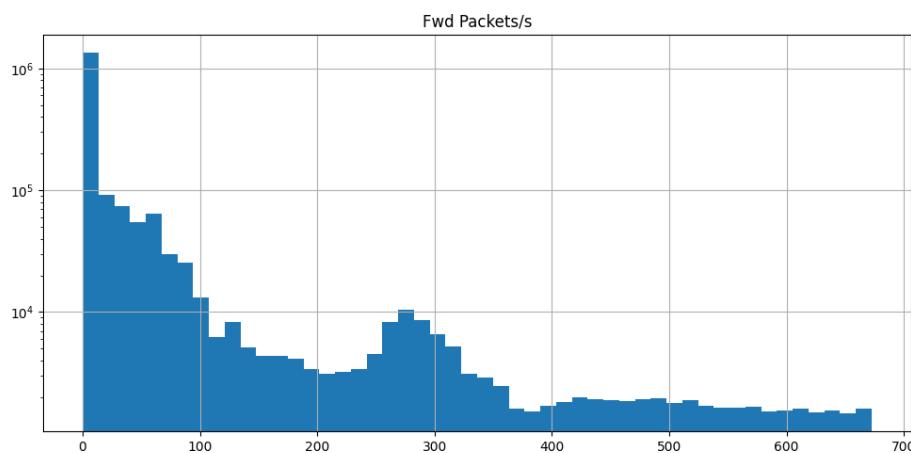


Figure 103: Histogram of Fwd Packets/s plotted on log scale after handling negative values and outliers

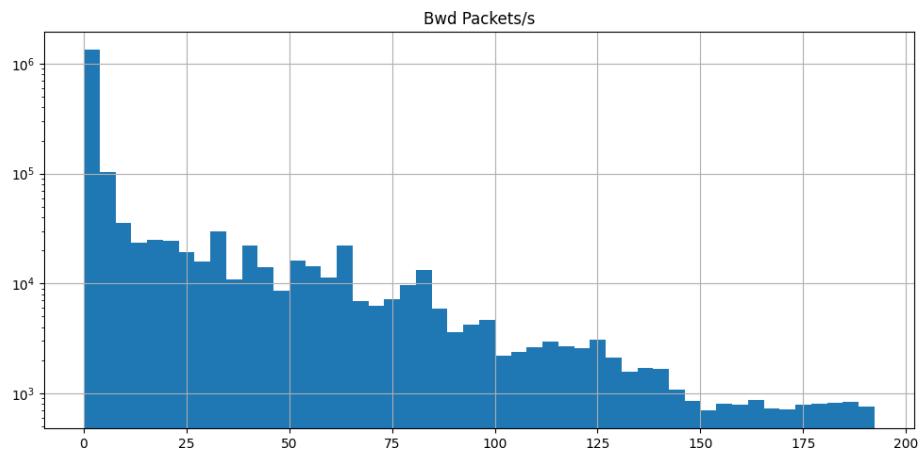


Figure 104: Histogram of Bwd Packets/s plotted on log scale after handling negative values and outliers

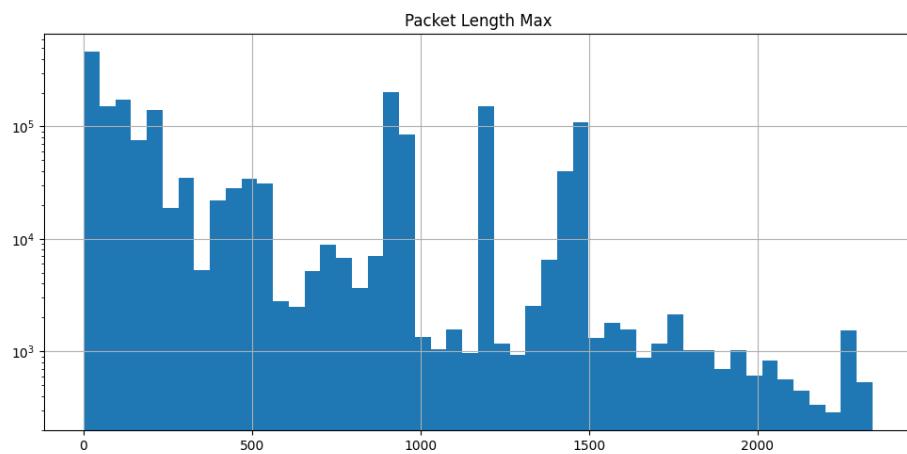


Figure 105: Histogram of Packet Length Max plotted on log scale after handling negative values and outliers

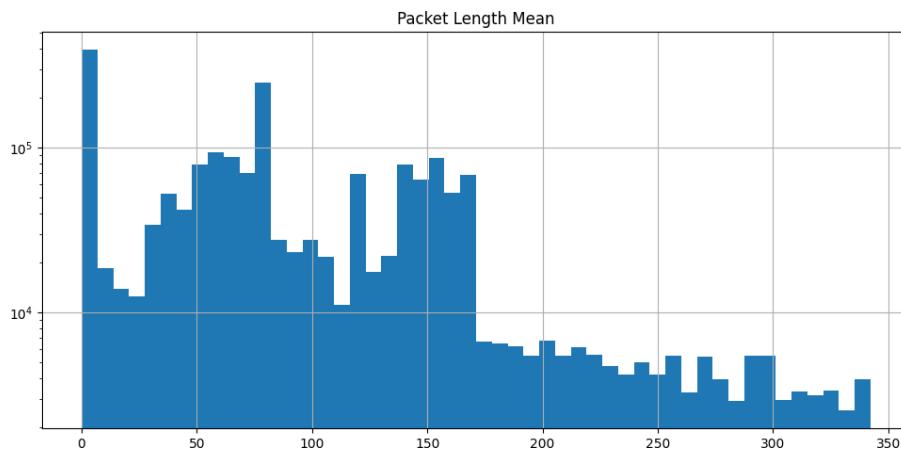


Figure 106: Histogram of Packet Length Mean plotted on log scale after handling negative values and outliers

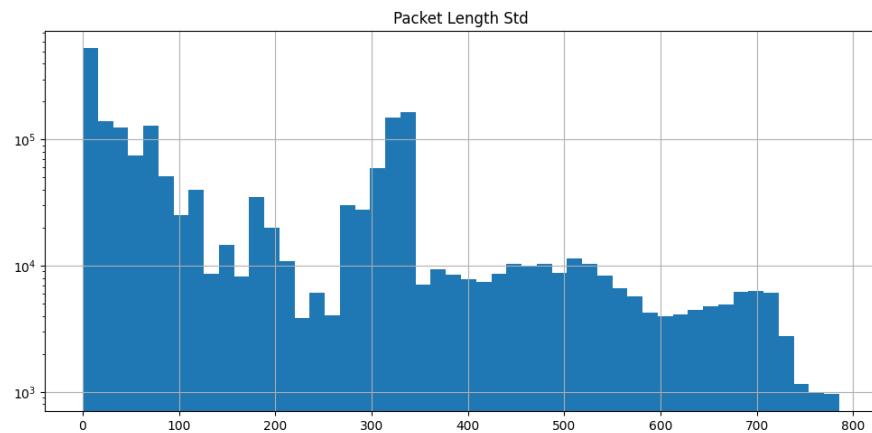


Figure 107: Histogram of Packet Length Std plotted on log scale after handling negative values and outliers

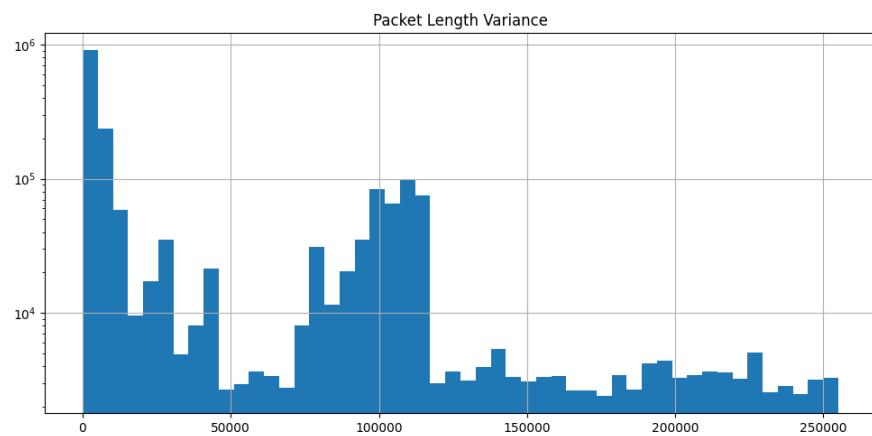


Figure 108: Histogram of Packet Length Variance plotted on log scale after handling negative values and outliers

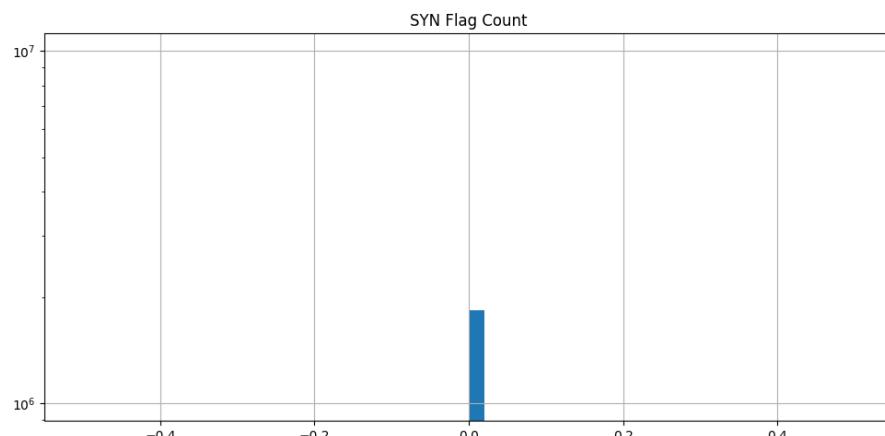


Figure 109: Histogram of SYN Flag Count plotted on log scale after handling negative values and outliers

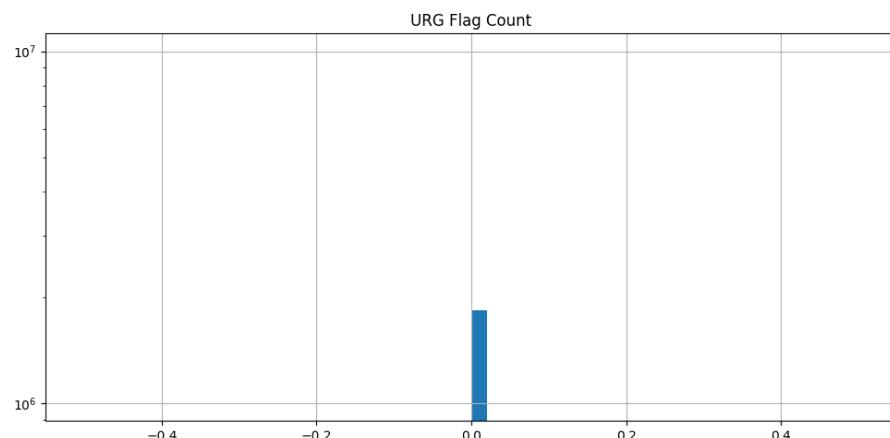


Figure 110: Histogram of URG Flag Count plotted on log scale after handling negative values and outliers

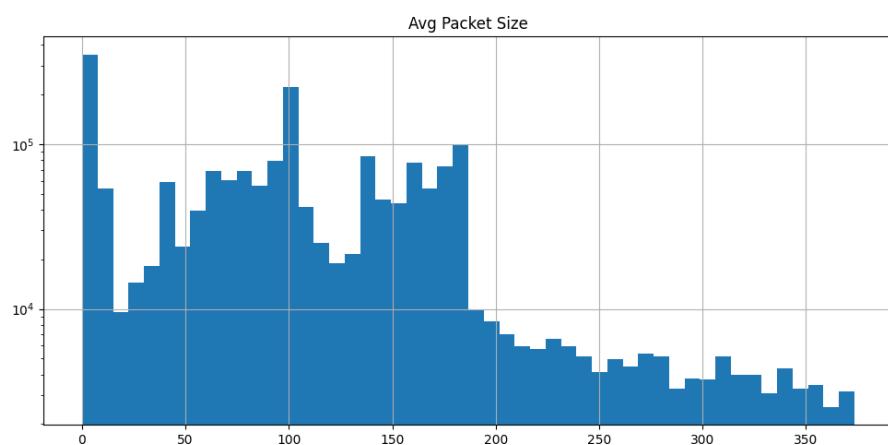


Figure 111: Histogram of Avg Packet Size plotted on log scale after handling negative values and outliers

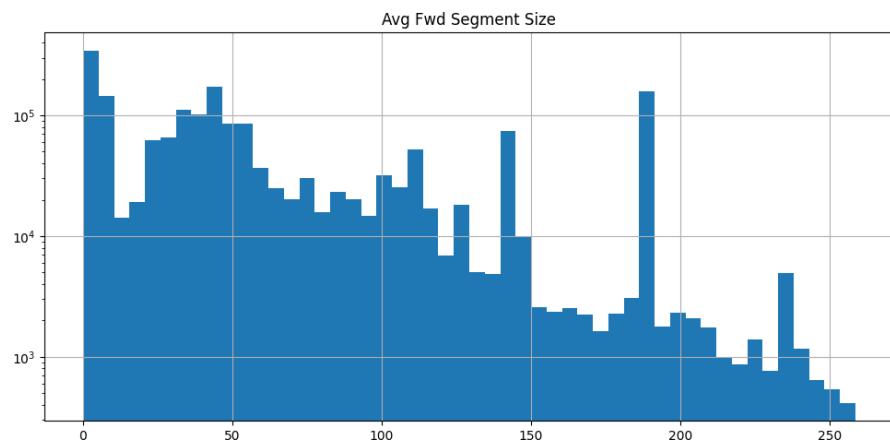


Figure 112: Histogram of Avg Fwd Segment Size plotted on log scale after handling negative values and outliers

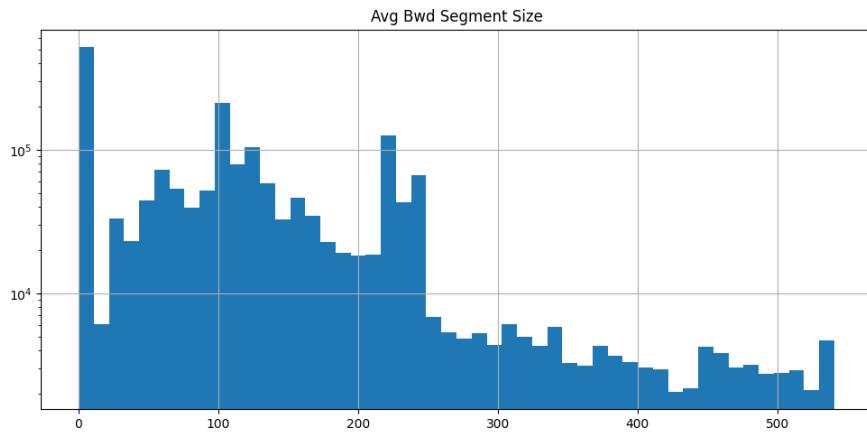


Figure 113: Histogram of Avg Bwd Segment Size plotted on log scale after handling negative values and outliers

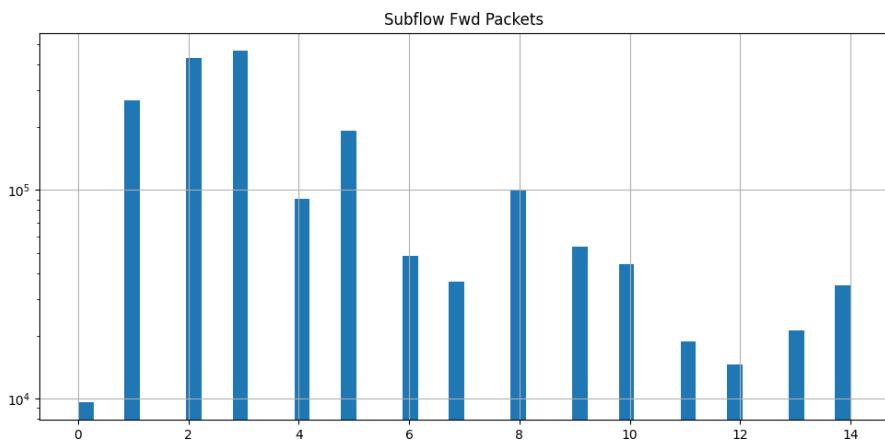


Figure 114: Histogram of Subflow Fwd Packets plotted on log scale after handling negative values and outliers

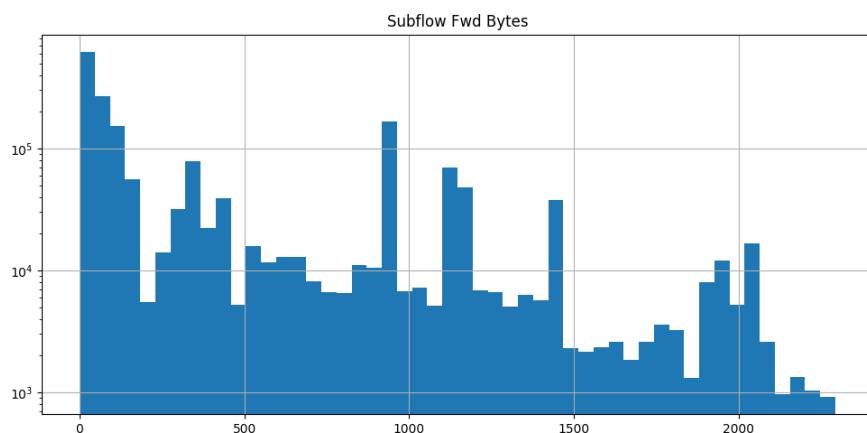


Figure 115: Histogram of Subflow Fwd Bytes plotted on log scale after handling negative values and outliers

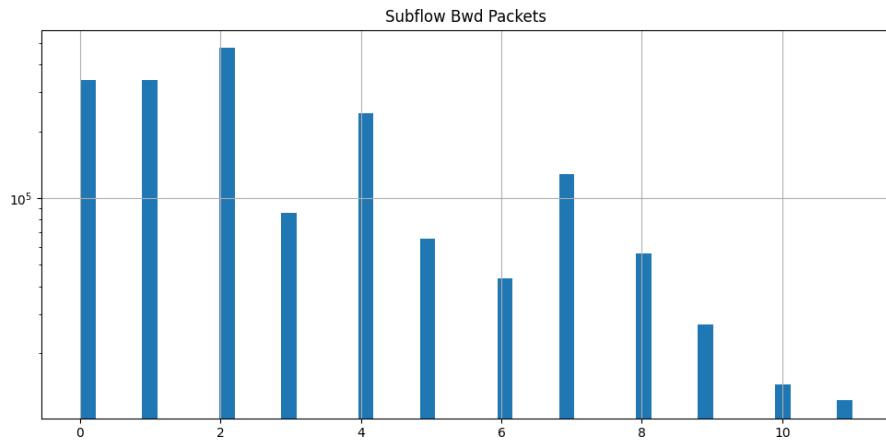


Figure 116: Histogram of Subflow Bwd Packets plotted on log scale after handling negative values and outliers

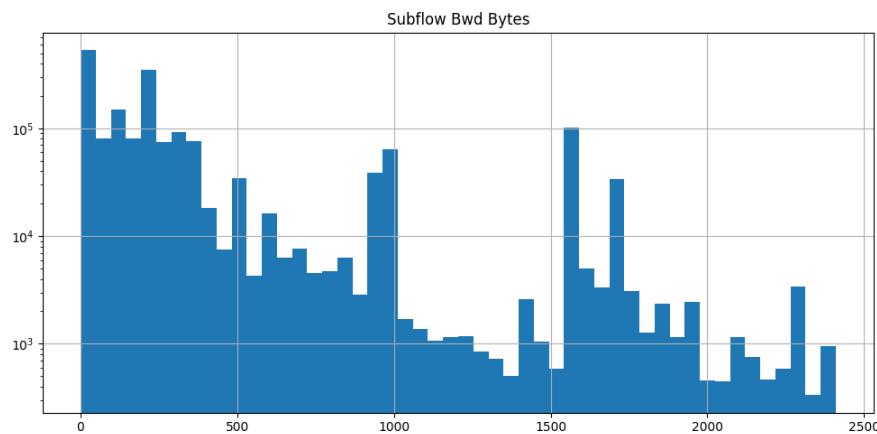


Figure 117: Histogram of Subflow Bwd Bytes plotted on log scale after handling negative values and outliers

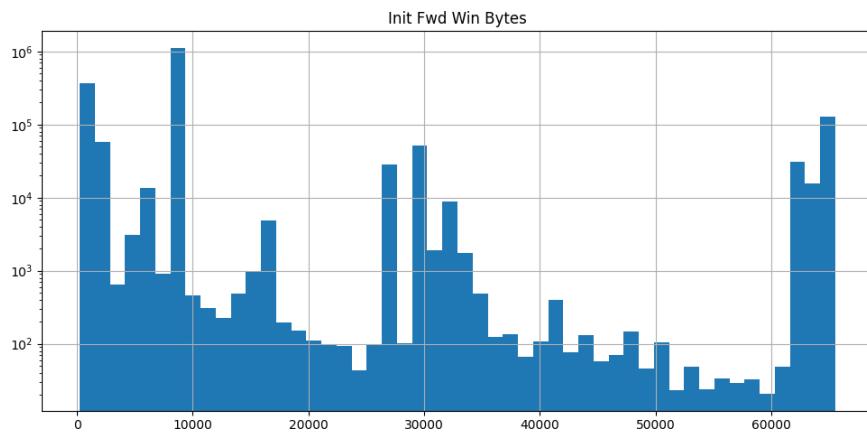


Figure 118: Histogram of Init Fwd Win Bytes plotted on log scale after handling negative values and outliers

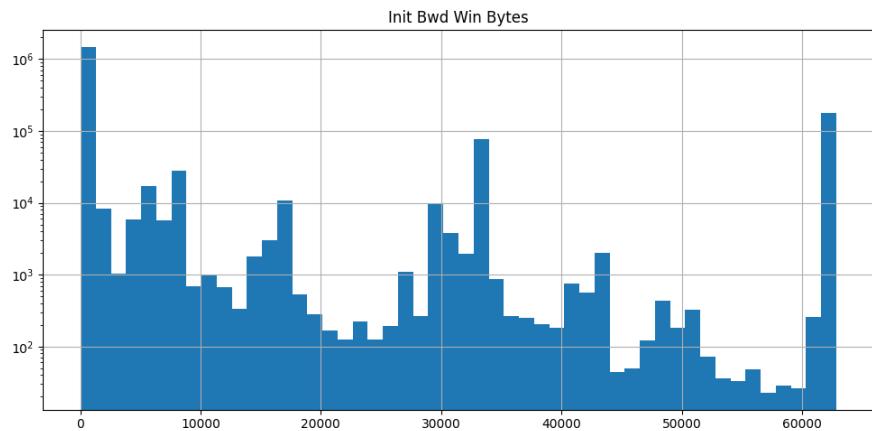


Figure 119: Histogram of Init Bwd Win Bytes plotted on log scale after handling negative values and outliers

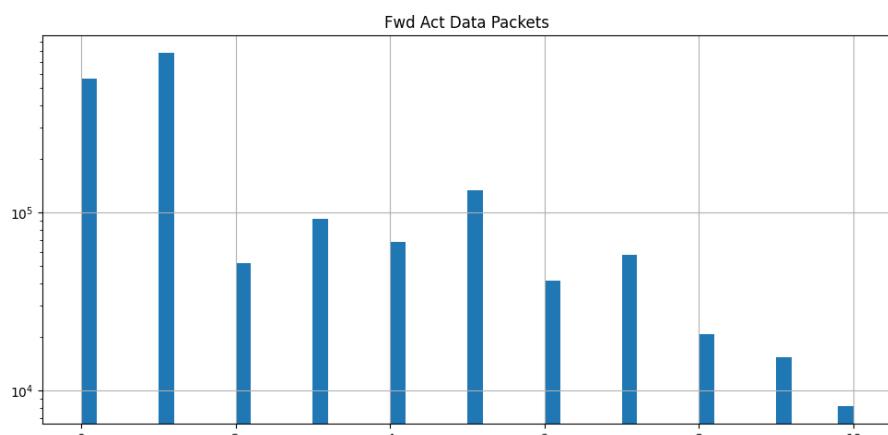


Figure 120: Histogram of Fwd Act Data Packets plotted on log scale after handling negative values and outliers

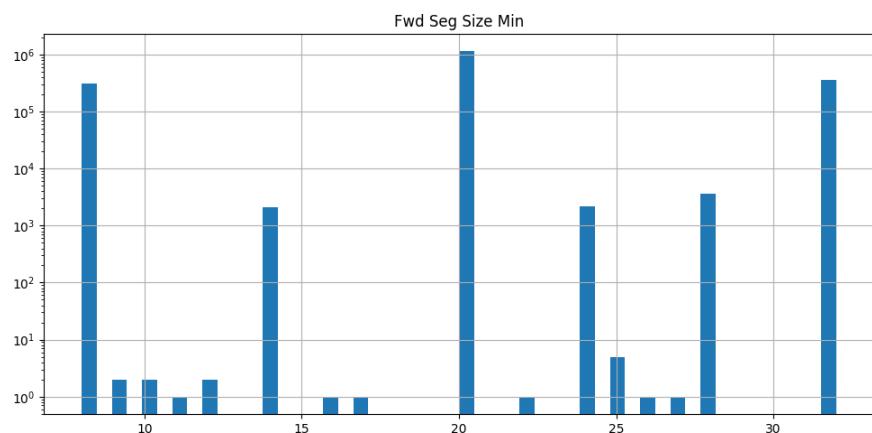


Figure 121: Histogram of Fwd Seg Size Min plotted on log scale after handling negative values and outliers

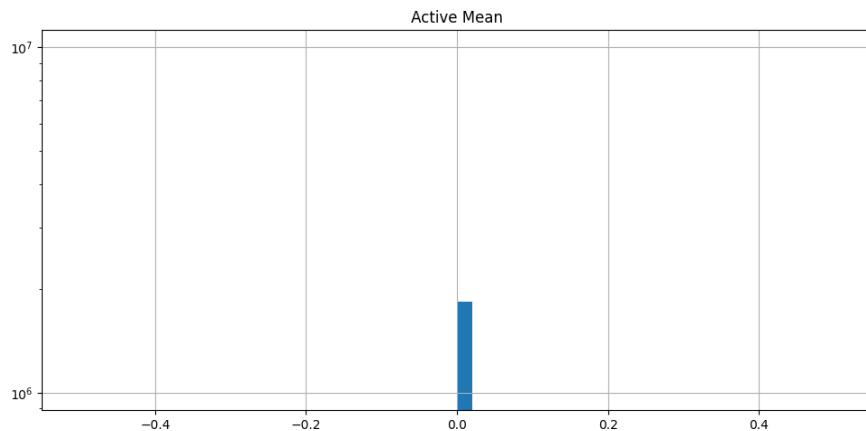


Figure 122: Histogram of Active Mean plotted on log scale after handling negative values and outliers

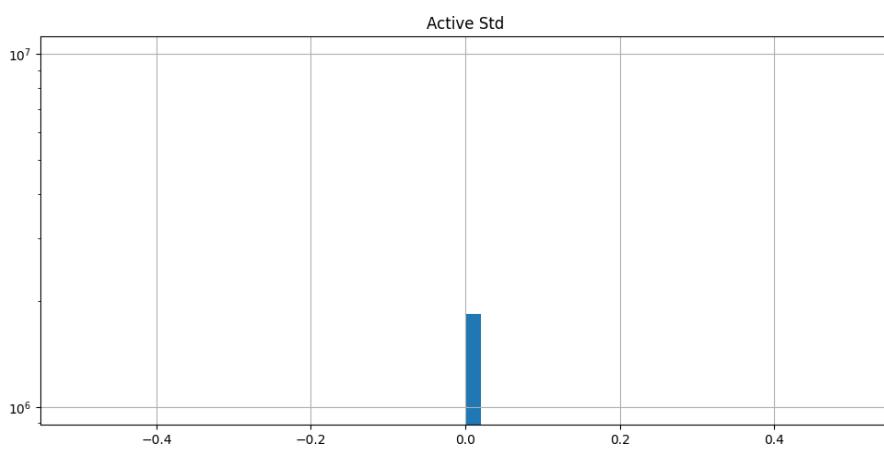


Figure 123: Histogram of Active Std plotted on log scale after handling negative values and outliers

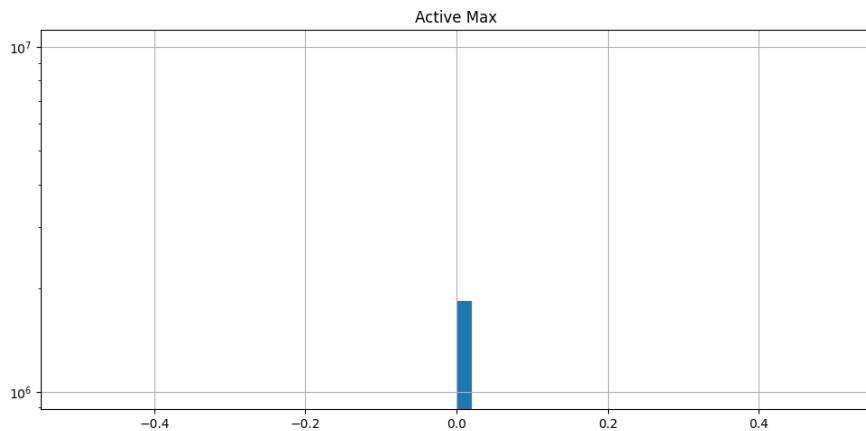


Figure 124: Histogram of Active Max plotted on log scale after handling negative values and outliers

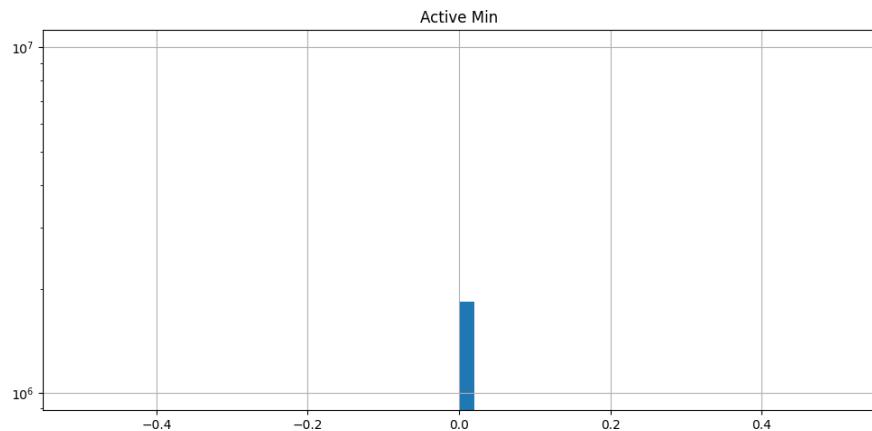


Figure 125: Histogram of Active Min plotted on log scale after handling negative values and outliers

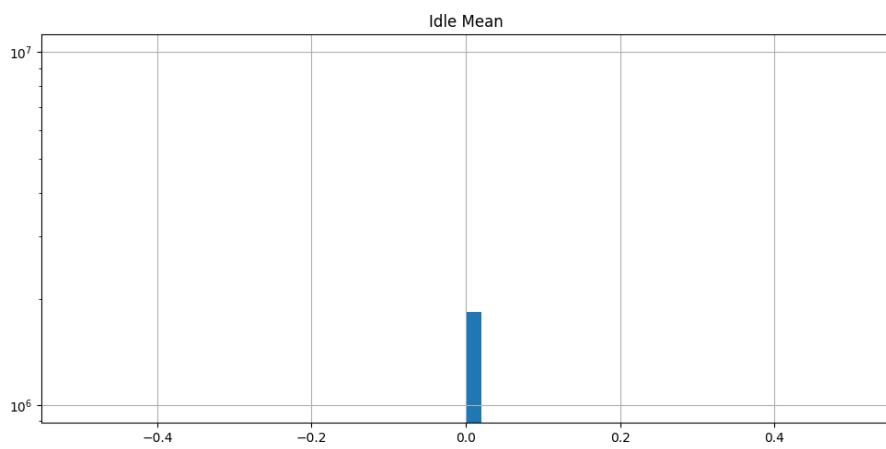


Figure 126: Histogram of Idle Mean plotted on log scale after handling negative values and outliers

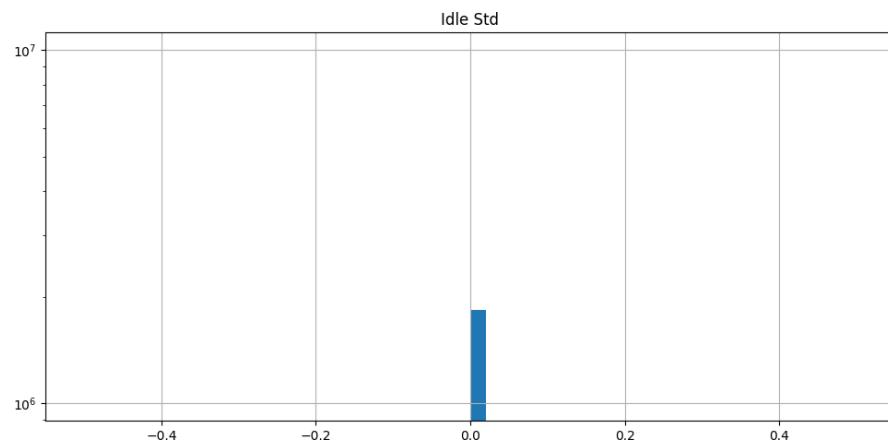


Figure 127: Histogram of Idle Std plotted on log scale after handling negative values and outliers

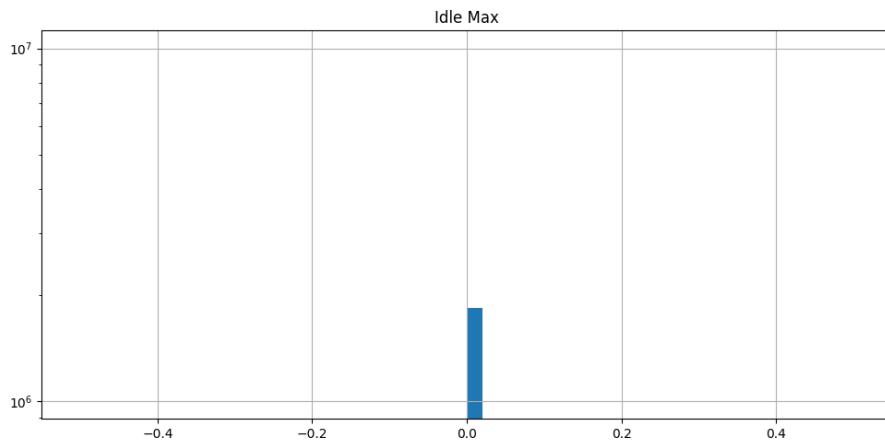


Figure 128: Histogram of Idle Max plotted on log scale after handling negative values and outliers

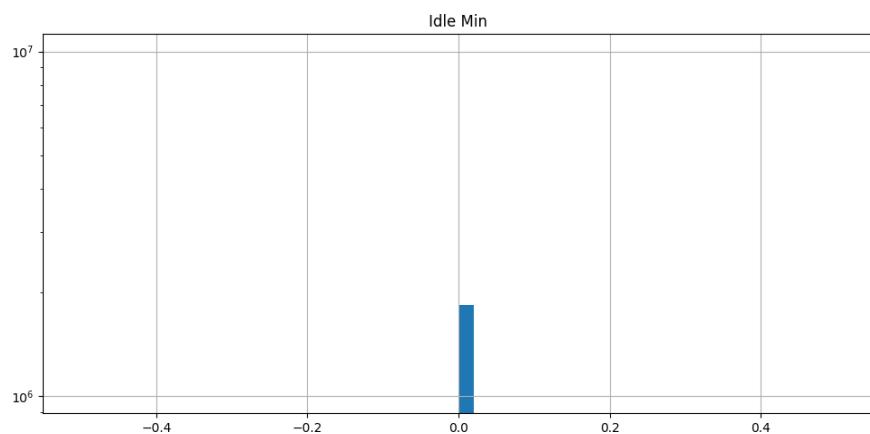


Figure 129: Histogram of Idle Min plotted on log scale after handling negative values and outliers

Among the new histograms, there were some features with datapoints only in single bin. Thus, they may be the features with single value. Such features will not help to train the classifier model because irrespective of category, those features will remain unchanged.

Thus, the list of features having a single value was fetched: -

1. Fwd PSH Flags
2. SYN Flag Count
3. URG Flag Count
4. Active Mean
5. Active Std
6. Active Max
7. Active Min
8. Idle Mean
9. Idle Std
10. Idle Max
11. Idle Min

The above features were dropped from the dataset because they will not contribute towards training the model.

New shape of the main dataset: (9167271, 48).

New shape of sampled dataset: (1833454, 48).

Pyramid chart with respect to isMalicious: -

Pyramid chart was plotted for each independent feature with respect to the target binary feature: isMalicious.

The continuous data in each feature was transformed into discrete categorical bins and then the charts were plotted to fetch new information from the dataset.

If the number of bins were too less, the graph will be too smooth and thus, no relationship with different ranges of data can be determined.

If the number of bins were too many, we will get a line for almost every datapoint.

Following are commonly known methods to determine the number of bins: -

1. Sturge's rule
2. Doane's rule
3. Rice rule
4. Square root rule
5. Scott's rule
6. Freedman-Diaconis rule
7. Knuth's rule
8. Scargle's Bayesian blocks

Bayesian algorithms such as Knuth's rule and Scargle's Bayesian blocks are useful when the data points are skewed, heavy-tailed and have multi-modal distribution.

However, plotting Pyramid charts based on the Bayesian algorithm was not feasible due to limitations of the system's configurations.

Thus, to compute the number of bins for each feature, Freedman-Diaconis rule was used.

Freedman-Diaconis rule was selected because: -

- i. It helps to compute bin width based on each feature's IQR. Thus, it helps to reduce the impact of skewness in data.
- ii. It does not assume the feature to be normally distributed.
- iii. Since it uses IQR, the rule also helps to handle values at extreme end

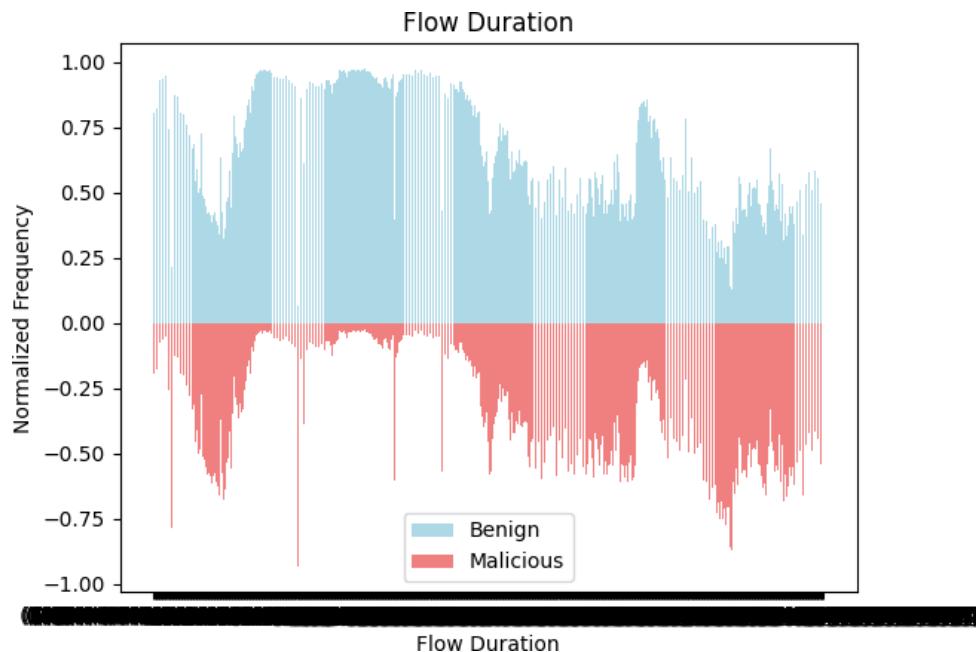


Figure 130: Pyramid chart of Flow Duration w.r.t isMalicious

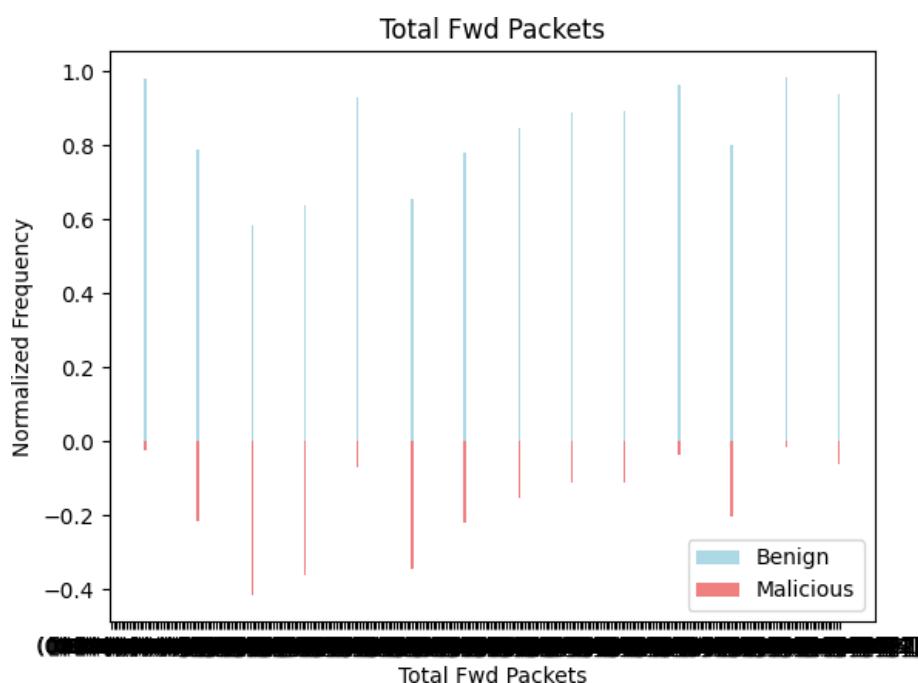


Figure 131: Pyramid chart of Total Fwd Packets w.r.t isMalicious

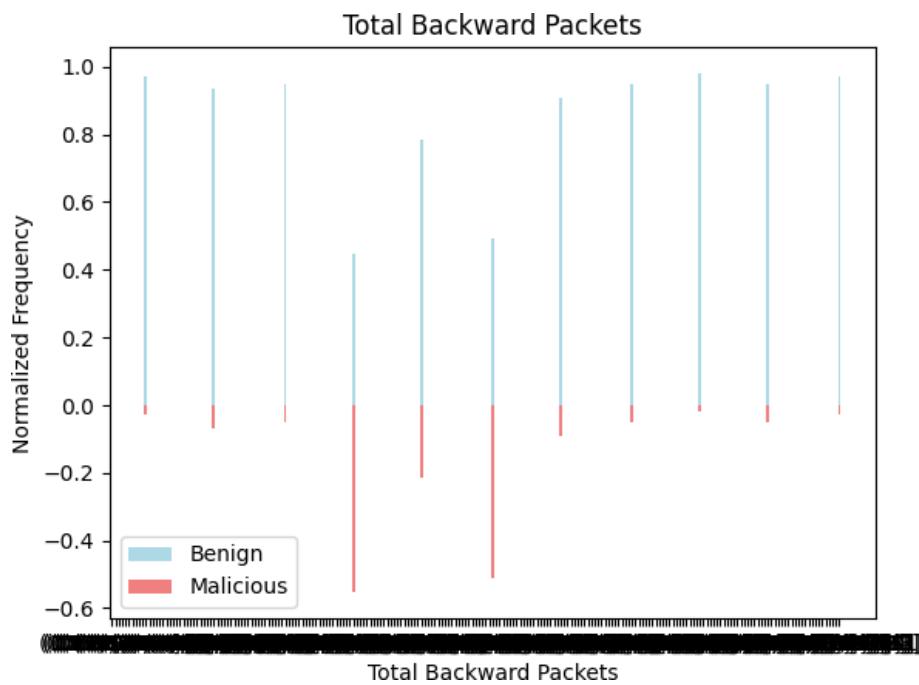


Figure 132: Pyramid chart of Total Backward Packets w.r.t isMalicious

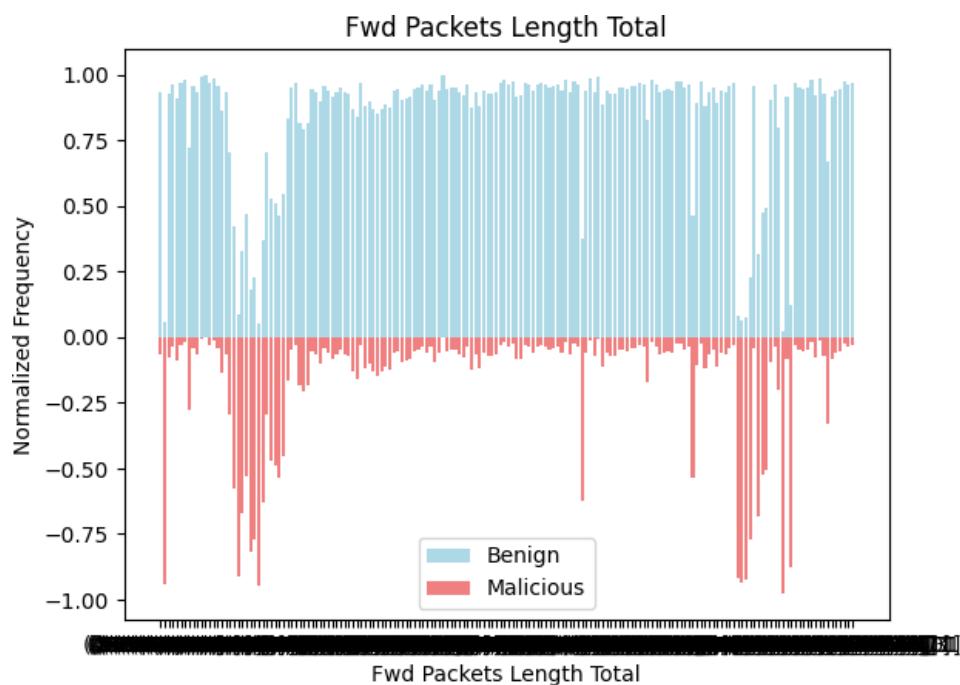


Figure 133: Pyramid chart of Fwd Packets Length Total w.r.t isMalicious

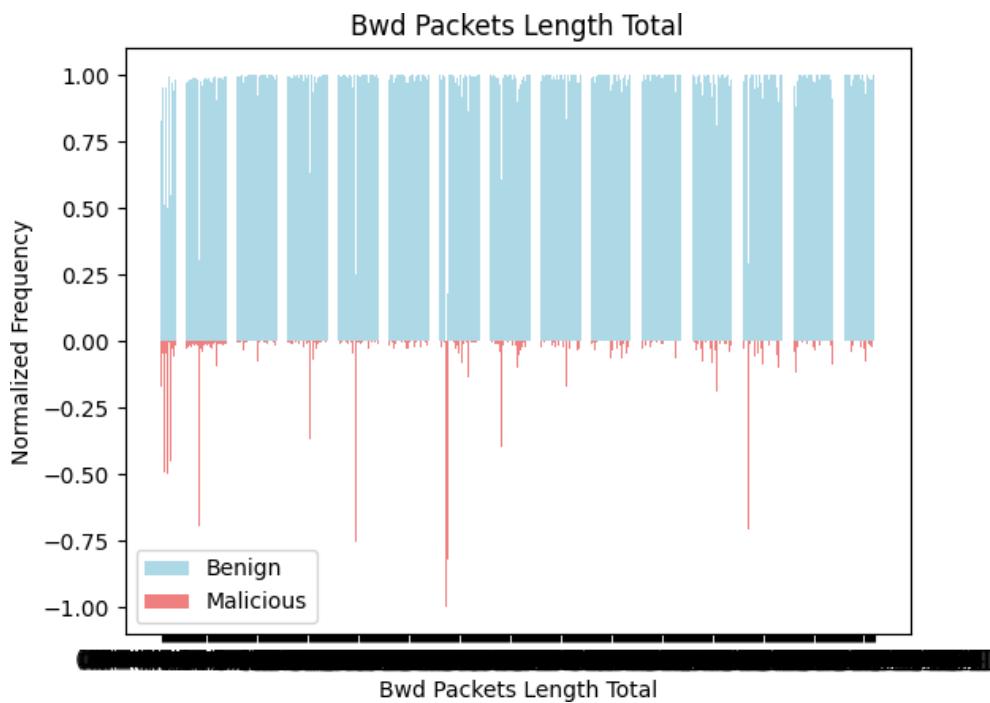


Figure 134: Pyramid chart of Bwd Packets Length Total w.r.t isMalicious

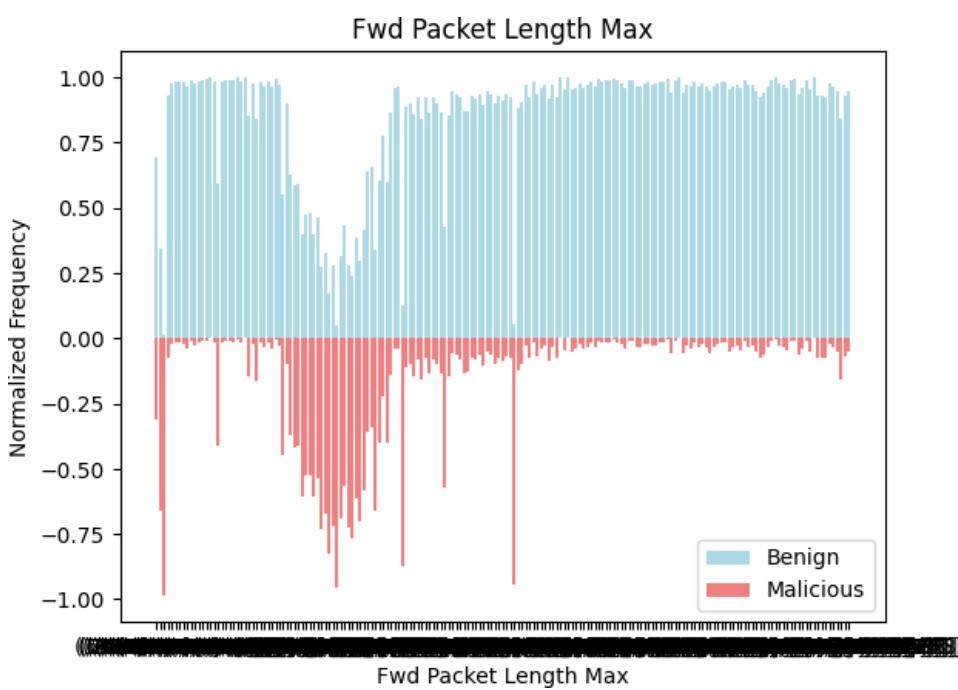


Figure 135: Pyramid chart of Fwd Packet Length Max w.r.t isMalicious

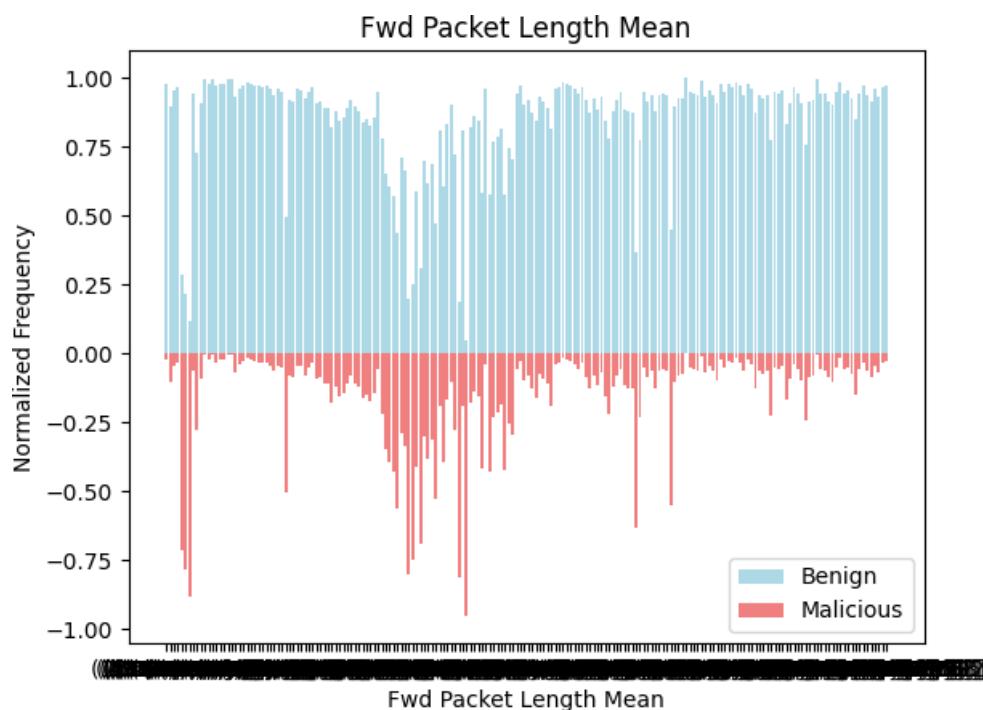


Figure 136: Pyramid chart of Fwd Packet Length Mean w.r.t isMalicious

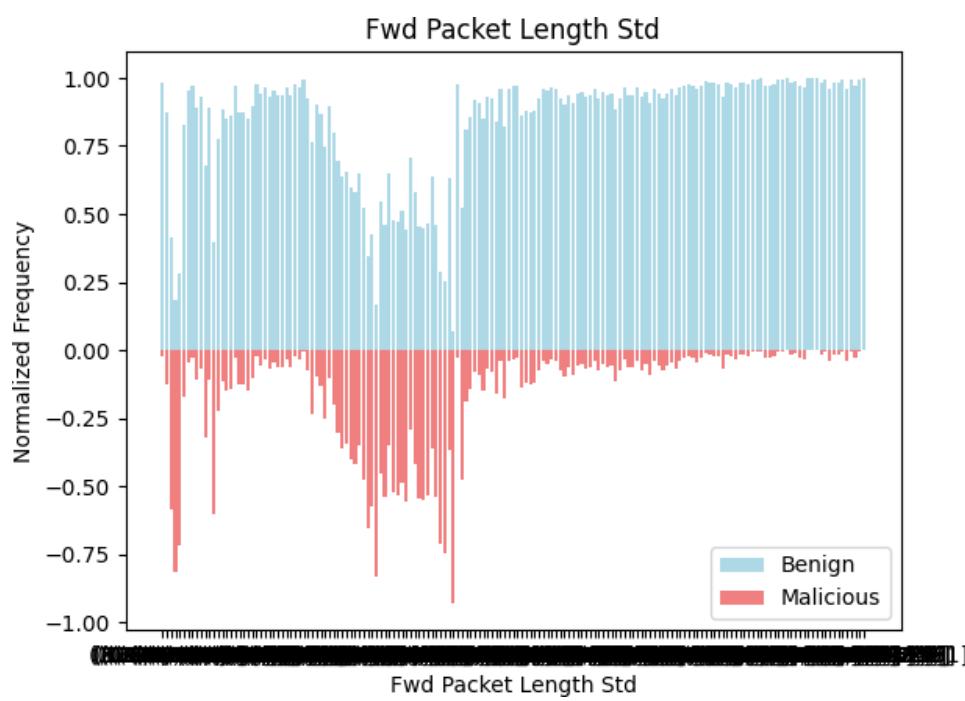


Figure 137: Pyramid chart of Fwd Packet Length Std w.r.t isMalicious

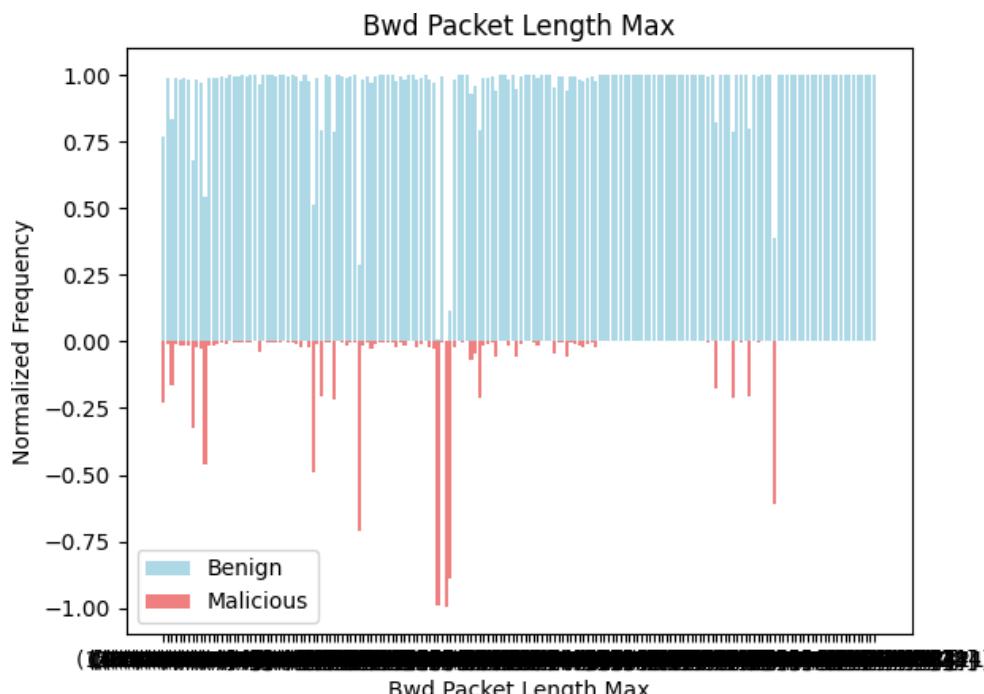


Figure 138: Pyramid chart of Bwd Packet Length Max w.r.t isMalicious

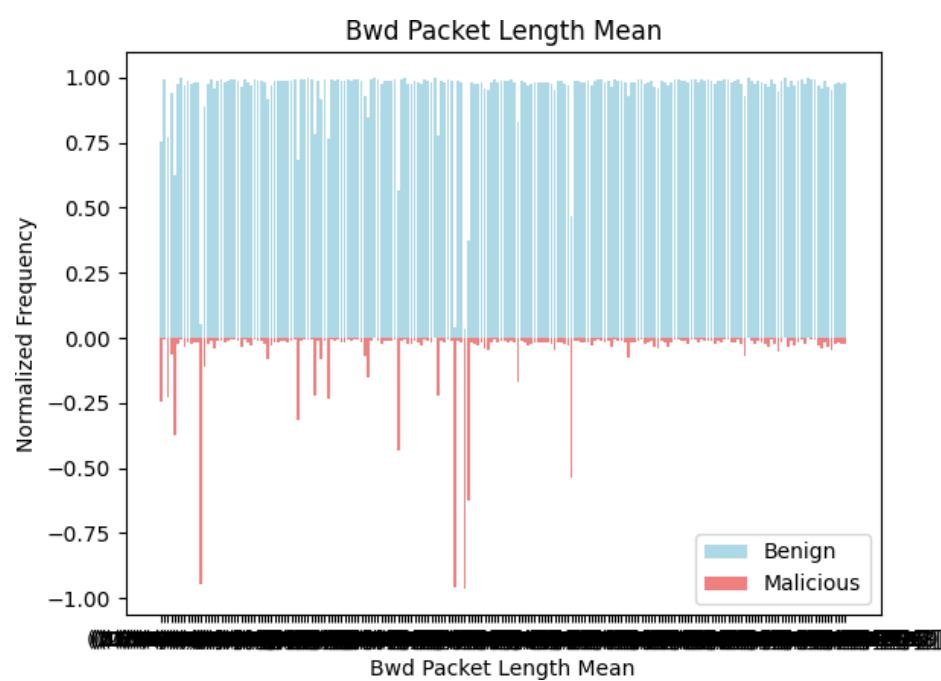


Figure 139: Pyramid chart of Bwd Packet Length Mean w.r.t isMalicious

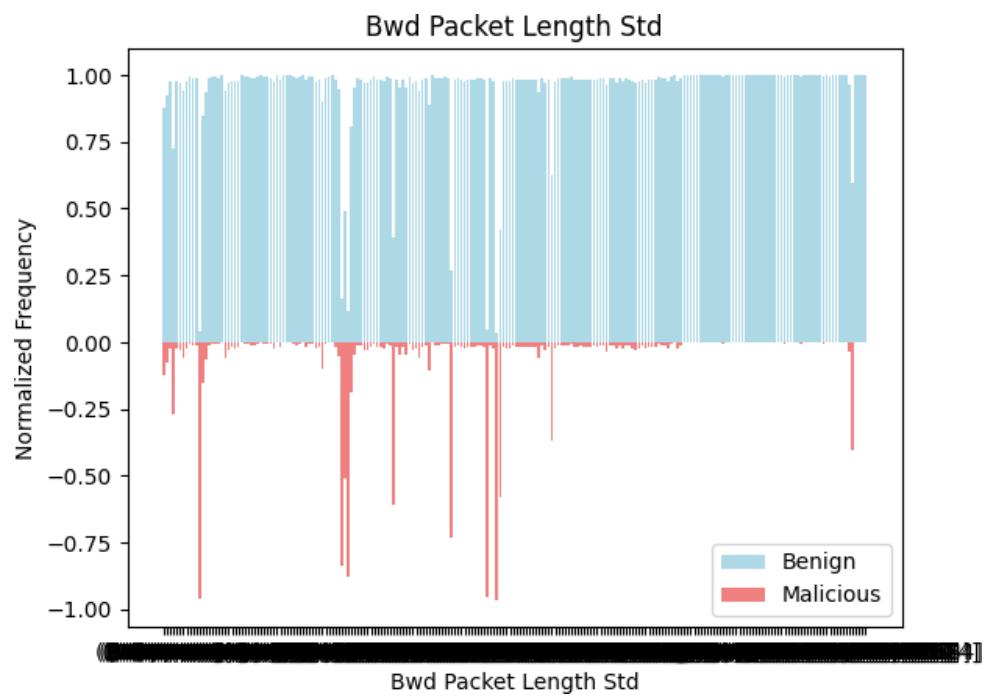


Figure 140: Pyramid chart of Bwd Packet Length Std w.r.t isMalicious

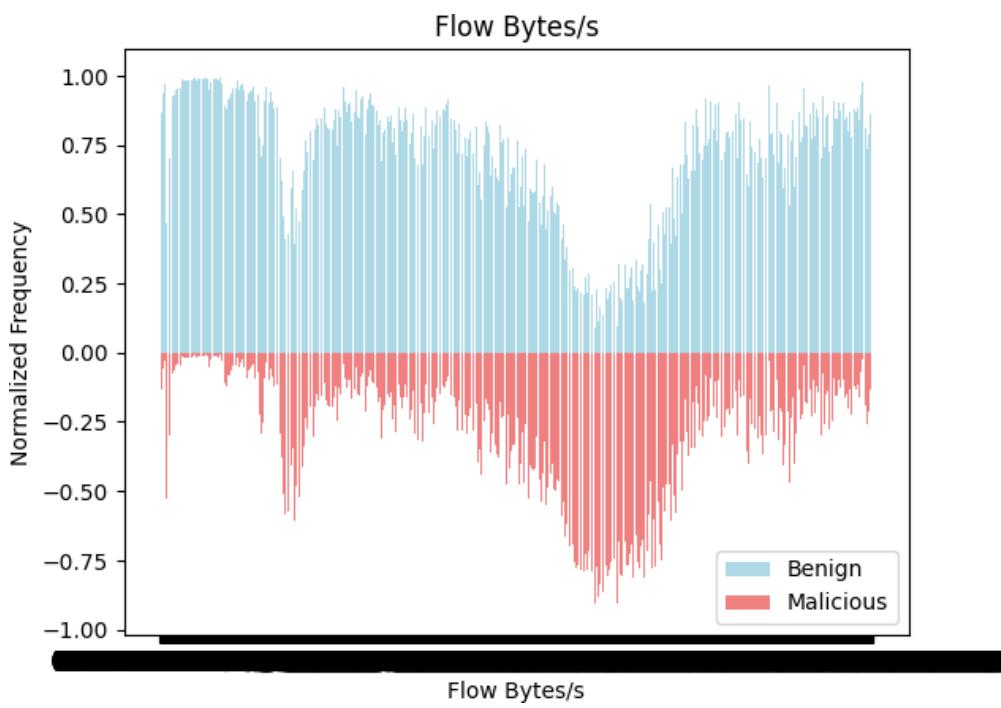


Figure 141: Pyramid chart of Flow Bytes/s w.r.t isMalicious

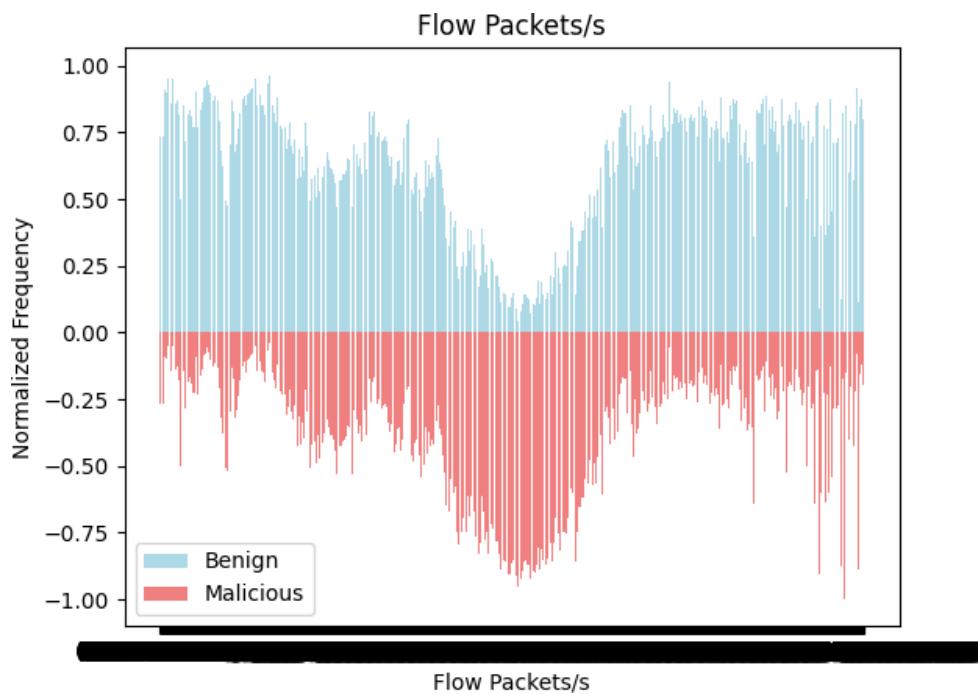


Figure 142: Pyramid chart of Flow Packets/s w.r.t isMalicious

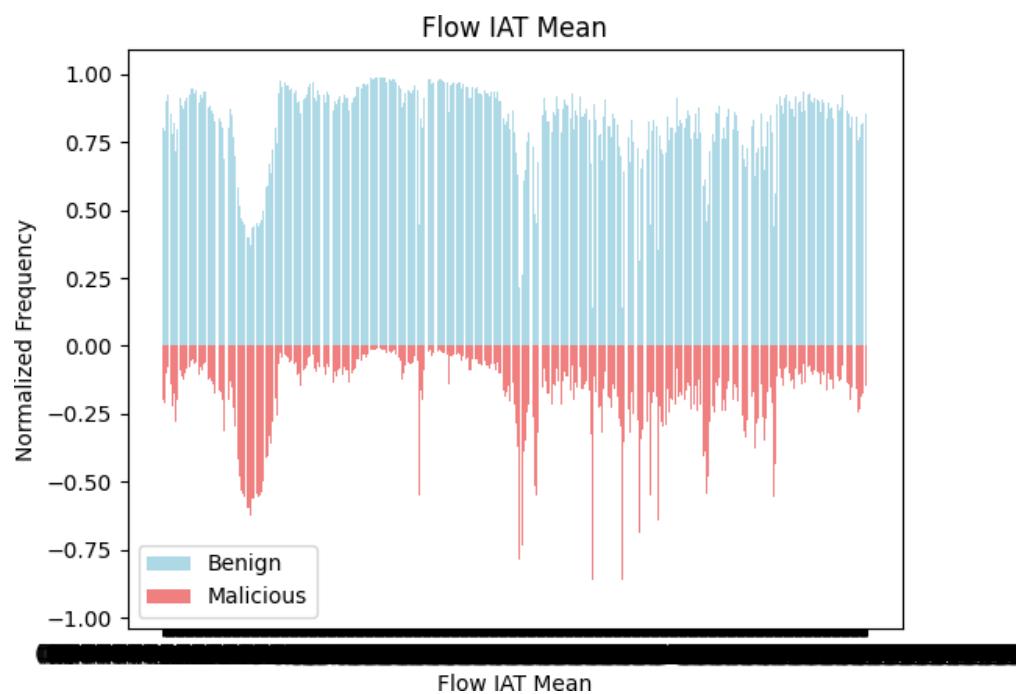


Figure 143: Pyramid chart of Flow IAT Mean w.r.t isMalicious

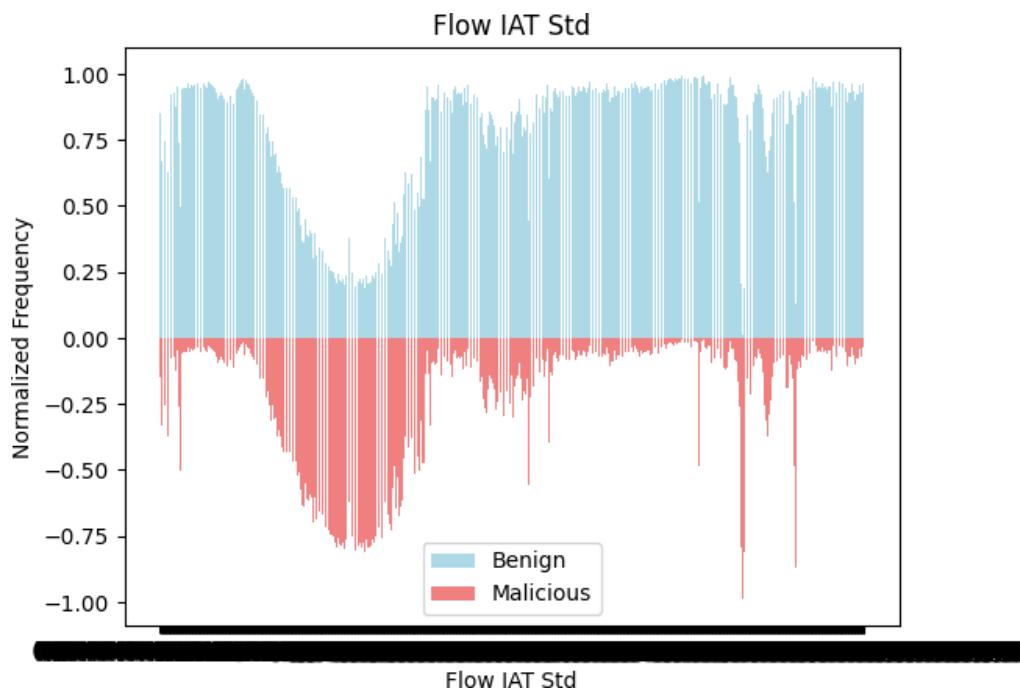


Figure 144: Pyramid chart of Flow IAT Std w.r.t isMalicious

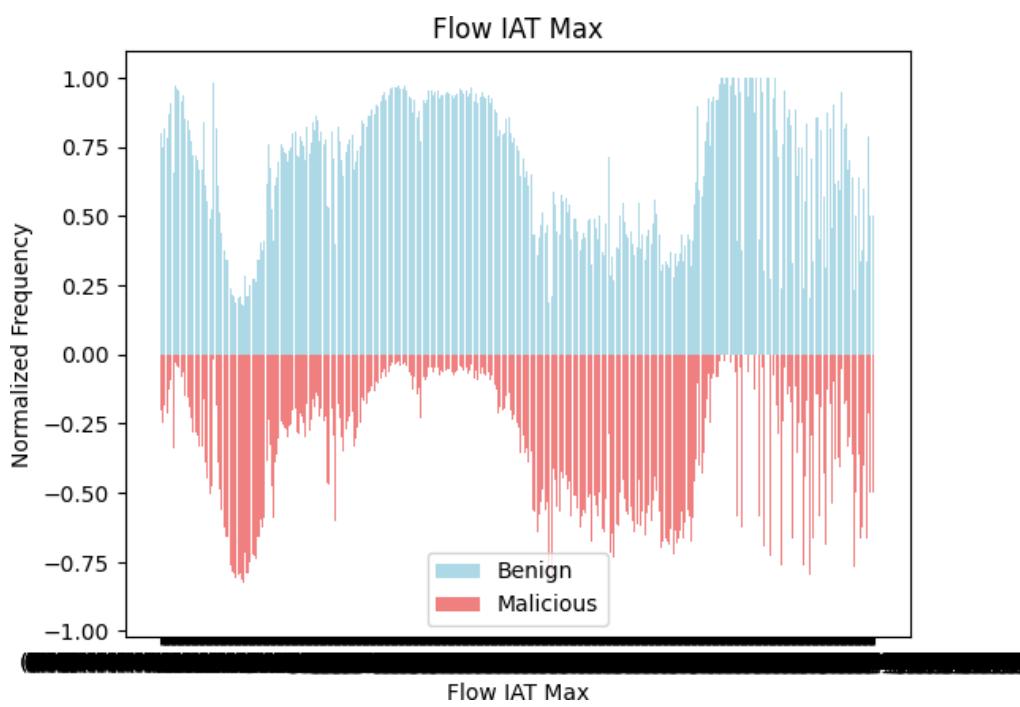


Figure 145: Pyramid chart of Flow IAT Max w.r.t isMalicious

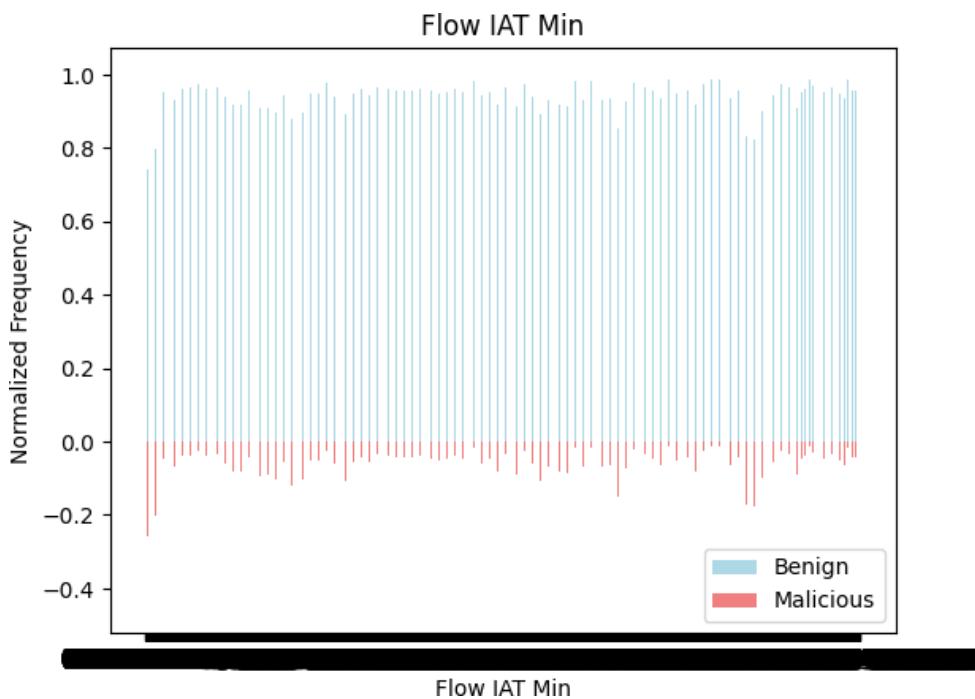


Figure 146: Pyramid chart of Flow IAT Min w.r.t isMalicious

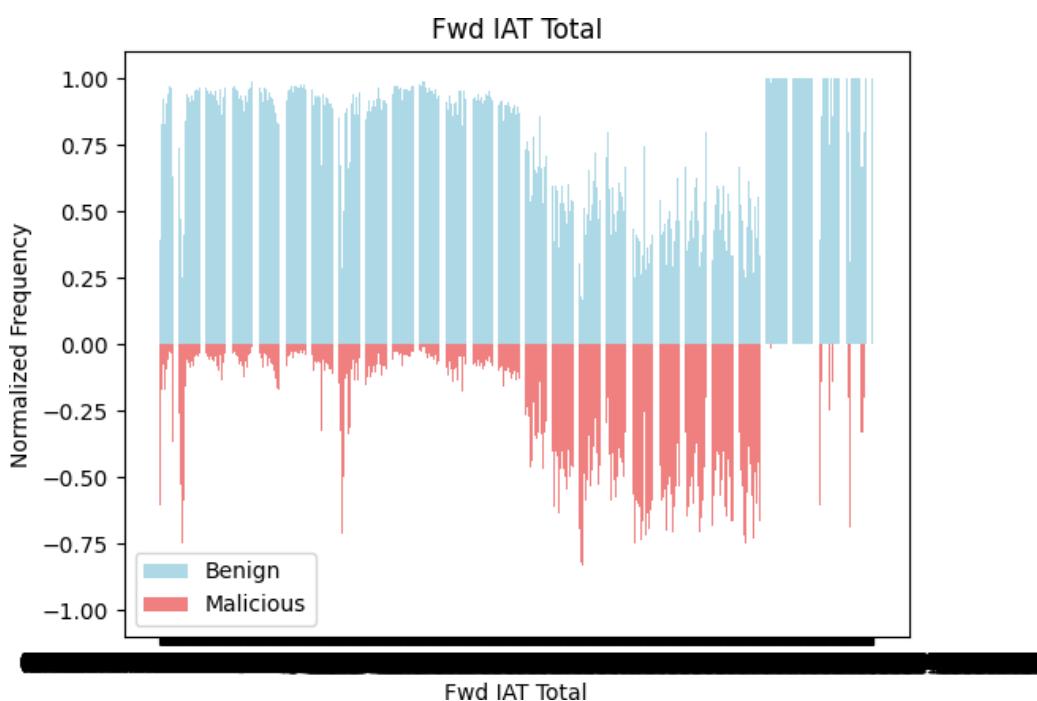


Figure 147: Pyramid chart of Fwd IAT Total w.r.t isMalicious

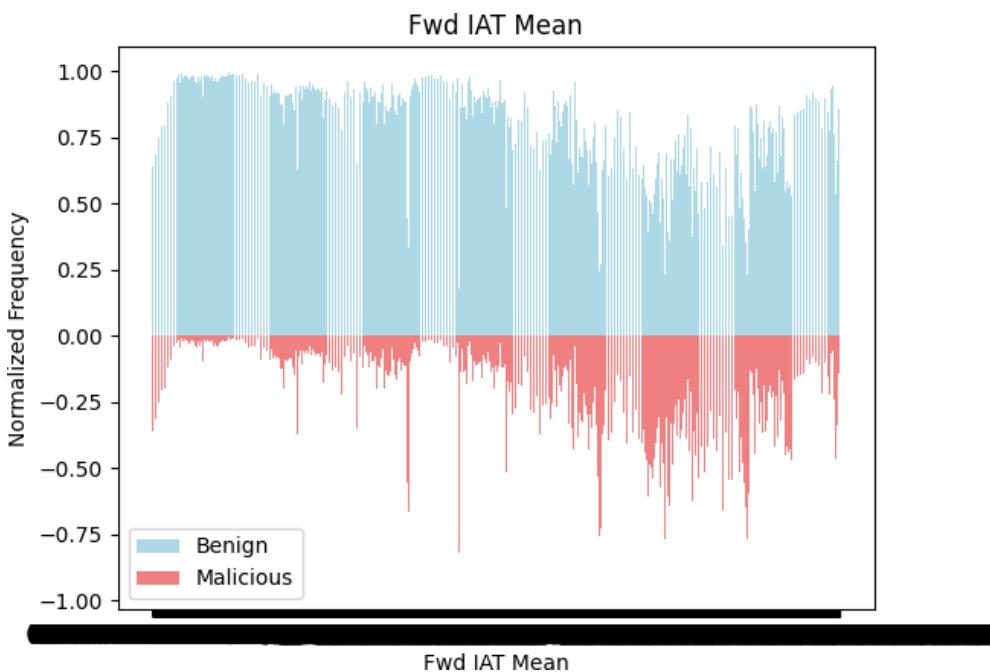


Figure 148: Pyramid chart of Fwd IAT Mean w.r.t isMalicious

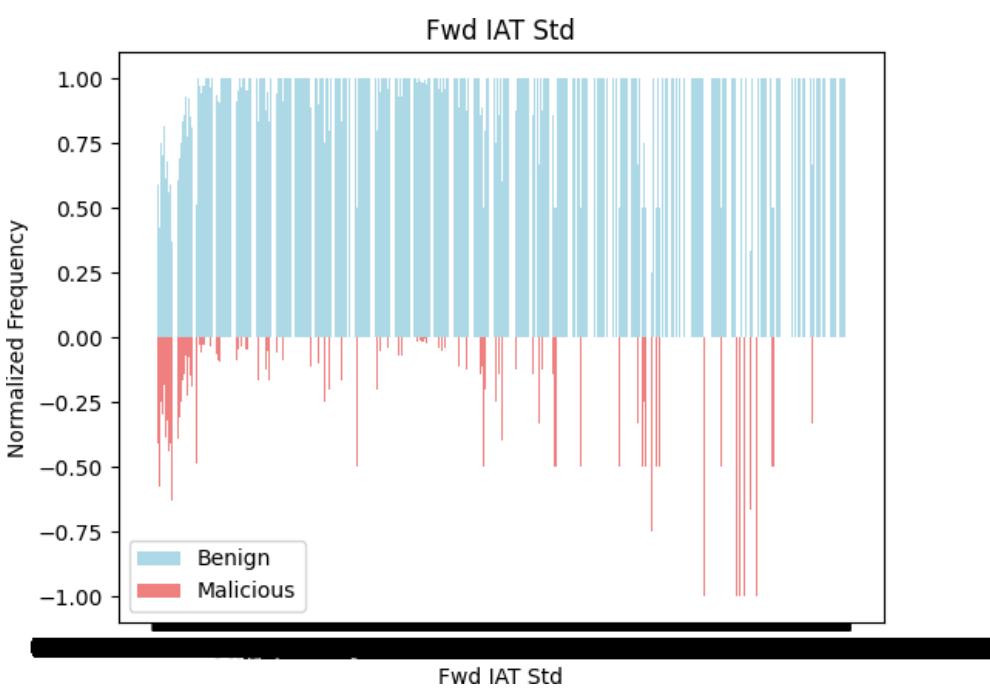


Figure 149: Pyramid chart of Fwd IAT Std w.r.t isMalicious

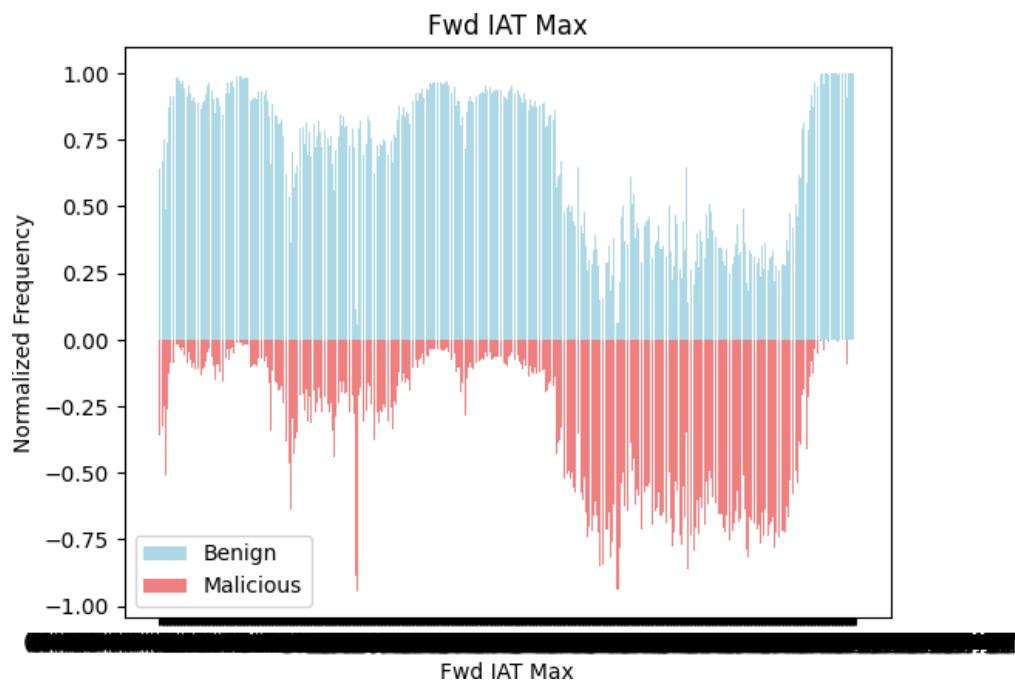


Figure 150: Pyramid chart of Fwd IAT Max w.r.t isMalicious

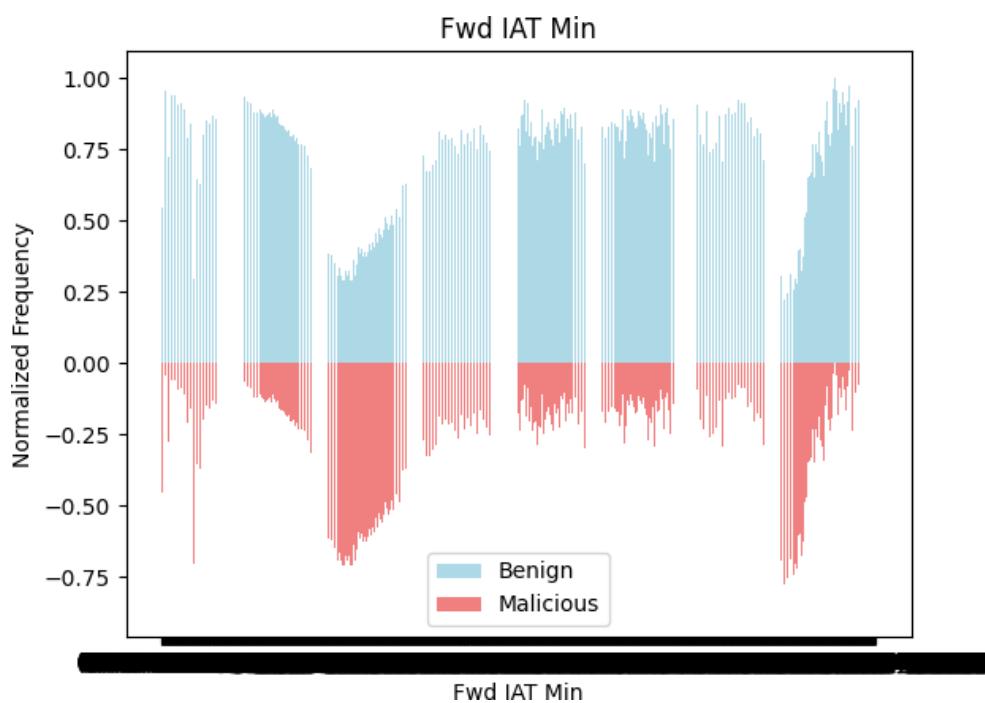


Figure 151: Pyramid chart of Fwd IAT Min w.r.t isMalicious

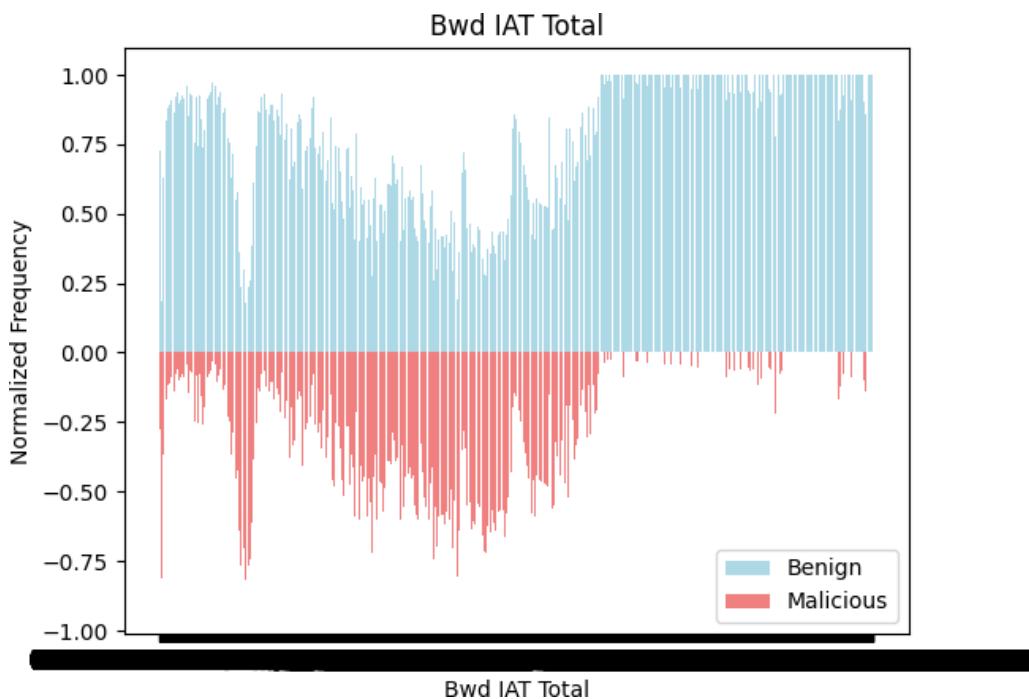


Figure 152: Pyramid chart of Bwd IAT Total w.r.t isMalicious

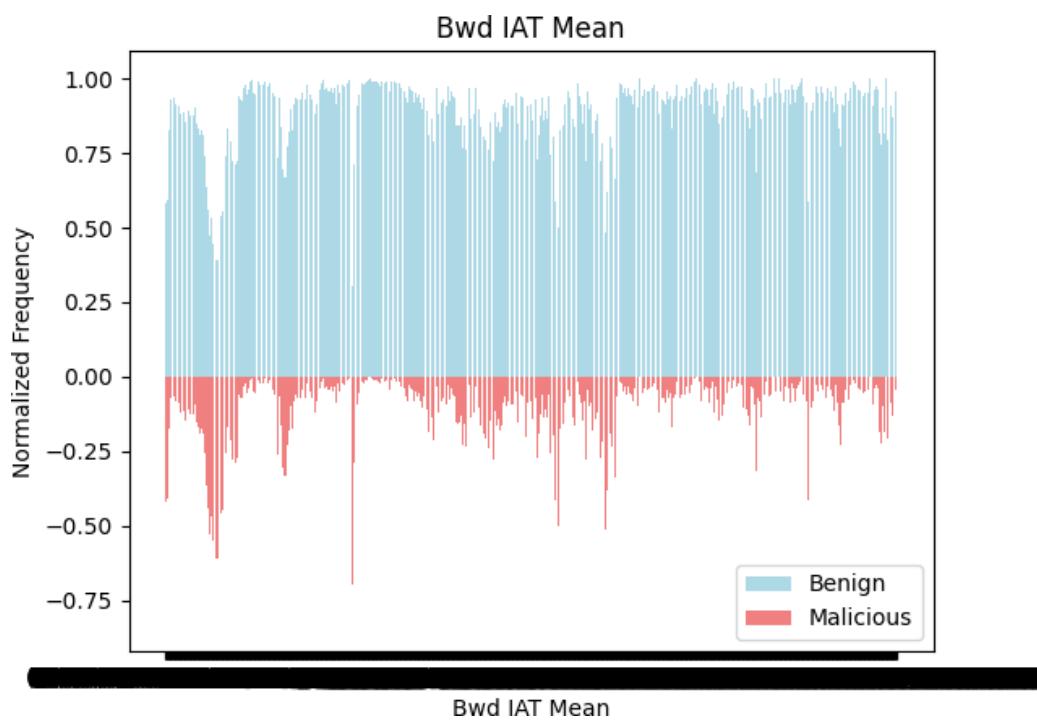


Figure 153: Pyramid chart of Bwd IAT Mean w.r.t isMalicious

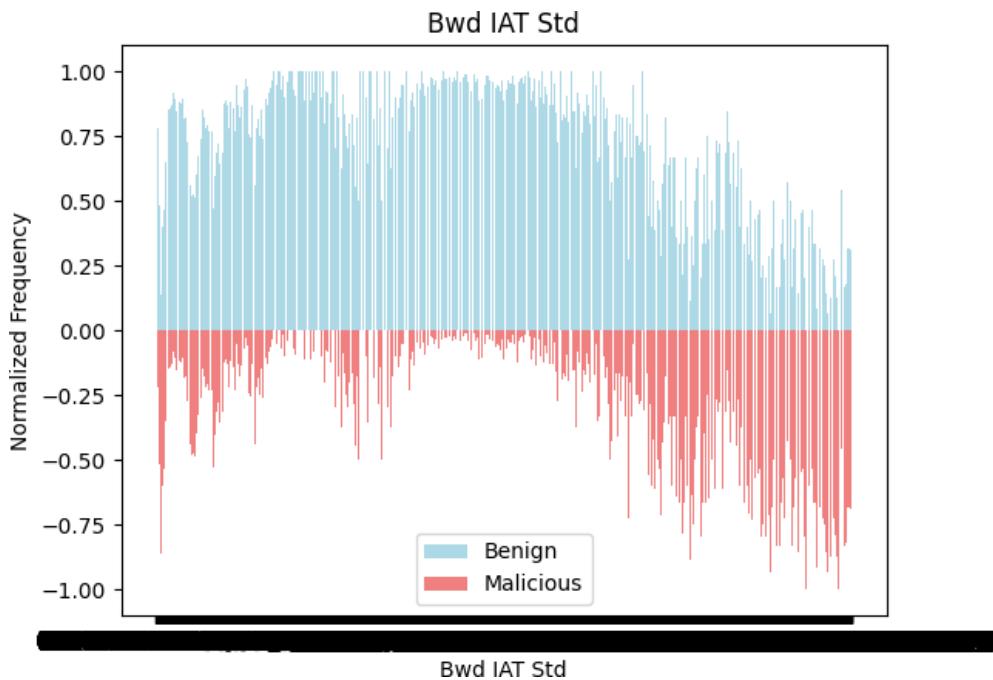


Figure 154: Pyramid chart of Bwd IAT Std w.r.t isMalicious

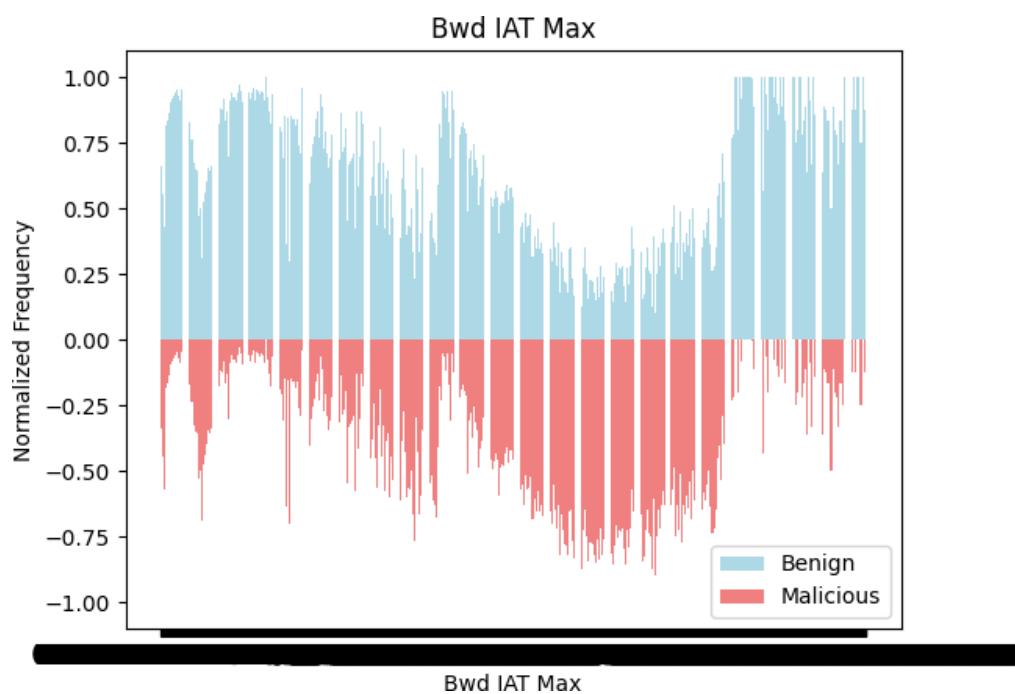


Figure 155: Pyramid chart of Bwd IAT Max w.r.t isMalicious

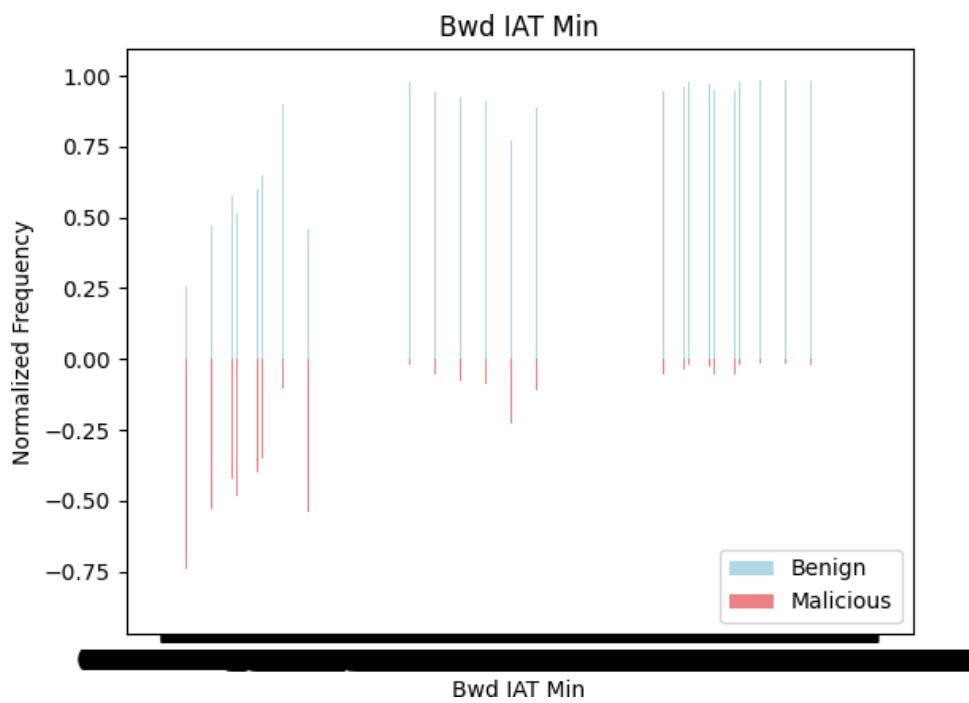


Figure 156: Pyramid chart of Bwd IAT Min w.r.t isMalicious

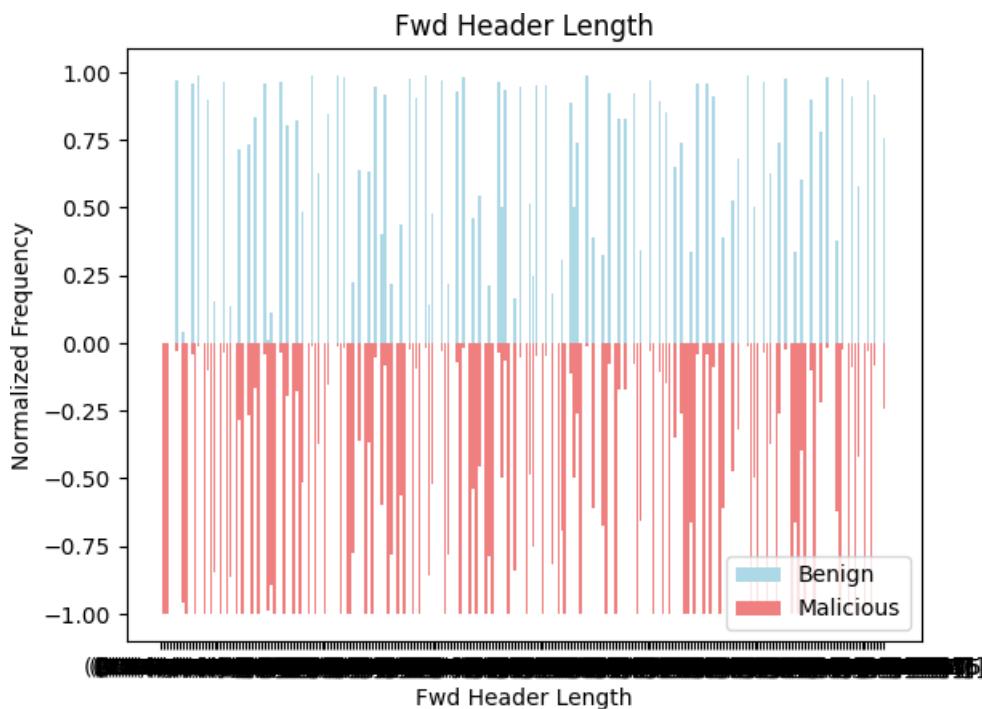


Figure 157: Pyramid chart of Fwd Header Length w.r.t isMalicious

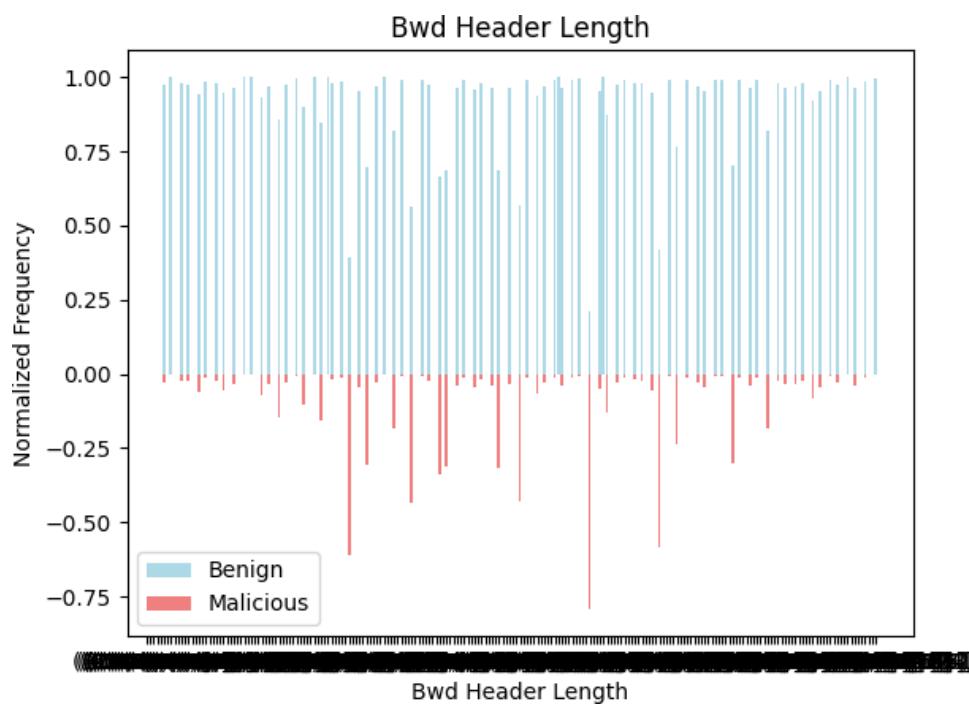


Figure 158: Pyramid chart of Bwd Header Length w.r.t isMalicious

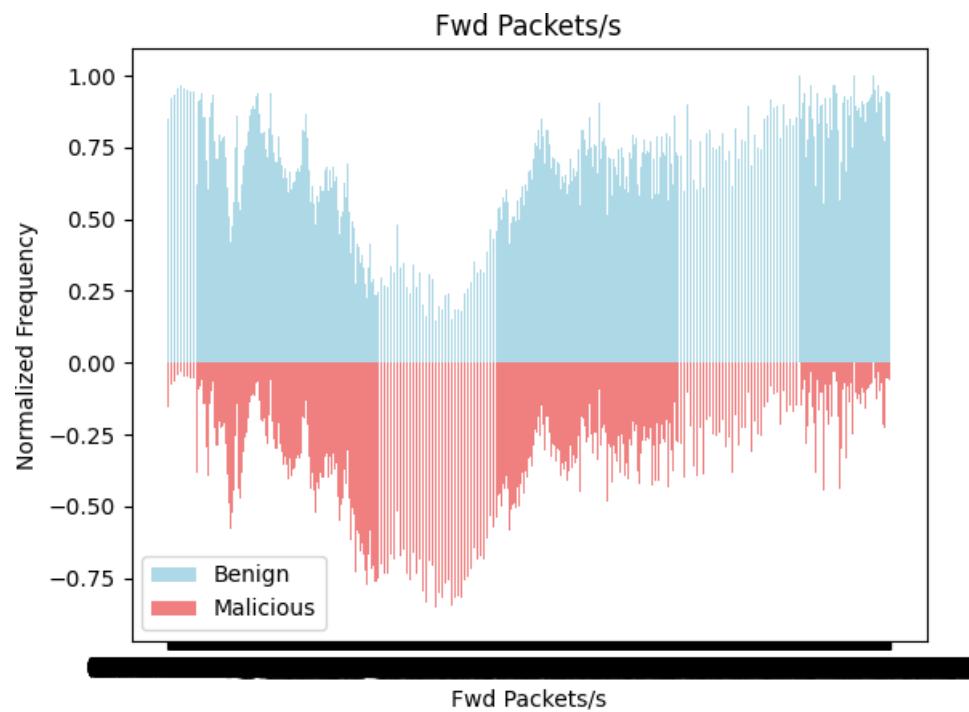


Figure 159: Pyramid chart of Fwd Packets/s Length w.r.t isMalicious

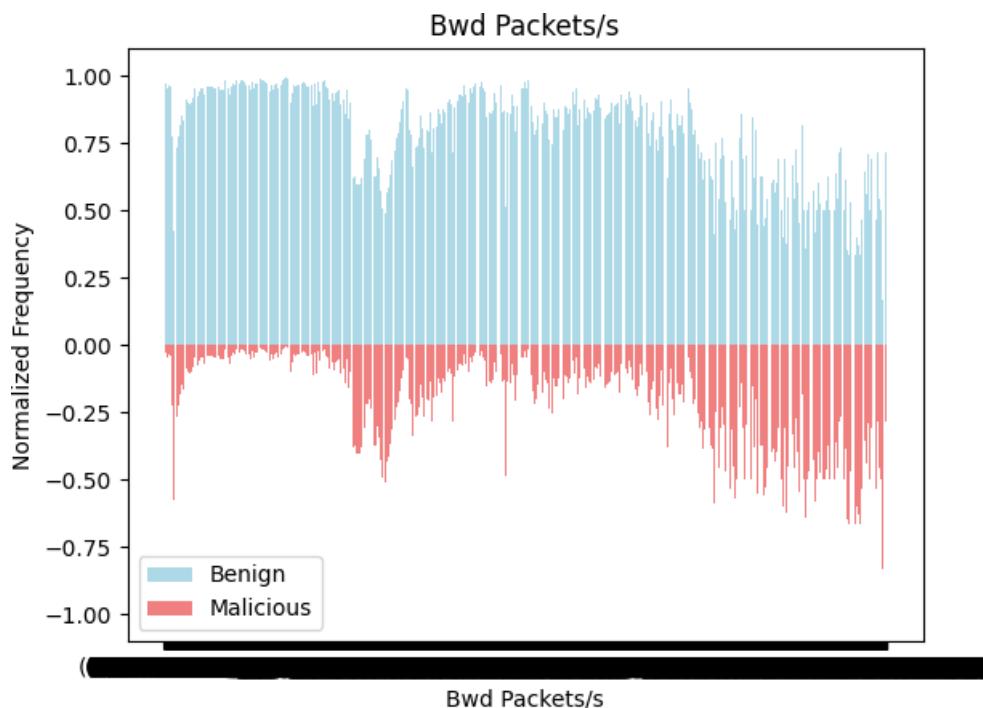


Figure 160: Pyramid chart of Bwd Packets/s Length w.r.t isMalicious

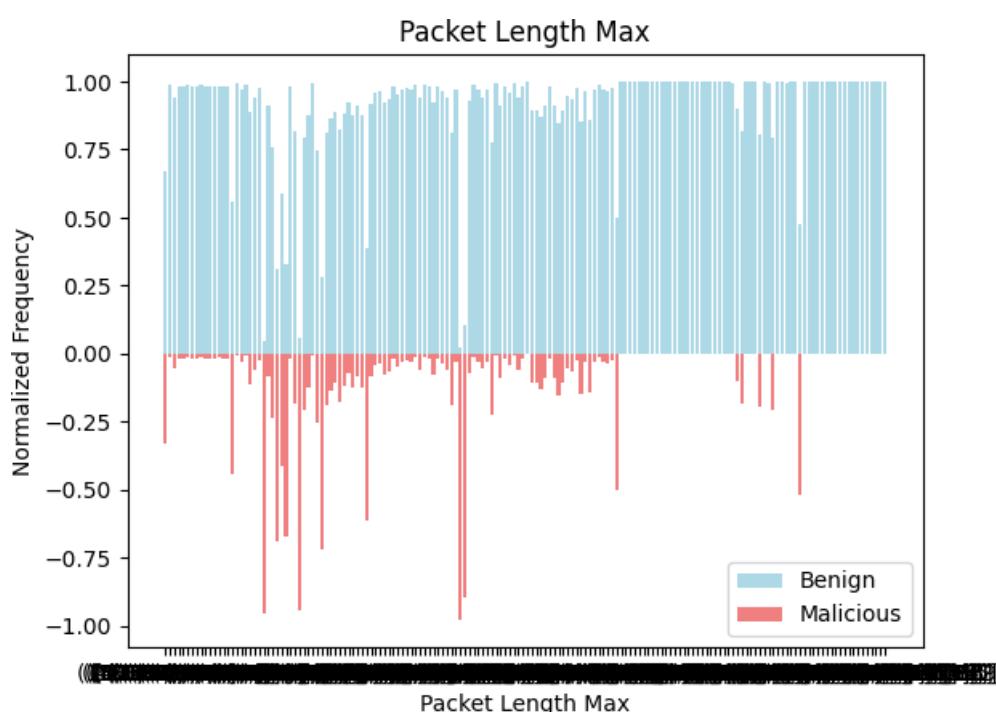


Figure 161: Pyramid chart of Packet Length Max w.r.t isMalicious

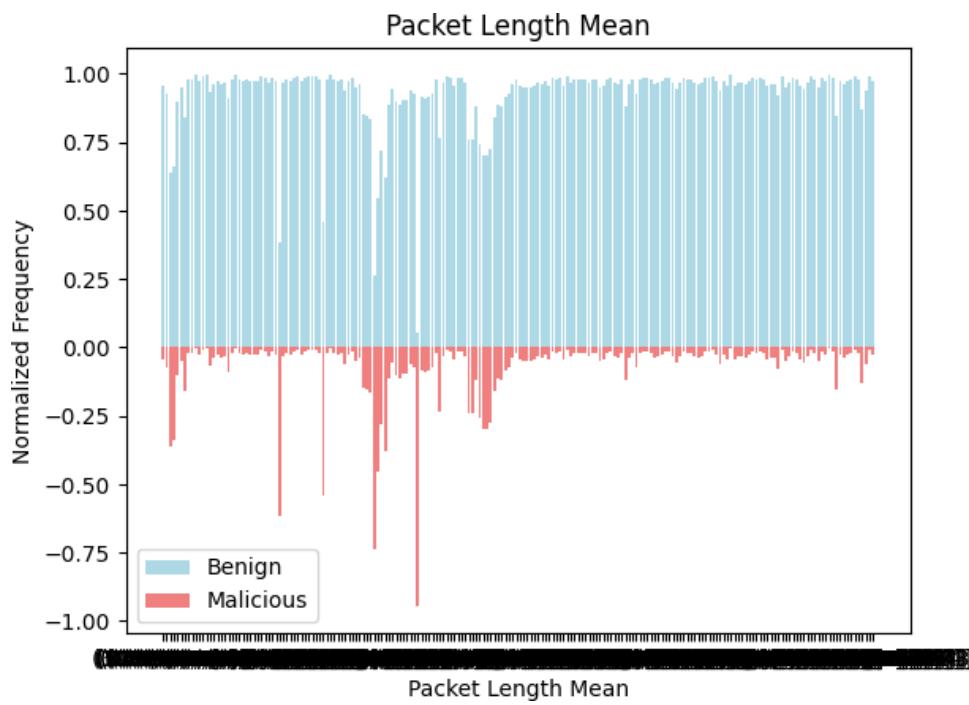


Figure 162: Pyramid chart of Packet Length Mean w.r.t isMalicious

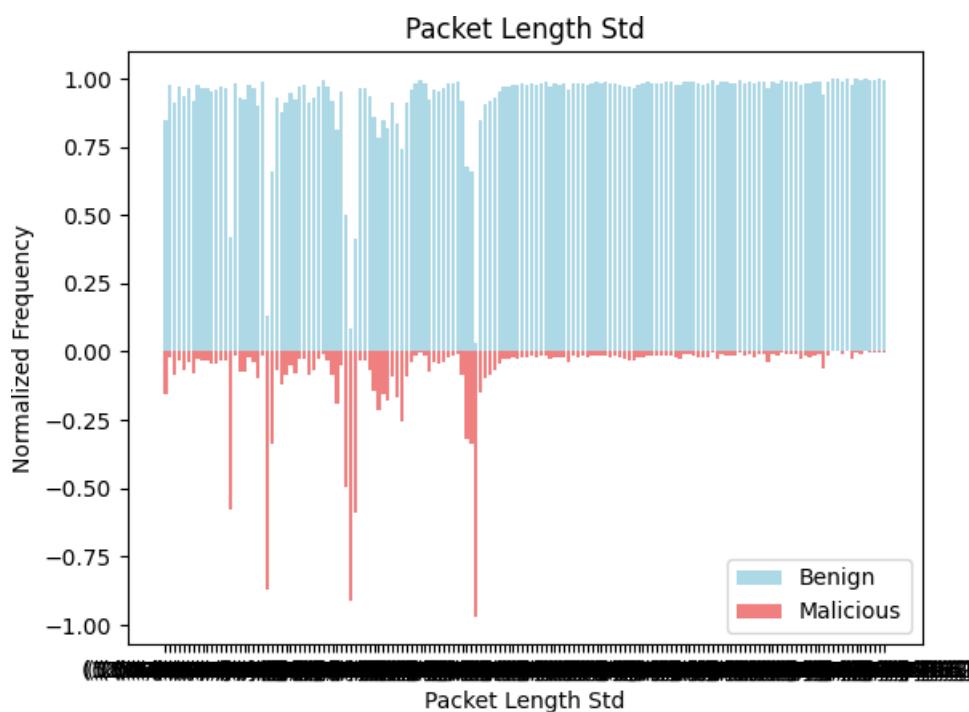


Figure 163: Pyramid chart of Packet Length Std w.r.t isMalicious

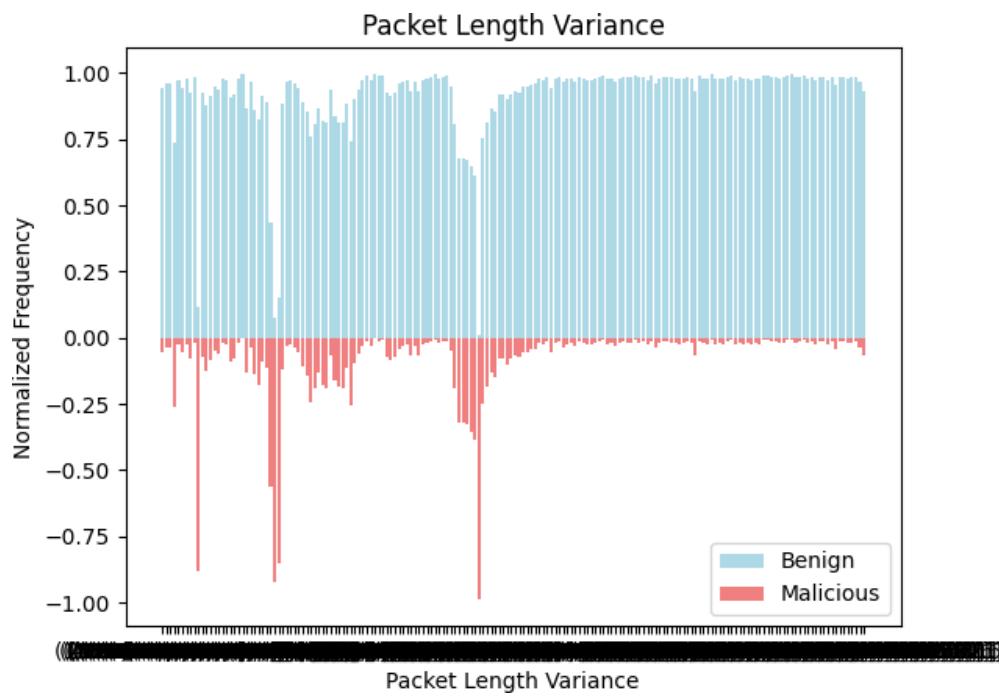


Figure 164: Pyramid chart of Packet Length Variance w.r.t isMalicious

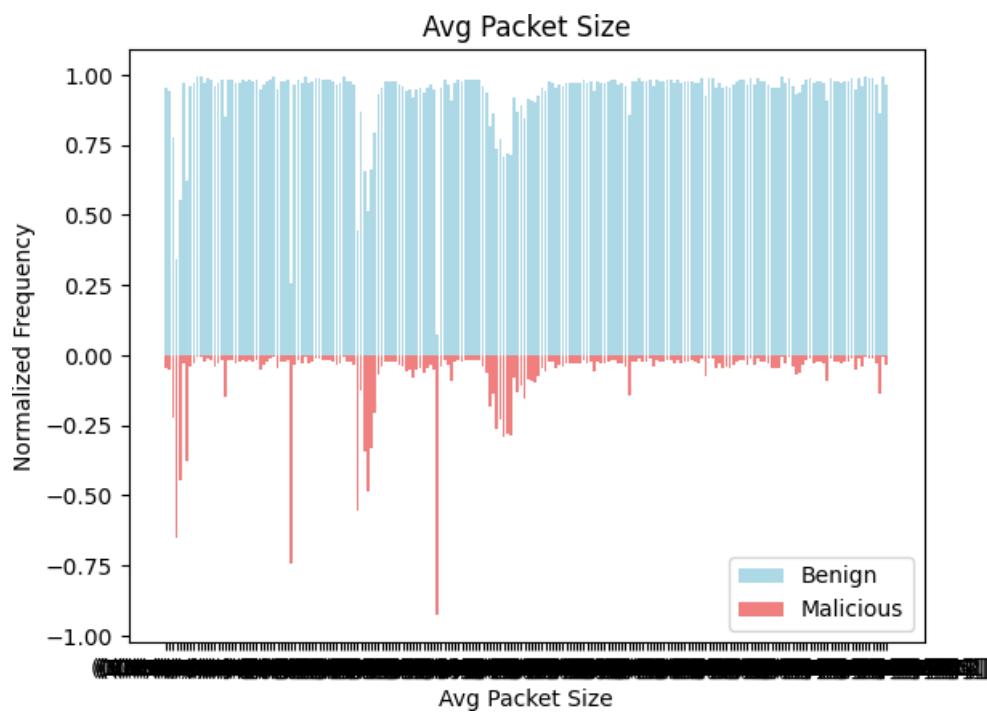


Figure 165: Pyramid chart of Avg Packet Size w.r.t isMalicious

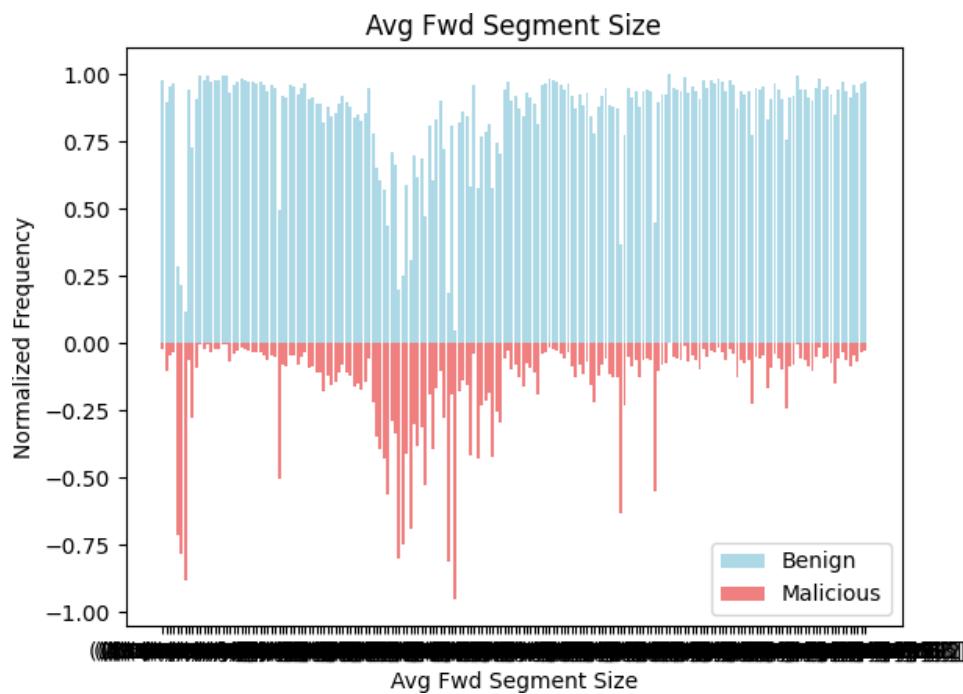


Figure 166: Pyramid chart of Avg Fwd Segment Size w.r.t isMalicious

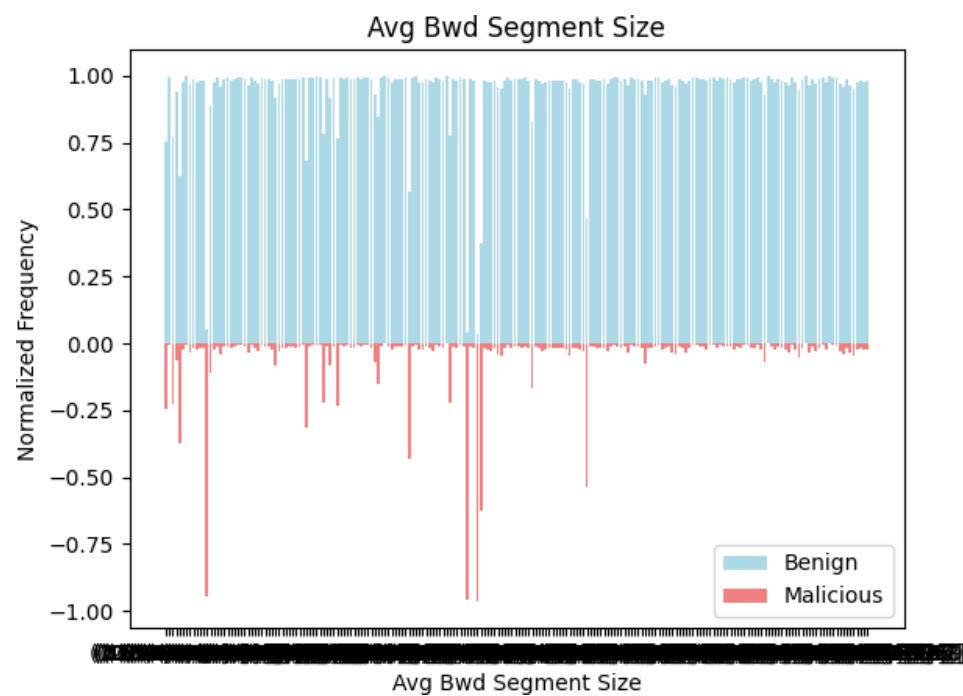


Figure 167: Pyramid chart of Avg Bwd Segment Size w.r.t isMalicious

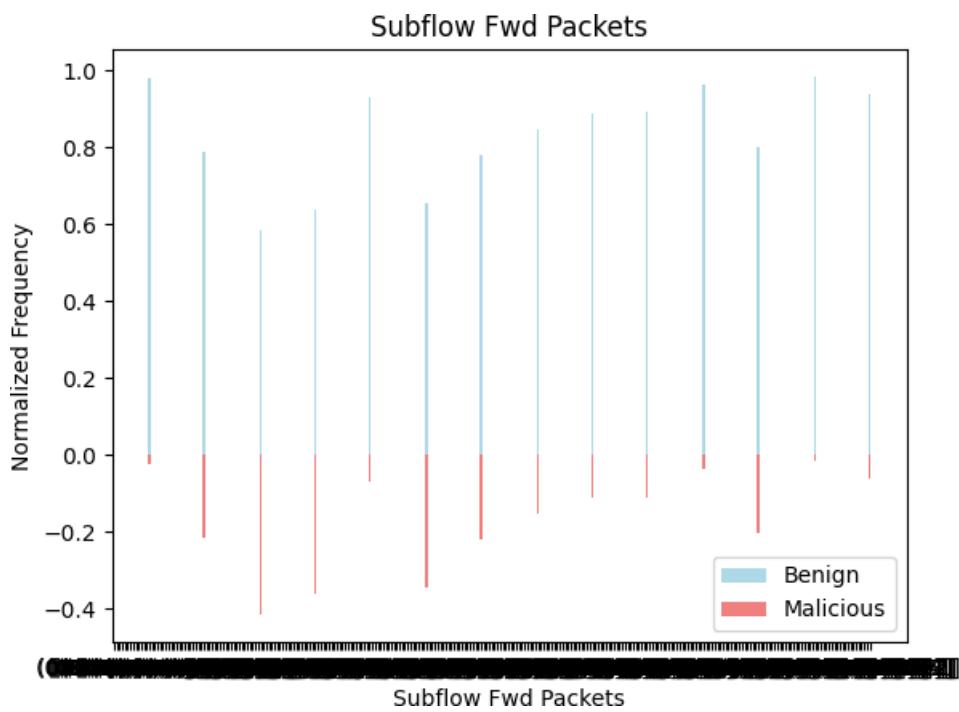


Figure 168: Pyramid chart of Subflow Fwd Packets w.r.t isMalicious

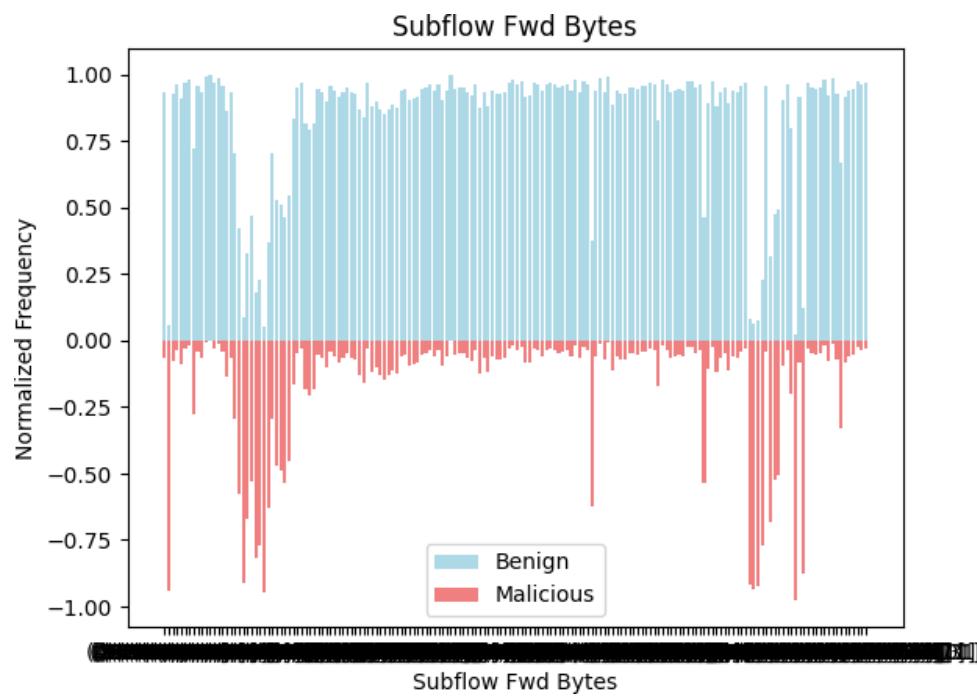


Figure 169: Pyramid chart of Subflow Fwd Bytes w.r.t isMalicious

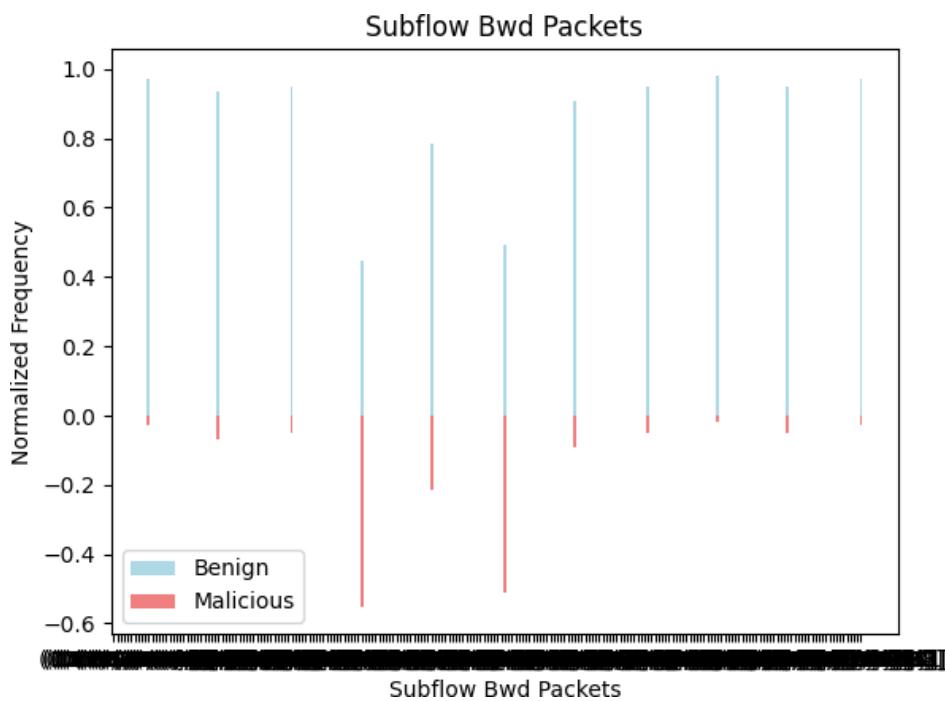


Figure 170: Pyramid chart of Subflow Bwd Packets w.r.t isMalicious

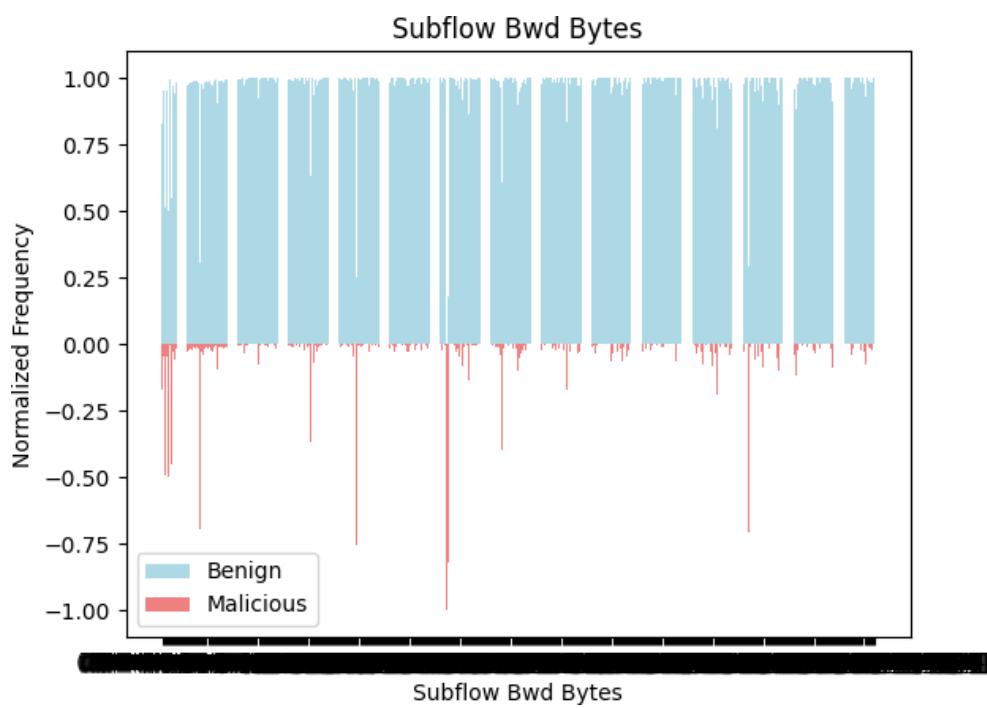


Figure 171: Pyramid chart of Subflow Bwd Bytes w.r.t isMalicious

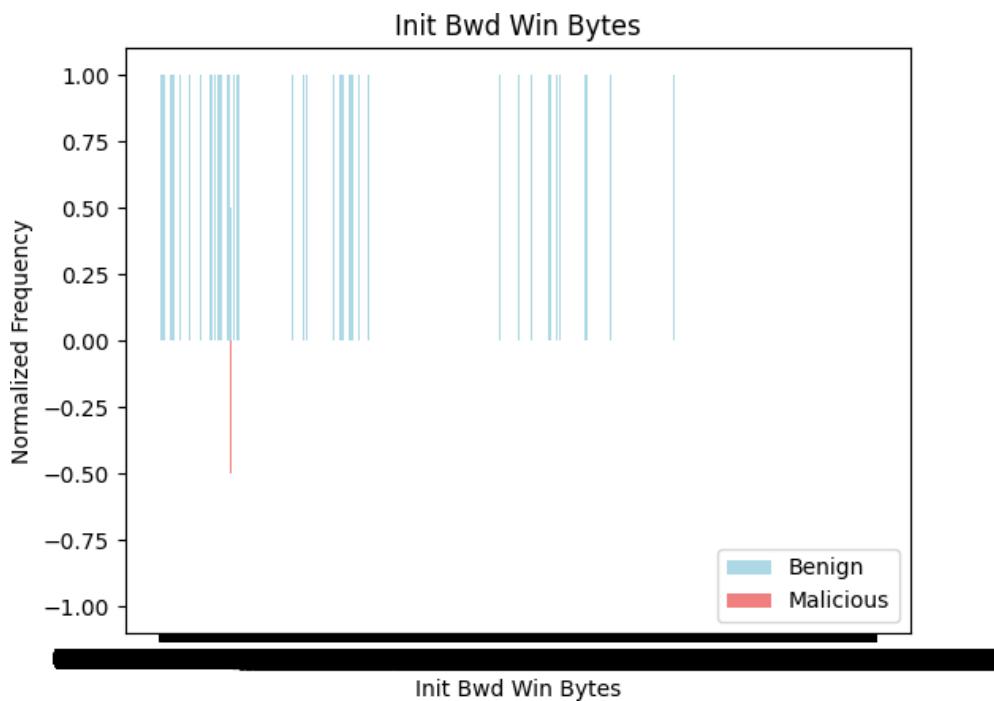


Figure 172: Pyramid chart of Init Bwd Win Bytes w.r.t isMalicious

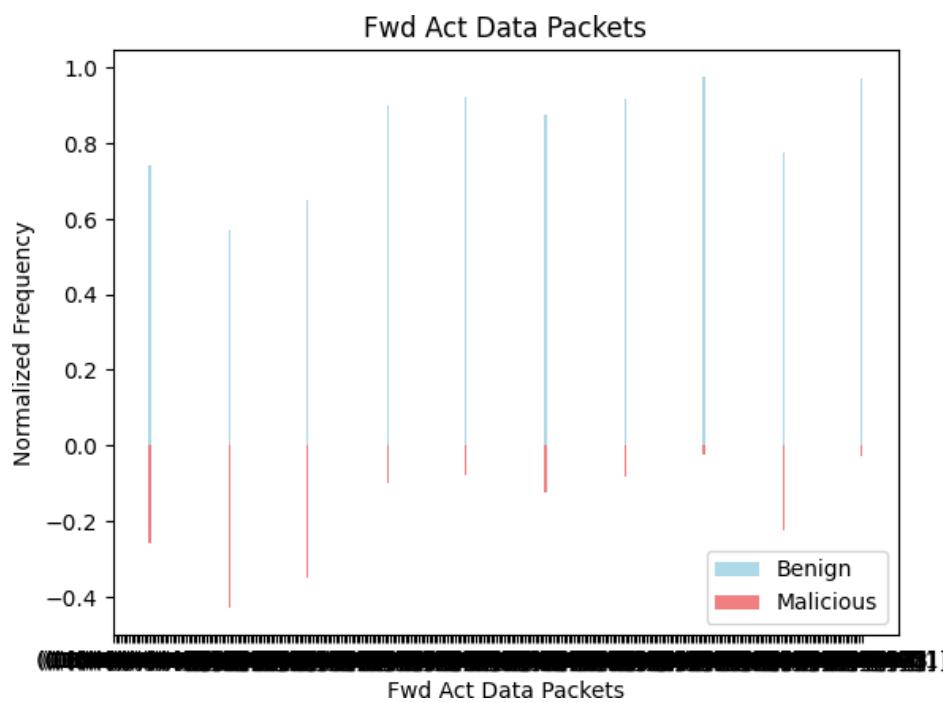


Figure 173: Pyramid chart of Fwd Act Data Packets w.r.t isMalicious

Following features have almost equal number of Malicious and Benign records in most of the bins: -

1. Flow Duration
2. Flow IAT Max
3. Fwd Header Length

Following features have some bins where number of Malicious records are relatively more than the number of Benign records and thus, change in patterns were observed over a set of bins: -

1. Flow Bytes/s
2. Flow Packets/s
3. Flow IAT Std
4. Fwd IAT Max
5. Bwd IAT Std
6. Bwd IAT Max
7. Fwd Packets/s
8. Bwd Packets/s

'Init Bwd Win Bytes' was a rare feature which had only 1 bin with Malicious records and rest all bins had Benign records.

Remaining all features have relatively very high number of Benign records compared to Malicious records in most of the bins.

While carrying out the above interpretation small variations and changes were not recorded as decisions based on minor changes may result in incorrect analysis. Only the patterns which were thick and broadly visible were recorded from Pyramid charts plotted with respect to target binary feature: isMalicious.

Label encoding: -

Label encoding on target feature: ClassLabel was done and results were stored in a new feature: attack_id. Thus, after label encoding: -

Table 6: Encoded values of ClassLabel

ClassLabel	attack_id
Benign	0
Botnet	1
Bruteforce	2
DDoS	3
DoS	4
Infiltration	5
Portscan	6
Webattack	7

Correlation matrix: -

Using the sampled dataset, correlation matrix could not get plotted due to limitations of system's configurations, which led to insufficient memory error.

As the result, 20% of records from sampled dataset were taken and used to plot the heat map for correlation matrix, with target feature as attack_id.

Shape of the new sub-sampled dataset on which correlation matrix was computed: (366690, 47).

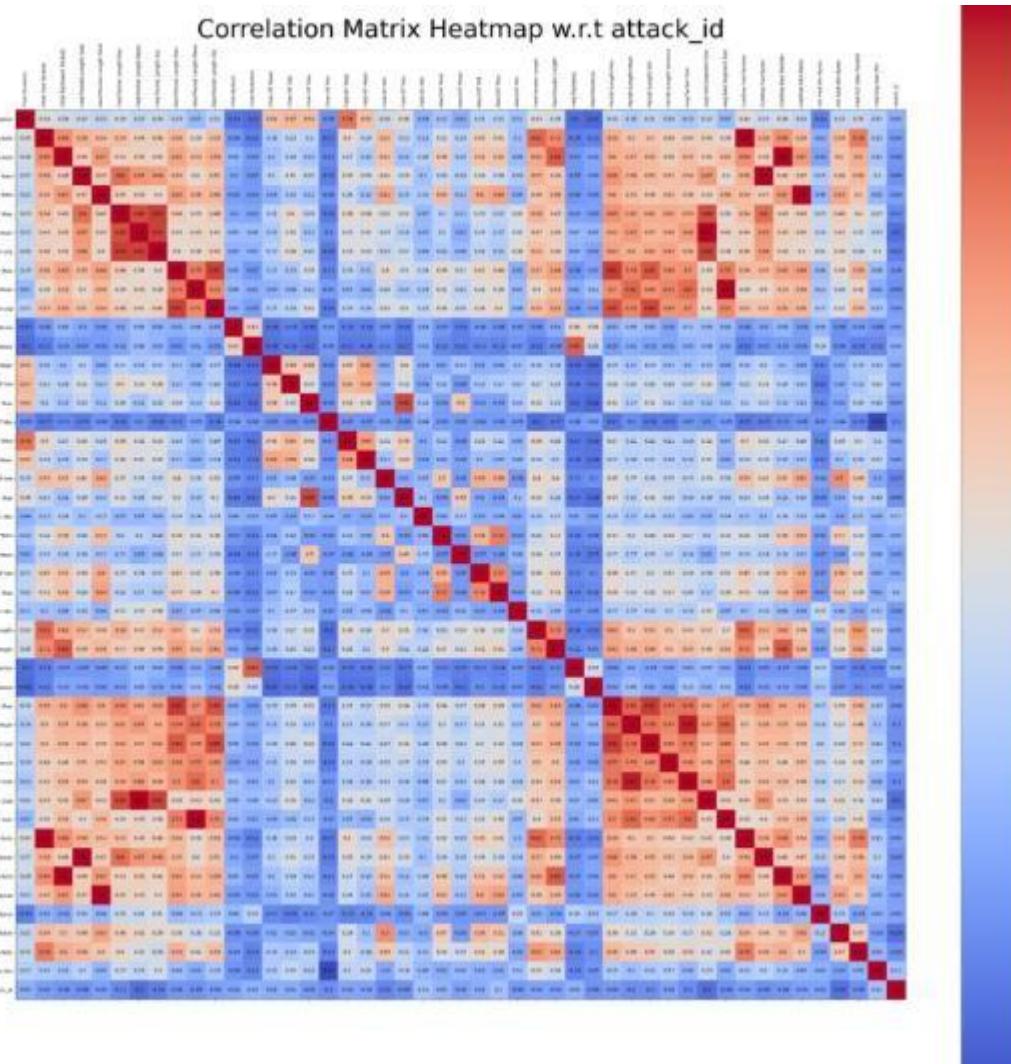


Figure 174: Correlation matrix based on sub-sampled dataset

Observations from the above heat map: -

1. Many independent features have red and dark red squares, indicating strong relation among them.

2. Some of the examples are: -

Fwd Packet Length Max – Fwd Packet Length Mean = 0.88

Fwd Packet Length Max – Fwd Packet Length Std = 0.91

Bwd Packet Length Std – Packet Length Max = 0.91

Bwd Packet Length Std – Packet Length Mean = 0.71

3. All independent features have weak relation with attack_id.

However, due to sub-sampled dataset used for plotting the correlation matrix, it was difficult to determine whether to use the results observed in correlation matrix on the main dataset or on the sampled dataset. As the result, the results of the above heat map were not used.

Renaming of feature names: -

The columns were renamed for ease of use by replacing space with underscore.

Analysis based on descriptive statistics: -

Three lists were created: -

1. columns_equal_min_and_Q1 = Features with equal minimum and Q1 value.
2. columns_equal_Q1_and_Q3 = Features with equal Q1 and Q3 value.
3. columns_equal_Q3_and_max = Features with equal Q3 and maximum value.

Following features were captured in columns_equal_min_and_Q1: -

1. Bwd_Packet_Length_Total
2. Fwd_Packet_Length_Std
3. Bwd_Packet_Length_Max
4. Bwd_Packet_Length_Mean
5. Bwd_Packet_Length_Std
6. Flow_IAT_Std
7. Fwd_IAT_Std
8. Bwd_IAT_Total
9. Bwd_IAT_Mean
10. Bwd_IAT_Std
11. Bwd_IAT_Max
12. Bwd_IAT_Min
13. Avg_Bwd_Segment_Size
14. Subflow_Bwd_Bytes
15. Fwd_Act_Data_Packets

Thus, for the above list of features it was inferred that a large number of records are clustered in lower range.

These features may have many zero values or many constant values in lower range of data points. For all of the 15 features, minimum value and Q1 value equal to 0.0

Thus, 25% of the values are zero in all of the 15 features. And thus the features may also be categorized as Zero-inflated features due to their high percentage of zero values.

Since the features are having data concentrated in lower range, they are positively skewed.

The results were grouped into two categories: Non-zero, Zero.

1. Non-zero: Data points having value not equal to 0.
2. Zero: Data points having value equal to 0.

Based on the above two categories, the frequency of data points with respect to the target binary feature: isMalicious was plotted for each feature.

Comparison of isMalicious for Zero and Non-Zero Bwd_Packets_Length_Total

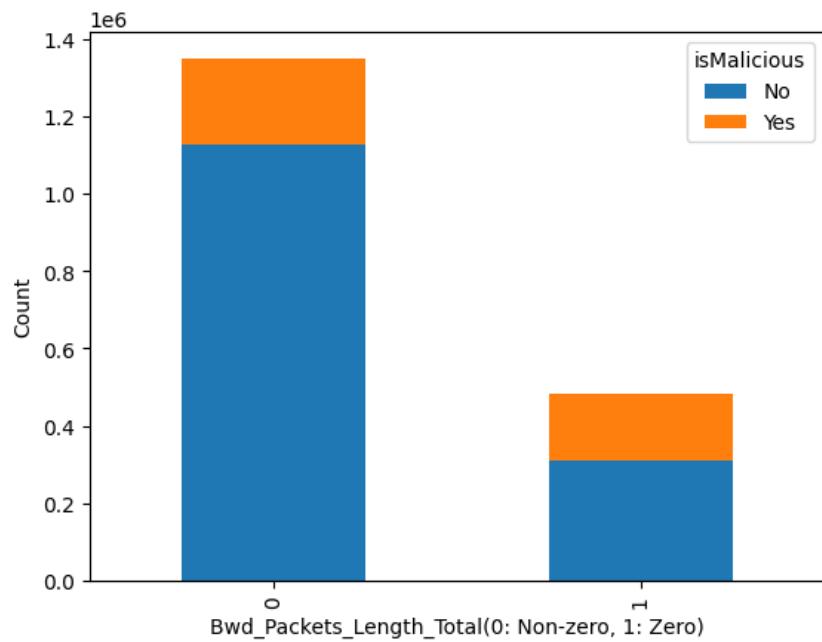


Figure 175: Stacked bar chart for Bwd Packets Length Total w.r.t isMalicious

Comparison of isMalicious for Zero and Non-Zero Fwd_Packet_Length_Std

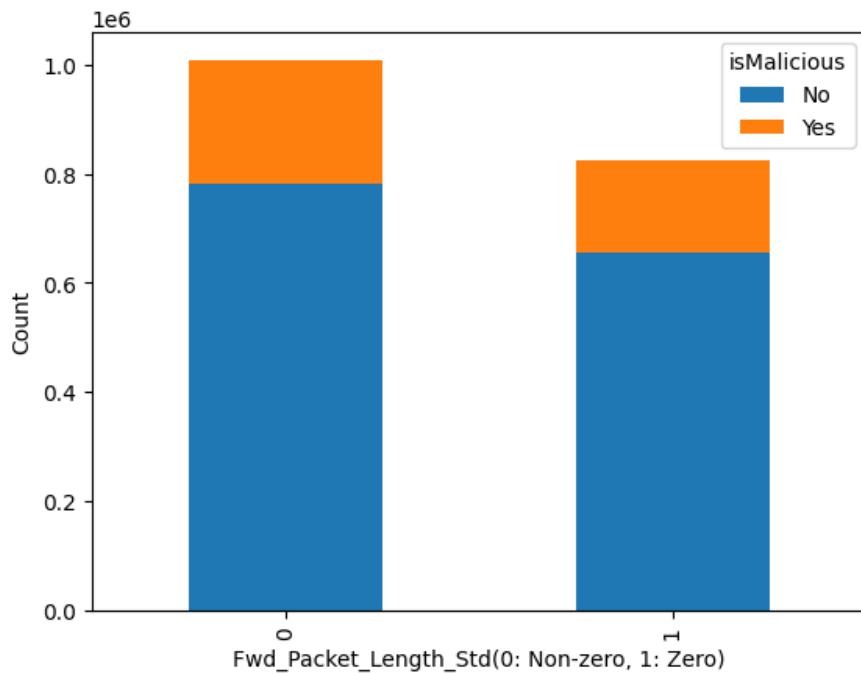


Figure 176: Stacked bar chart for Fwd Packet Length Std w.r.t isMalicious

Comparison of isMalicious for Zero and Non-Zero Bwd_Packet_Length_Max

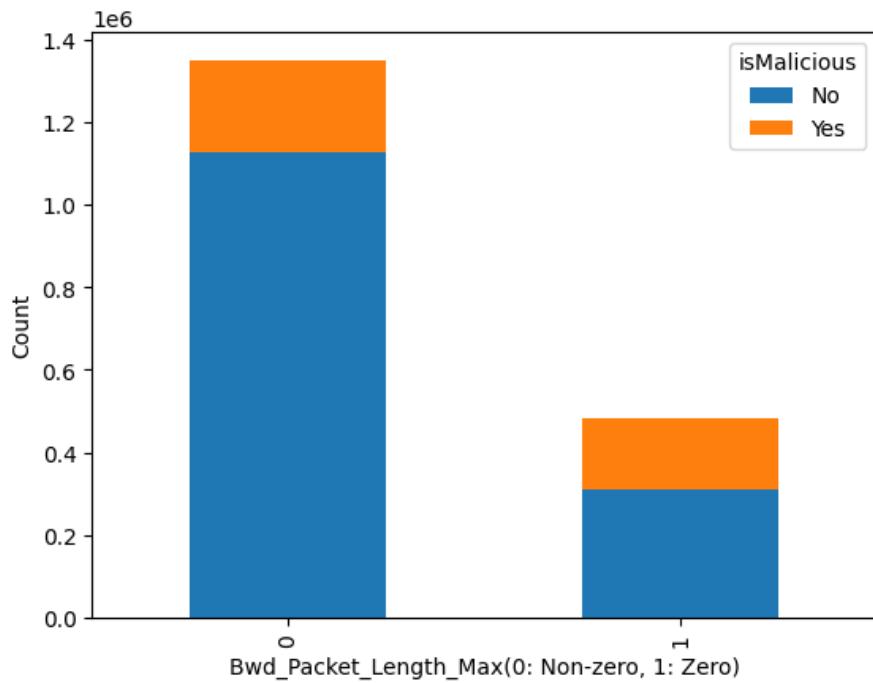


Figure 177: Stacked bar chart for Bwd Packet Length Max w.r.t isMalicious

Comparison of isMalicious for Zero and Non-Zero Bwd_Packet_Length_Mean

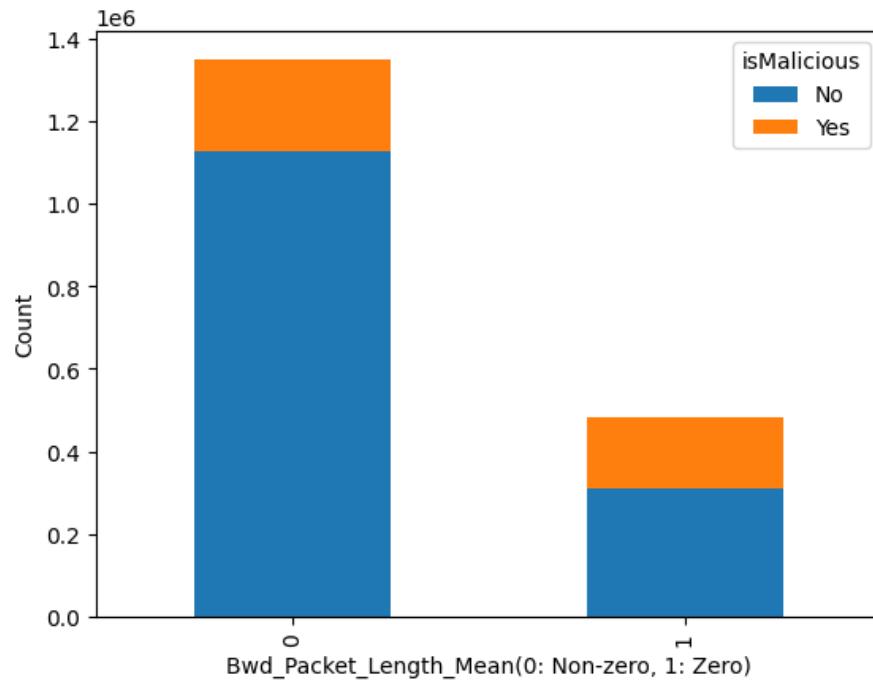


Figure 178: Stacked bar chart for Bwd Packet Length Mean w.r.t isMalicious

Comparison of isMalicious for Zero and Non-Zero Bwd_Packet_Length_Std

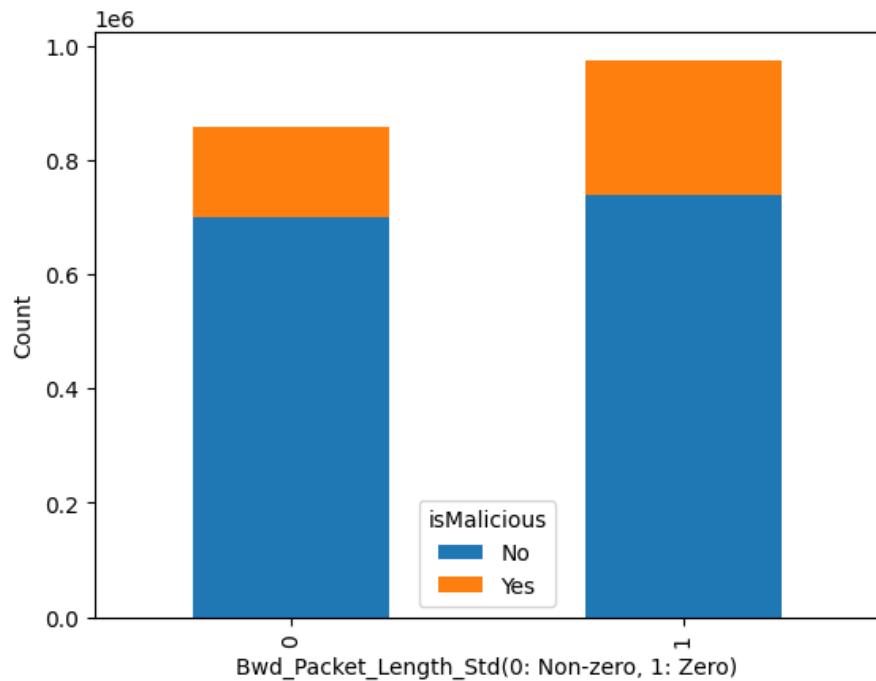


Figure 179: Stacked bar chart for Bwd Packet Length Std w.r.t isMalicious

Comparison of isMalicious for Zero and Non-Zero Flow_IAT_Std

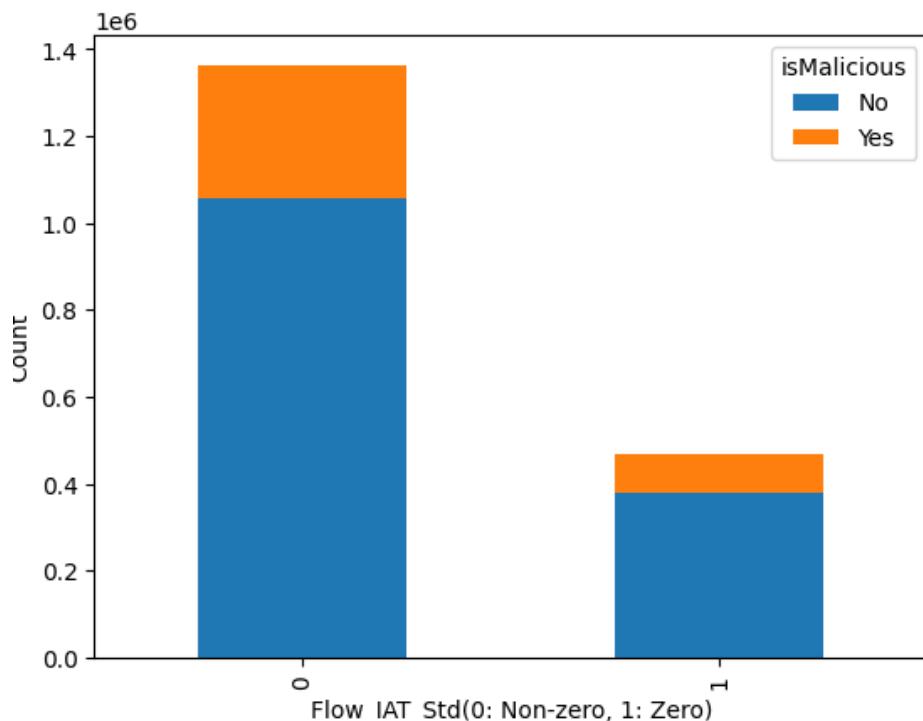


Figure 180: Stacked bar chart for Flow IAT Std w.r.t isMalicious

Comparison of isMalicious for Zero and Non-Zero Fwd_IAT_Std

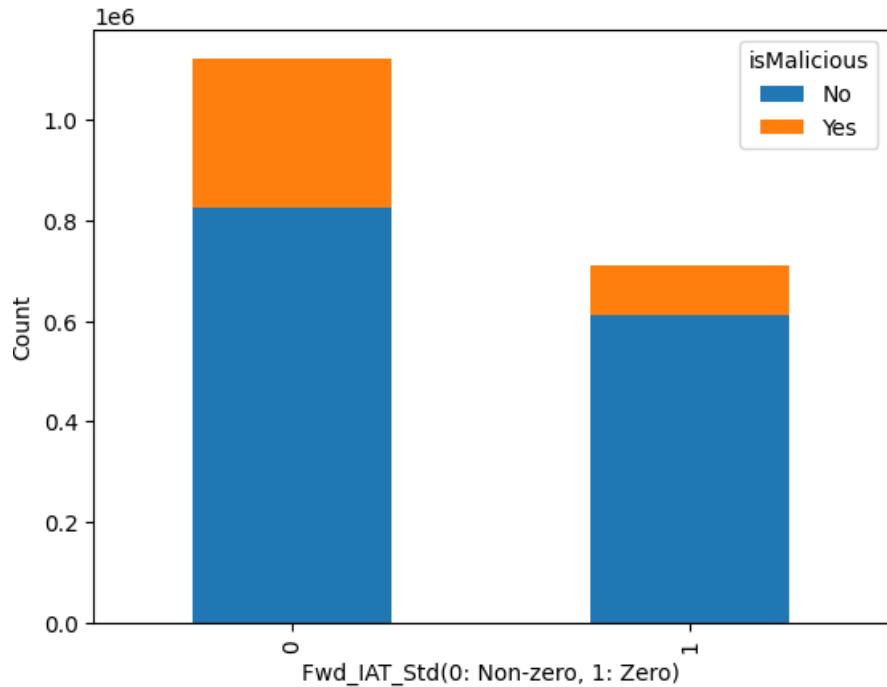


Figure 181: Stacked bar chart for Fwd IAT Std w.r.t isMalicious

Comparison of isMalicious for Zero and Non-Zero Bwd_IAT_Total

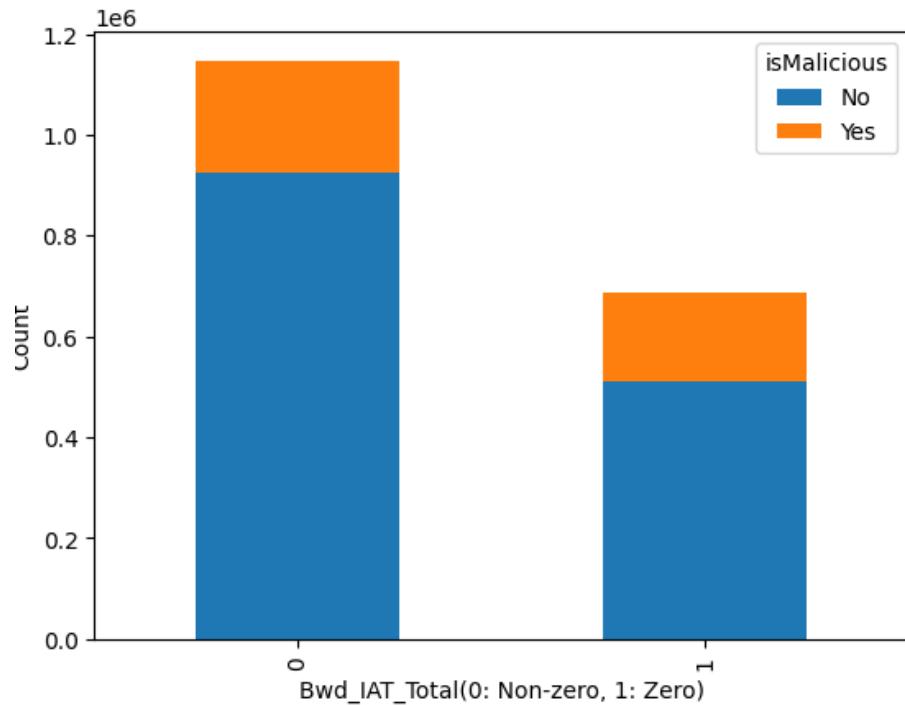


Figure 182: Stacked bar chart for Bwd IAT Total w.r.t isMalicious

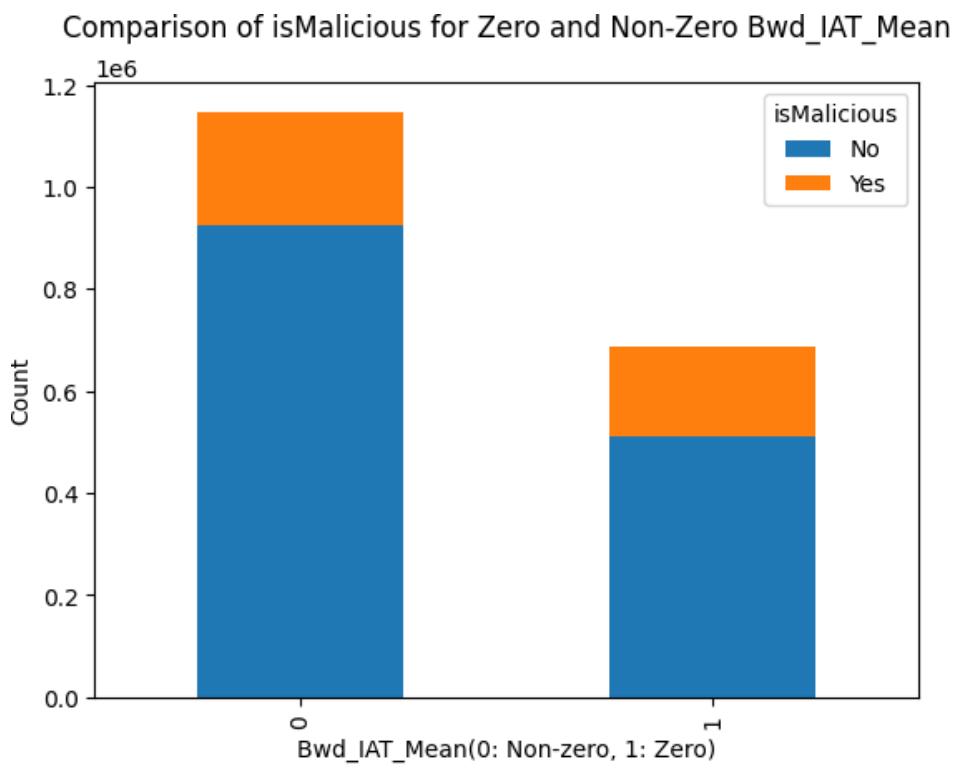


Figure 183: Stacked bar chart for Bwd IAT Mean w.r.t isMalicious

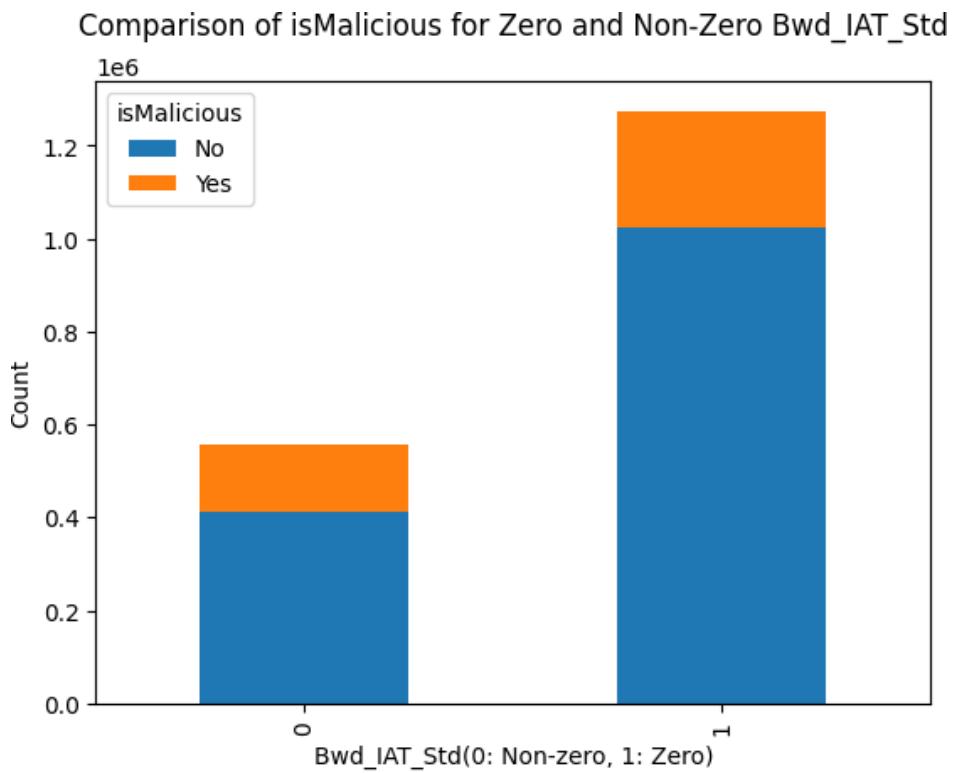


Figure 184: Stacked bar chart for Bwd IAT Std w.r.t isMalicious

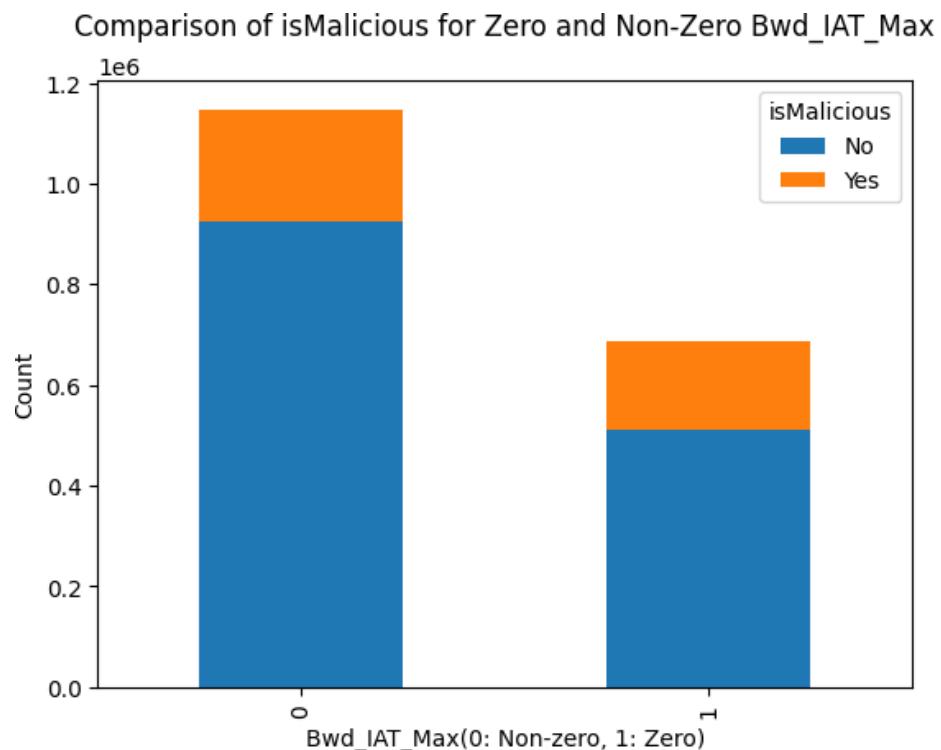


Figure 185: Stacked bar chart for Bwd IAT Max w.r.t isMalicious

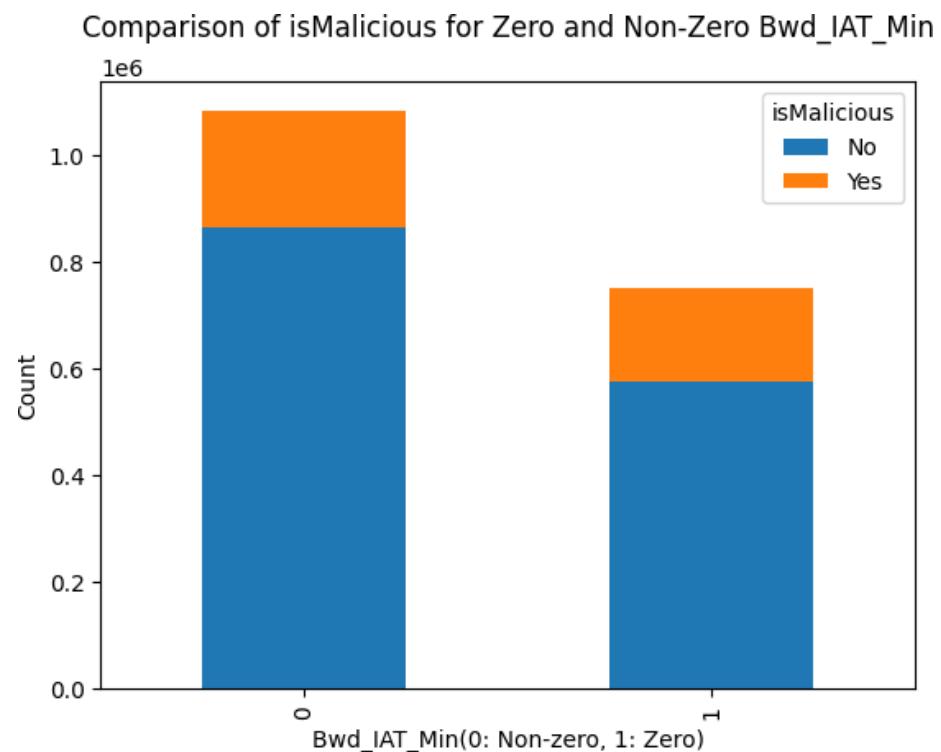


Figure 186: Stacked bar chart for Bwd IAT Min w.r.t isMalicious

Comparison of isMalicious for Zero and Non-Zero Avg_Bwd_Segment_Size

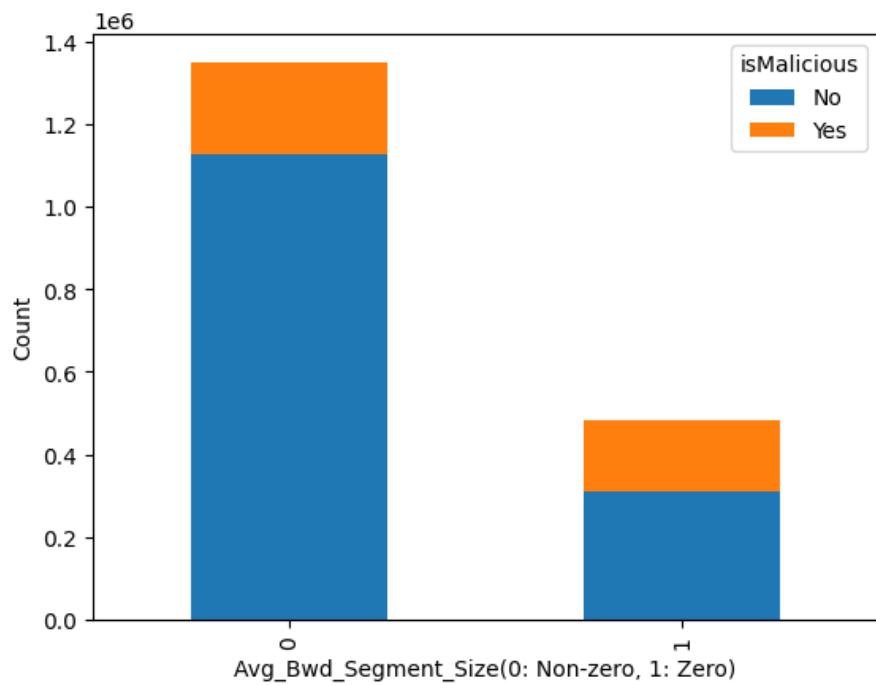


Figure 187: Stacked bar chart for Avg Bwd Segment Size w.r.t isMalicious

Comparison of isMalicious for Zero and Non-Zero Subflow_Bwd_Bytes

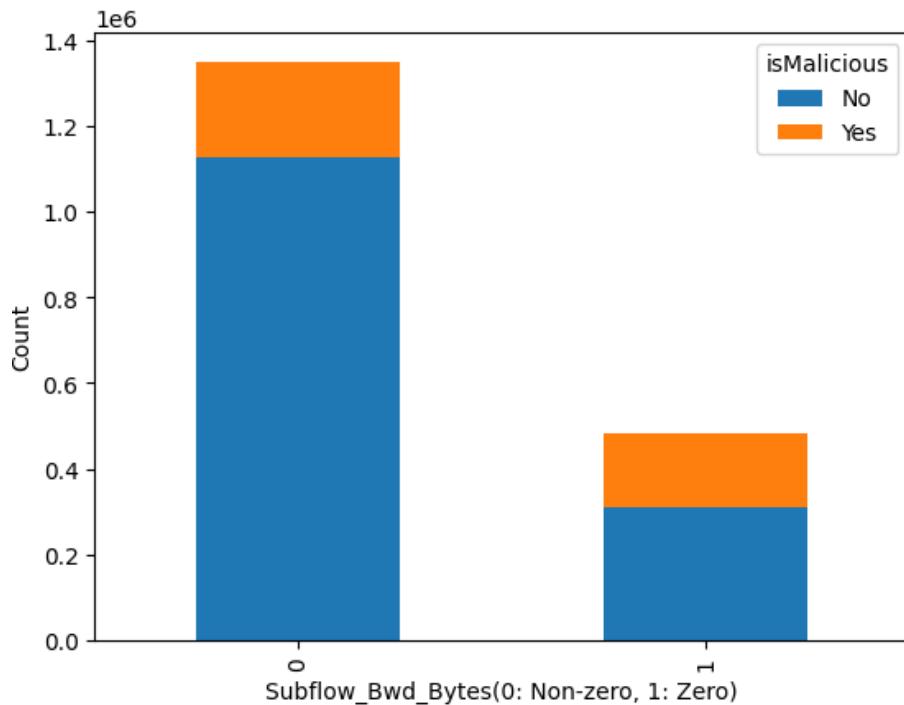


Figure 188: Stacked bar chart for Subflow Bwd Bytes w.r.t isMalicious

Comparison of isMalicious for Zero and Non-Zero Fwd_Act_Data_Packets

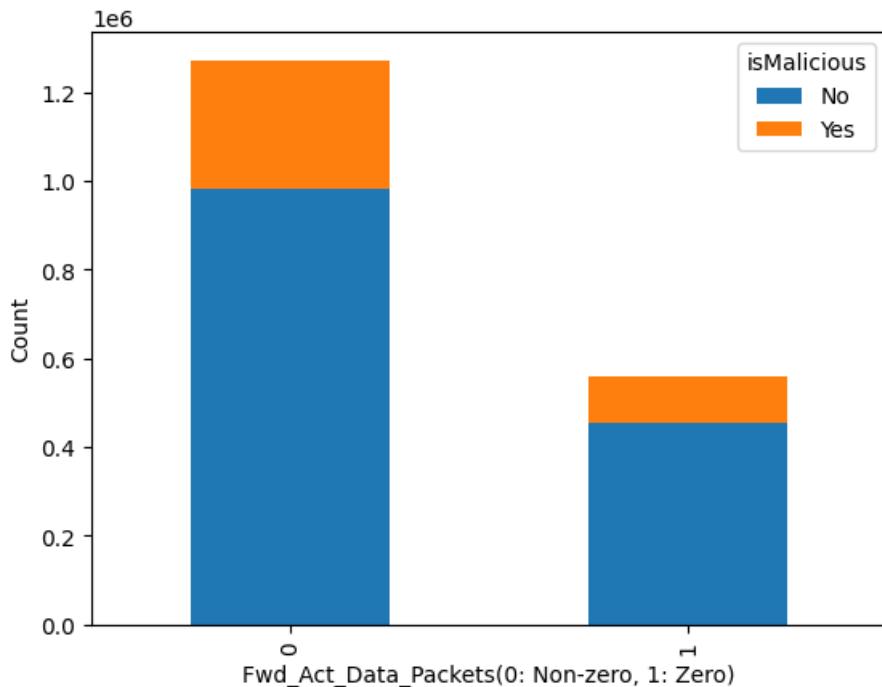


Figure 189: Stacked bar chart for Fwd Act Data Packets w.r.t isMalicious

There was no differentiation found among the 15 features to get more information to identify malicious events based on zero and non-zero values.

Following features were captured in columns_equal_Q1_and_Q3: -

1. Init_Fwd_Win_Bytes
2. Fwd_Seg_Size_Min

Thus, for the above list of features it was inferred that 50% of the datapoints are clustered at a single value.

Init_Fwd_Win_Bytes: Q1=Q2=Q3=8192.0

Fwd_Seg_Size_Min: Q1=Q2=Q3=20.0

These features may have very low variability and many constant values.

The results were grouped into two categories: Not mid-range, Mid-range.

Not mid-range: Data points having value not equal to median (Q2).

Mid-range: Data points having value equal to median (Q2).

Based on the above two categories, the frequency of data points with respect to the target binary feature: isMalicious was plotted for each feature.

Comparison of isMalicious for Mid-range and Non-mid-range Init_Fwd_Win_Bytes

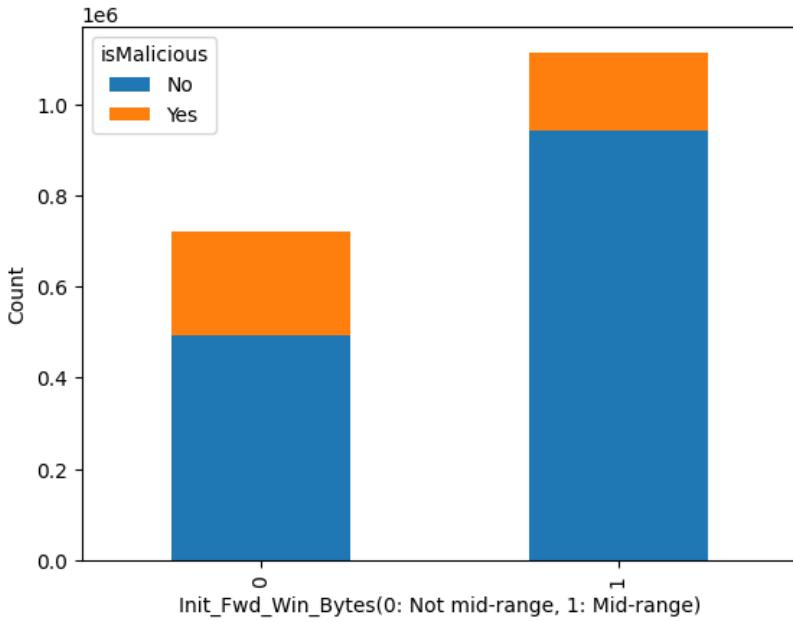


Figure 190: Stacked bar chart for Init Fwd Win Bytes w.r.t isMalicious

Comparison of isMalicious for Mid-range and Non-mid-range Fwd_Seg_Size_Min

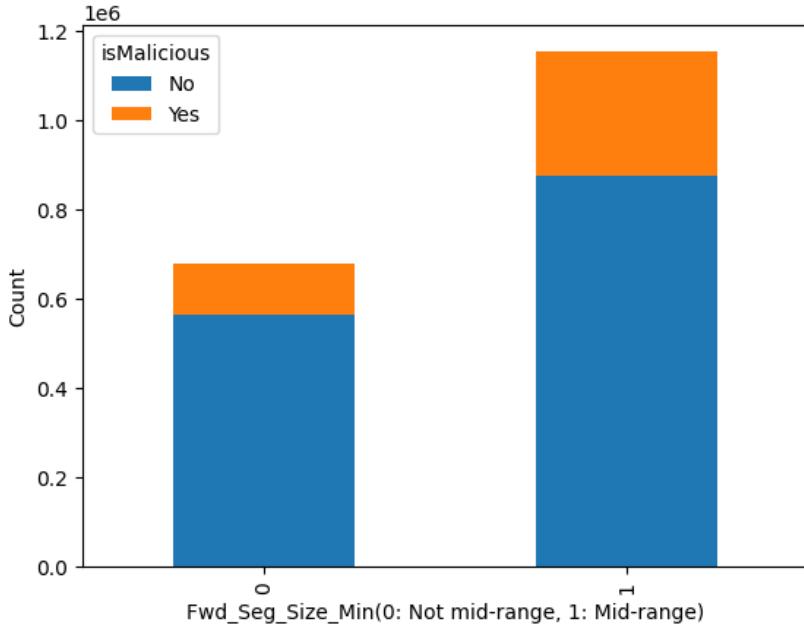


Figure 191: Stacked bar chart for Fwd Seg Size Min w.r.t isMalicious

There was no differentiation found among the 2 features to get more information to identify malicious events based on mid-range and not mid-range values.

No features were captured in columns_equal_Q3_and_max.

Thus, from the above it was inferred that all features in upper range have high variability and are

spread out.

As the result, there are no negatively skewed features in the dataset.

Fetching the most frequent category of records: -

Number of records for each category of event in ClassLabel were fetched in the sampled dataset:

Benign : 1437467

DDoS : 246982

DoS : 79186

Botnet : 29348

Bruteforce : 20546

Infiltration : 18870

Webattack : 625

Portscan : 430

It was checked if all records of each category are unique or duplicate.

1437467 records for Benign were duplicate.

246982 records for DDoS were duplicate.

79186 records for DoS are duplicate.

18870 records for Infiltration are duplicate.

625 records for Webattack are duplicate.

29348 records for Botnet are unique.

20546 records for Bruteforce are unique.

430 records for Portscan are unique.

A subset of data was selected: -

1. Top two categories having duplicate records: Benign, DDoS.
2. Top two categories having unique records: Botnet, Bruteforce.

Reason: -

1. Given the size of sampled dataset, it becomes extremely difficult to perform further processing and tasks such as feature selection and training the model.
2. Webattack and Portscan have too less number of records compared to other categories for training the model. Thus, it becomes extremely difficult to have a common model that can be trained to identify events with vast difference in frequency.

New shape of the sampled dataset: (1734343, 49).

Due to limitations of system's configurations and memory, the sampled dataset will be used further for training the model.

Label encoding: -

Since the label encoding was previously performed on ClassLabel and stored the results in attack_id, after dropping the rows, the encoded values will have gap.

Thus, the attack_id column was dropped, label encoding was again performed on ClassLabel, and new results were stored in attack_id.

Table 7: Encoded values of ClassLabel after dropping the rows

ClassLabel	attack_id
Benign	0
Botnet	1
Bruteforce	2
DDoS	3

The column ClassLabel was dropped, since its equivalent numerical feature is available in the form of attack_id.

Thus, at this stage the two target features in the dataset are: -

- isMalicious: For binary classification
- attack_id: For multi-class classification

Writing the records into a new file: -

The dataset was written in a new file: processed_dataset.parquet

This will allow to perform further activities on a new notebook and prevent the overhead of loading the complete dataset, and running all the tasks performed for pre-processing, analysis and feature engineering.

Handling imbalanced nature of dataset: -

Two types of classification models need to be built: -

1. binary_cic_df
2. multiclass_cic_df

For binary_cic_df: -

Table 8: Distribution of records in sampled dataset based on isMalicious

isMalicious	Number of records
0	1437467

1	296876
---	--------

For multiclass_cic_df: -

Table 9: Distribution of records in sampled dataset based on attack_id

attack_id	Number of records
0	1437467
1	29348
2	20546
3	246982

Since dataset is imbalanced, building classifier models using it will result in: -

1. Biased predictions
2. Low sensitivity
3. Poor generalization

Approaches to handle imbalanced nature of dataset: -

1. Oversampling of minority class
 - a. Here we create duplicate of records having minority class and make the count same as majority class.
2. Undersampling of majority class
 - a. Here we reduce the records of majority class and make the count same as minority class.
3. Cost sensitive learning
 - a. The algorithm is forced to correctly identify minority class by adding penalty for incorrect classification.
4. Anomaly detection approach
 - a. The minority class is treated as an anomaly and majority class is treated as baseline.
 - b. Algorithms that can help with this approach: -
 - i. Isolation forest
 - ii. One-class SVM

Chapter 5

Research about optimization algorithms for feature selection

Some of the heuristic algorithms that were studied and can be used for feature selection are: -

1. Particle Swarm Optimization (PSO)
2. Artificial Immune System (AIS)
3. Artificial Bee Colony optimization (ABC)

Following algorithms were explored in more detail to understand about their implementation for Feature selection: -

Particle Swarm Optimization: -

- It replicates the behaviour of a swarm of insects or a school of fish.
- It is motivated from foraging and social behaviour of swarms.
- The solutions are evaluated and they are compared and new solutions are generated in the process to find an optimal solution for the given problem.
- PSO starts with initializing population randomly.
- Each solution in PSO is referred to as particle.
- There are 3 distinct features of PSO: -
 - Best fitness of each particle.
 - Best fitness of swarm.
 - Velocity and position update of each particle.
- pbest(i): The best solution (fitness) achieved so far by particle i.
- gbest(i): The best solution (fitness) achieved so far by any particle in the swarm.
- Velocity and position update: For exploring and exploiting the search space to locate the optimal solution.
- Tuning parameters required for implementation: -
 - Population size
 - Termination criteria
 - Acceleration coefficients, c_1, c_2
 - Inertia

Working of PSO: -

1. We start with random initial population: $P(t)$
2. We evaluate the random initial population and assign fitness.
3. Then we enter decision box based on number of generations.
4. Update local best $p(i,lb)_t$ of each particle and find global best of the swarm $p(gb)_t$
5. Then we enter decision box with respect to number of particles. We perform operations for each particle.
6. The first operation performed on the particle is to update the velocity $v(i)t+1$.
7. The second operation performed on the particle is to update the position $x(i)t+1$.
8. The third operation performed on the particle is to evaluate the updated position.
9. Thus, by evaluating $x(i)t+1$ we assign fitness to it.
10. Then we increase the counter by 1 $\rightarrow l = i+1$.
11. Thus, the process will be repeated until we cover each particle.

12. As the result, we calculate updated velocity and updated position for each particle.
13. The process continues until we meet the termination condition $\rightarrow t > T$.
14. Once the termination condition gets satisfied, we terminate PSO and report the results.

Flowchart of Particle Swarm Optimization: -

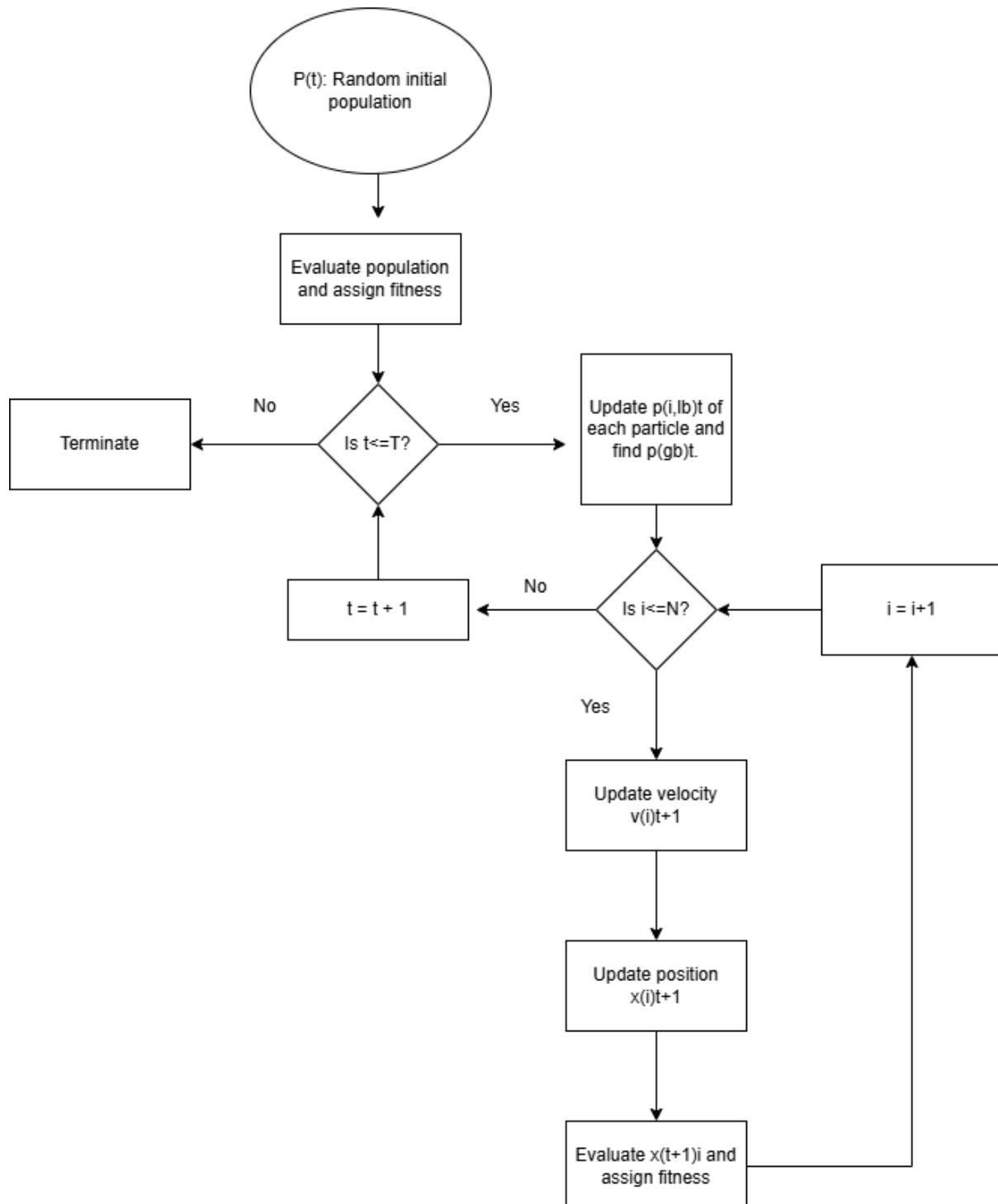


Figure 192: Flowchart for PSO algorithm

Using PSO algorithm for feature selection: -

1. Initialize the population
 - a. Creating a population of particles, where each particle represents a subset of features.
 - b. The initial population is generated randomly; thus, we fetch the initial subset of features using random approach.
2. Evaluate fitness
 - a. For each particle, its fitness is evaluated by training the model based on the subset of feature using training data.
 - b. After training the model, using validation data, the evaluation is carried out and scores such as accuracy, precision, recall and F1-score are computed.
3. Update personal best and global best
 - a. For each particle, personal best: pBest is updated if the current fitness is better than its previous best.
 - b. Based on the best personal best of all particles, if it exceeds the global best gBest, then gBest is updated.
4. Update velocity and position
 - a. Velocity and position of each particle are updated.
5. Repeat the process from step 2 to 4 for T iterations.
6. Select the best subset of features

Artificial Bee Colony optimization: -

- It is a swarm intelligence algorithm.
- It has three phases: -
 - a. Employed bee phase
 - b. Onlooker bee phase
 - c. Scout bee phase
- It has 3 components: -
 - a. Food sources
 - b. Employed foragers
 - c. Unemployed foragers
- Food sources: -
 - a. Its value depends on proximity, richness and ease of extraction.
 - b. It can be represented with a single quantity: profitability.
 - c. It is similar to solutions in the problem. Each solution has an objective value associate with it. Thus, the objective function value can be considered as Profitability.
 - d. Thus, food sources will be solutions in optimizations.
- Employed foragers: -
 - a. Here currently exploiting a food source.
 - b. It contains information on distance, profitability and direction from the nest.
 - c. It shares information with a certain probability to other bees.
- Unemployed foragers: -
 - a. Onlookers: - Here waggle dances to become a recruit and starts searching for a food source.
 - b. Scout: - Here we start searching around the nest spontaneously.
- Employed bee phase: -
 - a. Employed bees try to identify better food source than the one associated with it.
 - b. Generate a new solution using a partner solution.
 - c. Greedy selection is applied, a new solution is accepted if it is better than the current solution.
 - d. In terms of optimization, we generate a new solution using a partner solution and then perform greedy solution.
 - e. Every bee associated with a food source is generating a new solution.
- Onlooker bee phase: -
 - a. Select a food source with a probability related to nectar amount.
 - b. Generate a new solution using a partner solution.
 - c. Greedy selection is applied, if the new solution it is better than the current solution.
- Scout bee phase: -
 - a. Exhausted food source is abandoned.
 - b. In terms of optimization, we discard a particular and generate a new solution.

- Fitness is related to objective function using below relation: -

$$\text{fit} = \frac{1}{1+f}, \text{ if } f \geq 0$$

$$\text{fit} = 1 + |f|, \text{ if } f < 0$$

Thus, if objective value ≥ 0 , then fitness function value = $1/(1+f)$

If objective value < 0 , then fitness function value = $1+|f|$

Thus, as objective function value increases, fitness function value decreases.

As the result, when we perform Greedy selection we need to select the solution with higher fitness.

Employed bee phase: Generation of new solution: -

1. Number of employed bees is equal to number of food sources.
2. All solutions get an opportunity to generate a new solution in the employed bee phase.
3. A partner is randomly selected to generate a new solution.
4. Partner and the current solution should not be the same.

Employed bee phase: Selection of new solution: -

1. Evaluate the objective function and fitness of newly generated solution.
2. Perform greedy selection to update the current solution.
3. 'trial' counter is used to track the number of failures encountered by each solution.
4. The counter value for current solution is increased by one if the new solution is inferior.
5. Reset the counter value if a better solution is generated.
6. Thus, the trial vector keeps track of total number of failures irrespective of employed bee phase or onlooker bee phase.

Onlooker bee phase: -

1. Here there is a condition to be fulfilled to allow a bee to exploit a food source.
2. For each food source, we compute its probability: -

$$\text{Prob}(i) = 0.9 * (\text{fit}(i) / \max(\text{fit})) + 0.1$$

where $\text{Prob}(i)$: Probability of i th solution

$\text{fit}(i)$: Fitness of i th solution

3. Probability values of all solutions are determined before onlooker phase begins.
4. A solution with higher fitness value will have higher probability.
5. Fitter solutions may undergo onlooker bee phase more than once.

'limit' is an integer parameter defined by us which helps to decide if a solution can enter Scout phase.

If the trial value of a solution is greater than limit, the solution can potentially enter the Scout phase.

The trial counter of abandoned solution is reset

Scout bee phase: -

1. Solutions with trial greater than limit are the candidates to be discarded.
2. A solution with its trial greater than limit is replaced with new random solution.

Using ABC algorithm for feature selection: -

1. **Initialization:** -
 - a. We create an initial population of candidate feature subsets.
2. **Fitness evaluation:** -
 - a. We define the objective function to evaluate the results.
 - b. Based on the results of objective function the respective fitness value is computed.
3. **Employed bee phase:** -
 - a. Each employed bee searches for a new feature subset in the neighbourhood of its current position.
 - b. Thus, the search space around the current feature subset is explored.
4. **Onlooker bee phase:** -
 - a. Each onlooker bee selects feature subsets based on their fitness and search for new feature subsets in their neighbourhood.
 - b. They observe the solutions found by employed bees and chose the one with higher fitness to exploit.
5. **Scout bee phase:** -
 - a. If a feature subset cannot be improved further, Scout bees search for new feature subsets randomly.
6. **Termination:** -
 - a. Step 2 to 5 are repeated until the T generations.

Chapter 6

Research about different classification algorithms

Some of the classification algorithms are: -

1. Logistic Regression
2. Support Vector Machines
3. Decision Trees
4. Random Forest
5. Naïve Bayes
6. K-Nearest Neighbours

Logistic Regression: -

1. It is used to predict probabilities for a given datapoint.
2. Since it predicts probabilities, the range of outcomes is between 0 and 1.
3. The same can be used to perform binary classification between two classes.
4. However, it is sensitive to outliers and assumes linear relationship between input variables.

Support Vector Machines: -

1. It helps to build the hyperplane that enables to differentiate between two classes in the dataset.
2. It can handle complex, non-linear classifications.
3. It is inefficient when we have large number of features.
4. It can be computationally expensive.

Decision Trees: -

1. It has a flowchart-like structure with if else conditions.
2. But it tends to overfit and prone to errors if there is small change in dataset.
3. It can be used for both binary classification and multi-class classification.

Random Forest: -

1. It uses many decision trees to make predictions.
2. The results of different trees are combined to get the final outcome of the classifier.
3. Since many trees are used, there is lesser chance of overfitting and higher probability of better results.
4. It works well on scaled and complex data.
5. Since it uses decision trees, Random Forest can also be used for both binary classification and multi-class classification.

Naïve Bayes: -

1. It assumes each feature is independent and computes probability for each class based on the independent features.
2. It can perform well on large datasets.
3. It handles irrelevant features.
4. It is mainly used for text dataset.

K-Nearest Neighbours: -

1. It is also represented as k-NN.
2. It classifies each input into one of the two classes based on the class having 'k' nearest points in the training dataset.
3. Thus, the datapoints that are similar are neighbours of each other.
4. It gets impacted by irrelevant features and the scale of the data.
5. It can be used for both binary classification and multi-class classification.

Chapter 7

Evaluation metrics for classification models

Why do we need evaluation metrics for classification models?

- In machine learning on broad aspect, we have a problem statement to address, then we fetch data related to it, perform analysis and feature engineering. Then use the data to train the model which we finally use to do the prediction.
- Thus, the output of trained machine learning model is consumed by end users for making their decisions.
- In our cybersecurity use case, the impact of the models becomes extremely critical because of the nature of outcome helps to make important decisions about benign and malicious events or type of malicious events.
- In order to use machine learning models in real world scenario, we need to address the fundamental questions such as:-
 1. Why should the end user trust the trained model?
 2. How does our model perform relative to the other models trained by others?
- To address the above fundamental questions, we need to define the governance and framework of evaluation of models which help us understand the given model's performance and also compare them on reliable and useful metrics with other models, which finally allows the end users to make decisions on determining the quality of output produced by the given model and describe the same in detail.
- In terms of building structure for evaluation of classification models, we need to perform seven major tasks:-
 1. List the metrics that can be used for the use case.
 2. Define each metric in detail and explain its benefits and limitations (if any).
 3. Document the evaluation results of all previous models observed from literature survey.
 4. Compute the performance of our model based on each metric defined in task 2.
 5. Quantitatively document the comparison of performance of our model with previously trained models observed in literature survey.
 6. Describe the performance of our model with respect to previously trained model using the data documented in task 6. We need to compute the gap between performance of our model with respect to other models for all available metrics.

7. Derive the inferences based on task 5 and task 6, explain reason for the same. If our model performs better than previously trained models, we need to explain the reasons for achieving better results. Similarly, if our model performs worse than previously trained models, we need to identify the gaps that we need to work on to reach that performance.
- Robust documentation of the above tasks will enable us define the performance of our models which will provide clarity about its application and also convey the same to end users.
 - Additionally, in real world scenarios, the evaluate and decisions to adopt machine learning solutions are taken by different stakeholders. Thus, the specific details in evaluation metrics along with relevant context and research will build the ability of our project to articulate well for different audiences.

Task 1: List of metrics for evaluation of classification models (both binary and multi-class)

1. Confusion Matrix
2. Accuracy
3. Precision
4. Recall
5. F1-Score
6. ROC curve
7. AUC score
8. Balanced accuracy
9. Matthews Correlation Coefficient (MCC)
10. Negative predictive value
11. False discovery rate
12. Cohen kappa metric
13. Precision – Recall curve

Task 2: Definition and details about each metric: -

1. Confusion matrix: -
 - It is used to consolidate data to measure and evaluate performance of classification model.
 - In binary classification, we have 2X2 matrix.
 - In multi-class classification, we have matrix of size same as number of classes in the target feature.
 - Representation of Confusion Matrix for Binary classifier: -

		Actual values	
		Positive	Negative
Predicted values	Positive	True Positive	False Positive
	Negative	False Negative	True Negative

- Representation of Confusion Matrix for Multi-class classifier: -
Assuming 3 classes: Class 1, Class 2, Class 3.

		Actual values		
		Class 1	Class 2	Class 3
Predicted values	Class 1	Cell 1	Cell 2	Cell 3
	Class 2	Cell 4	Cell 5	Cell 6
	Class 3	Cell 7	Cell 8	Cell 9

- True Positive: {Cell 1}, {Cell 5}, {Cell 9}
- False Positive: {Cell 2 + Cell 3}, {Cell 4 + Cell 6}, {Cell 7 + Cell 8}
- True Negative: {Cell 5 + Cell 6 + Cell 8 + Cell 9}, {Cell 1 + Cell 3 + Cell 7 + Cell 9}, {Cell 1 + Cell 2 + Cell 4 + Cell 5}
- False Negative: {Cell 4 + Cell 7}, {Cell 2 + Cell 8}, {Cell 3 + Cell 6}
- True Positive: The number of records model correctly predicts the positive outcome.
- True Negative: The number of records model correctly predicts the negative outcome.
- False Positive: The number of records model incorrectly predicts the positive outcome.
- False Negative: The number of records model incorrectly predicts the negative outcome.
- Examples of above four components in the domain of cyber security: -
 - True Positive: The model correctly predicts a malicious event as an attack.
 - True Negative: The model correctly predicts a normal event as normal.
 - False Positive: The model incorrectly predicts a normal event as an attack.
 - False Negative: The model incorrectly predicts an attack event as normal.
- Confusion matrix is useful in Binary classification due to compact nature of the structure and complex in multi-class classification due to a greater number of classes to be incorporated in the matrix its dimensions will have higher order and interpreting the results will become difficult.
- Confusion matrix can also give incorrect or misleading representation of a model's performance in the datasets having an imbalanced nature of target classes. This is because if the target class is heavily skewed in one direction, and thus the model may showcase high accuracy by predicting everything in the favour of dominant class while failing to detect the rare class.
- For example: In a network dataset, we have 1000 events, 995 are benign and 5 are malicious. Thus, the dataset is highly imbalanced and skewed against malicious events. Now if the classification model predicts everything as benign then the confusion matrix may portray the model has high accuracy but in reality, it failed to correctly detect malicious events which was more critical for

evaluating performance of the classification model.

2. Accuracy: -

- It gives overall correctness of the model.
- $\text{Accuracy} = (\text{True Positive} + \text{True Negative}) / (\text{True Positive} + \text{True Negative} + \text{False Positive} + \text{False Negative})$
- It helps us understand how often the model predicts correctly.
- It is useful in scenarios when the dataset is balanced that is the target feature has balanced representation of all classes.
- It fails to justify false negative in imbalanced dataset.

3. Precision: -

- It gives accuracy of positive predictions.
- $\text{Precision} = \text{True Positive} / (\text{True Positive} + \text{False Positive})$
- It emphasizes on positive predictions made by the model.
- It is better than accuracy while working on imbalanced datasets since it demands minimization of False Positives to have higher score.

4. Recall: -

- It is also called Sensitivity.
- It gives model's ability to find all positive cases.
- $\text{Recall} = \text{True Positive} / (\text{True Positive} + \text{False Negative})$
- In our cybersecurity use-case, if we have 10 malicious events, how many events were successfully classified as malicious by the model will give the value of recall.
- Thus, if a model has high recall, it means it mostly identifies malicious events correctly.
 - classified as benign. Then $\text{recall}=2/10 = 0.2$ (Scenario 4.3)
- It works well on imbalanced datasets.

5. F1-Score: -

- It takes both precision and recall as inputs to give the output.
- $\text{F1-Score} = 2 * (\text{Precision} * \text{Recall}) / (\text{Precision} + \text{Recall})$
- In binary classification, if F1-Score is close to 1, then the model has high accuracy and recall, which indicates model has good performance.
- It is useful when we work on imbalanced dataset.
- It assumes that both precision and recall have equal importance, however, it does not align with our cybersecurity use case. This is because, misclassification of malicious event as benign is a bigger problem than misclassification of benign event as malicious.

Examples:

Let us consider there are 20 events, 16 are benign and 4 are malicious. Thus, the unknown dataset for the classifier is imbalanced.

Case 1: - The classifier predicts all 20 events as Benign.

		Actual values	
		Positive	Negative
Predicted values	Positive	True Positive = 0	False Positive = 0
	Negative	False Negative = 4	True Negative = 16

Accuracy = 0.8

Precision = Not defined ~ 0

Recall = 0

F1-Score = 0

Case 2: - The classifier predicts all 20 events as Malicious.

		Actual values	
		Positive	Negative
Predicted values	Positive	True Positive = 4	False Positive = 16
	Negative	False Negative = 0	True Negative = 0

Accuracy = 0.2

Precision = 0.2

Recall = 1

F1-Score = 0.33

Case 3: - The classifier predicts all 4 Malicious events as Malicious. And it incorrectly predicts 3 Benign events as Malicious.

		Actual values	
		Positive	Negative
Predicted values	Positive	True Positive = 4	False Positive = 3
	Negative	False Negative = 0	True Negative = 13

Accuracy = 0.85

Precision = 0.57

Recall = 1

F1-Score = 0.72

Case 4: - The classifier incorrectly predicts 2 malicious events as Benign.

		Actual values	
		Positive	Negative
Predicted values	Positive	True Positive = 2	False Positive = 0
	Negative	False Negative = 2	True Negative = 16

Accuracy = 0.9

Precision = 1

Recall = 0.5

F1-Score = 0.67

6. ROC curve: -

- ROC: Reverse Operating Characteristics
- Here, we plot true positive rate (on y axis) and false positive rate (on x axis).
- The area under the curve is used to measure the model's performance.
- True Positive Rate = True Positive / (True Positive + False Negative)
- False Positive Rate = False Positive / (True Negative + False Positive)

7. AUC score: -

- AUC: Area Under the Curve
- It is used to for binary classifier and used to differentiate among the classes.
- It is computed using ROC curve.

8. Balanced accuracy: -

- Balanced accuracy = $(\text{True Positive Rate} + \text{True Negative Rate})/2$
- True Positive Rate = $\text{True Positive} / (\text{True Positive} + \text{False Negative})$
- True Negative Rate = $\text{True Negative} / (\text{True Negative} + \text{False Positive})$
- It is useful when classes of the dataset are imbalanced.
- Example: -
 - Case 1:-
 - True Positive Rate = 0
 - True Negative Rate = 1
 - Balanced accuracy = 0.5
 - Case 2:-
 - True Positive Rate = 1
 - True Negative Rate = 0
 - Balanced accuracy = 0.5
 - Case 3:-
 - True Positive Rate = 1
 - True Negative Rate = 0.8125
 - Balanced accuracy = 0.90625
 - Case 4:-
 - True Positive Rate = 0.5
 - True Negative Rate = 1
 - Balanced accuracy = 0.75
- If the score is closer to 1, then the model has higher performance.
- It ranges between 0 to 1.

9. Matthews Correlation Coefficient (MCC): -

- It ranges between -1 to +1.
- $\text{MCC} = (\text{True Negative} * \text{True Positive}) - (\text{False Negative} * \text{False Positive}) / \sqrt{((\text{True Positive} + \text{False Positive}) * (\text{True Positive} + \text{False Negative}) * (\text{True Negative} + \text{False Positive}) * (\text{True Negative} + \text{False Negative}))}$
- If MCC= 0, the classifier performs random classification.
- It is used for binary class and multi-class classification.

10. Negative predictive value: -

- It tells how likely the event is not malicious if it is classified as benign.
- Negative predictive value = $\text{True Negative} / (\text{True Negative} + \text{False Negative})$
- Example: -
 - Case 1: - $\text{NPV} = 16 / (16 + 4) = 0.8$
 - Case 2: - $\text{NPV} = 0 / (0 + 0) = \text{Not defined} \sim 0$
 - Case 3: - $\text{NPV} = 13 / (13 + 0) = 1$
 - Case 4: - $\text{NPV} = 16 / (16 + 2) = 0.89$

11. False discovery rate: -

- False Discovery Rate = $\text{False Positive} / (\text{True Positive} + \text{False Positive})$
- Here we determine out of all the events that the model classified as malicious; how many were incorrect.
- Thus, this helps us determine the noise generated by the model.

- Example: -
 - Case 1: - $FDR = 0 / (0 + 0) = \text{Not defined} \sim 0$
 - Case 2: - $FDR = 16 / (16 + 4) = 0.8$
 - Case 3: - $FDR = 3 / (3 + 4) = 0.428$
 - Case 4: - $FDR = 0 / (0 + 2) = 0$
- We can also use it to for feature selection, to determine features that are associated with malicious events.

12. Cohen kappa metric: -

- It takes into account that model may correctly classify the some of the events purely by chance.
- It is also called Kappa Score.
- $k=(p_0-p_e)/(1-p_e)$
 p_0 : Relative measured among models
 p_e : Hypothetical probability of chance agreement
- $k=1$: There is complete agreement between the models.
- $k<0$: There is no agreement between the models.
- Example: -

Case	p0	pe	k
1	0.8	0.8	0
2	0.2	0.2	0
3	0.85	0.59	0.63
4	0.9	0.74	0.615

- Following are the interpretations of Cohen's kappa: -

Table 10: Interpretation of Cohen's kappa values

Cohen's kappa	Interpretation
0	No agreement
0.10 - 0.20	Slight agreement
0.21 - 0.40	Fair agreement
0.41 - 0.60	Moderate agreement
0.61 - 0.80	Substantial agreement
0.81 - 0.99	Near perfect agreement
1	Perfect agreement

- As per our 4 cases: -

Case 1 and case 2 have no agreement. Thus, the two models are far away from expected model.

Case 3 and case 4 have substantial agreement. Thus, the two models are closer to the expected model.

13. Precision – Recall curve: -

- It is useful for imbalanced dataset.

- In our cyber security use case, instead of predicting the binary classifier classes: malicious and benign directly, we predict the probability of an event being malicious.
- Then we plot the graph with Recall on x-axis and Precision on y-axis.
- The area under Precision Recall curve is the indicator of performance. Larger the area, better the model performs.

Directions for future work after mid semester

Following are the activities to be performed after mid semester: -

1. Working on imbalanced nature of the dataset: -
 - We have fetched different ways for handling imbalanced nature of CIC dataset.
 - Based on the information fetched we need to determine the appropriate approach and implement the same.
2. Standardization of features: -
 - After handling of features, we need to determine and implement the suitable approach for standardizing the independent features of the dataset.
3. Selection of classification algorithms: -
 - We have fetched different ways for classification algorithms for binary and multiclass classification.
 - Thus, based on the information, we need to determine the suitable algorithms that can be used during feature selection and training the models, given the size of dataset and approach adopted for handling its imbalanced nature.
4. Implementation of feature selection using heuristic algorithms: -
 - We have studied different heuristic algorithms that can be used for feature selection.
 - Now we need to implement those algorithms to get best subset of features for training the models.
5. Training the classifier models: -
 - We need to train the models based on features obtained from feature selection process.
6. Evaluation of models: -
 - After training the models, we need to fetch a new sample of dataset from the main dataset with different seed number, apply data pre-processing and transformation steps that were performed earlier.
 - And then use the dataset to evaluate the models based on the evaluation metrics.
 - Comparison of evaluation results with results of models in literature survey.

7. Documentation of results: -

- Interpretation of results for each evaluation metric.
- Understanding whether our models gave better results or worse results than the ones in literature survey, identify the factors for improvement.

Bibliography/ References

1. A Journal Paper: Applied Artificial Intelligence
Duygu Yilmaz and Umut Akcan, 'An Adapted Ant Colony Optimization for Feature Selection', Taylor & Francis Group, Vol. 38, No 1, Mar. 2024.
2. A Journal Paper: Computers, Materials & Continua
Ting Cai, Chun Ye, Zhiwei Ye, Ziyuan Chen, Mengqing Mei, Haichao Zhang, Wanfang Bai and Peng Zhang, 'Multi-Label Feature Selection Based on Improved Ant Colony Optimization Algorithm with Dynamic Redundancy and Label Dependence', Tech Science Press, Vol. 84, No 1, pp. 1157-1175, Oct. 2024
3. A Journal Paper: International Journal of Computer Applications
Harshit Saxena and Vineet Richaariya, 'Intrusion Detection in KDD99 Dataset using SVM-PSO and Feature Reduction with Information Gain', Foundation of Computer Science (FCS), NY, USA, Vol. 98, No. 6, pp. 25-29, July 2014
4. A Journal Paper: Soft Computing
Ahmed Abdullah Alqarni, 'Towards support-vector machine-based any colony optimization algorithms for intrusion detection', Soft Computing, Vol. 27, pp. 6297-6305, Feb. 2023
5. A Journal Paper: International Journal of Systems Science
Siva S. Sivatha Sindhu, S. Geetha and A. Kannan, 'Evolving optimised decision rules for intrusion detection using particle swarm paradigm', Vol. 43, No. 12, pp. 2334-2350, May 2011
6. A Journal Paper: IEEE Transactions on Evolutionary Computation
Grzegorz Dudek, 'An Artificial Immune System for Classification with Local Feature Selection', IEEE Computational Intelligence Society, Vol 16, No. 6, pp. 847-860, Dec 2012
7. A Journal Paper: Journal of Theoretical and Applied Information Technology
T. Sumathi, S. Karthik and M. Marikkannan, 'Artificial Bee Colony Optimization for Feature selection in Opinion mining', Vol. 66, No. 1, pp. 368-379, Aug. 2014
8. A Journal Paper: EURASIP Journal on Image Video Processing
Mauricio Schiezaro and Helio Pedrni, 'Data feature selection based on Artificial Bee Colony algorithm', Image Video Proc, Vol 2013, No 47, pp. 1-8, Aug. 2013
9. A Journal Paper: AppliedMath
Efe Precious Onakpojeruo, Nuriye Sancar, 'A Two-Stage Feature Selection Approach Based on Artificial Bee Colony and Adaptive LASSO in High-Dimensional Data', AppliedMath, Vol. 2024, No 4, pp. 1522-1538, Dec. 2024
10. A Journal Paper: International Journal of Computer Science Issues (IJCSI)
Shunmugapriya Palanisamy and Kanmani S, 'Artificial Bee Colony Approach for Optimizing Feature Selection', IJCSI, Vol. 9, No. 3, pp. 432-438, May 2012