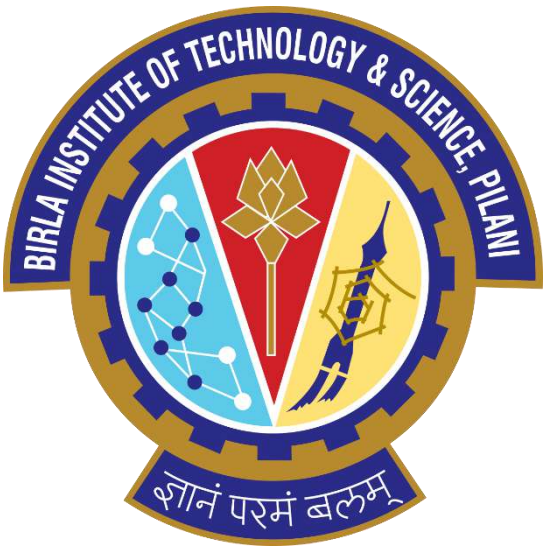


BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI

FIRST SEMESTER 2024-25

MTech in Data Science and Engineering

Dissertation

To Establish Baseline for Threat Detection

Goyal Taruchit Tarun Chitra

2022DC04496

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

(February, 2025)

Acknowledgements

I want to thank BITS-Pilani and all their faculty for the course on Data Science and enabling me to learn fundamentals and implement the same during the project.

I want to thank the evaluator of the project, Prof. S. Geetha, for sharing her guidance and inputs to improve during phase 1 and phase 2 of the evaluation of the project.

I want to express gratitude to my mentor Ms. Prathibha Rao, for her guidance and support during the project.

I also want to express gratitude to my functional manager Mr. Myke Hamada, for his constant support and guidance during the course and the project.

I acknowledge the support and guidance of following people from my workplace who contributed and guided during different stages of the project: -

1. Mr. Ajish George
2. Mr. Rohan Dhume
3. Mr. Rafael Derett
4. Ms. Anna Melikyan
5. Ms. Sudha Narayanan
6. Ms. Sowmya Chamathakundil
7. Mr. Sriram Balakrishnan
8. Mr. Alok Raj

I want to thank following people from the industry whose guidance and inputs helped during the project: -

1. Dr. Munmum Das
2. Ms. Monisha M.
3. Mr. Manoj Balaji
4. Mr. Alex Terixeira

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE, PILANI

CERTIFICATE

This is to certify that the Dissertation entitled To Establish Baseline for Threat Detection

and submitted by Mr. Goyal Taruchit Tarun Chitra ID No.2022DC04496 in partial fulfillment of

the requirements of DSECLZG628T
Dissertation, embodies the work

done by him/her under my supervision.

Signature of the Supervisor

Place: BANGALORE

Date: 01/03/2025

Name PRATHIBHA PANDURANGA RAO
Designation Vice President

BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI
Work Integrated Learning Programmes Division
I SEMESTER 24-25

DSECLZG628T **DISSERTATION**

(Final Evaluation Sheet)

NAME OF THE STUDENT: TARUCHIT GOYAL

ID NO. : 2022DC04496

Email Address : Pkandurangarao@statestreet.com.

NAME OF THE SUPERVISOR: PRATHISTHA PANDURANGA RAO

PROJECT TITLE : To Establish Baseline for Threat detection

(Please put a tick (✓) mark in the appropriate box)

S.No.	Criteria	Excellent	Good	Fair	Poor
1	Work Progress and Achievements	✓			
2	Technical/Professional Competence	✓			
3	Documentation and expression	✓			
4	Initiative and originality	✓			
5	Punctuality	✓			
6	Reliability	✓			
	Recommended Final Grade	✓			

EVALUATION DETAILS

EC No.	Component		Weightage	Marks Awarded
1	Dissertation Outline		10%	9.0/0
2	Mid-Sem Progress			
		Seminar	10%	9.5/0
		Viva	5%	
		Work	15%	14.0/0
	Progress			
3	Final Seminar/Viva		20%	18.0/0
4	Final Report		40%	38.0/0
	Total out of		100%	92.5/0

Note : Mark awarded should be in terms of % of weightage (consider 10% weightage as 10 marks)

	Organizational Mentor	
Name	PRATHIBHA PANDURANGA RAO	
Qualification	M.Sc in I.T	
Designation & Address	VICE PRESIDENT ADARSH PALM RETREAT, DODDAKENNELLI	GB RM2 ECO WORLD RD BILORE - 560103.
Email Address	PPandurangarao@statestreet.com.	
Signature	Prathibha	
Date	01/03/2025	

NB: Kindly ensure that recommended final grade is duly indicated in the above evaluation sheet.

The Final Evaluation Form should be submitted separately in the viva portal.

BIRLA INSTITUTE OF INFORMATION TECHNOLOGY & SCIENCE, PILANI
FIRST SEMESTER 2024-2025

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.

Summary of literature survey

Importance of feature selection: -

Optimal feature selection is key to training a high-performance classifier model. It involves managing interaction among independent features such that a subset of their best combination helps us to differentiate efficiently between the classes of dataset while reducing the model complexity.

Benefits of performing feature selection over using advanced techniques to build classification models: -

It was observed that in many researches, people rely on advanced deep learning, graph neural networks for building high performance classifiers, which requires high volume and high computation cost. It does not help us explicitly determine which features were ultimately used for classification of data, and thus, limits the collaboration with respective domain experts. However, in real-world use cases, models are trained with collaboration of domain experts so that the models do not learn incorrect patterns from the dataset which is important for their reliability and usage.

It also enables us to set a very high benchmark of outcomes achieved using ML algorithms, and enable the advanced techniques to achieve much better results.

Observations about usage of Heuristic approaches used for optimal feature selection: -

Most of the prior researches have following common attributes for achieving optimal feature selection using Heuristic algorithms: -

1. Accuracy is used as the fitness function/objective function

2. Evaluation of results is broadly done on 4 metrics: Accuracy, Precision, Recall, F1-Score.

3. Commonly used heuristic algorithms: Ant Colony Optimization, Particle Swarm Optimization, Artificial Bee Colony optimization.

4. Commonly used machine learning algorithms: SVM, Decision trees, Naïve Bayes.

5. Datasets used were relatively very small and synthetically generated. Most of the datasets have number of instances between 50 to 3000, and number of features between 4 to 25. KDD99 dataset has 42 features and over 490000 instances, but around 70% are duplicates.

6. Number of iterations to fetch optimal results range between 100 to 1000.

Limitations observed in above research papers: -

1. In real-world scenarios, handling class imbalance is essential to build a reliable classifier which has less error percentage. In most of the prior research, “accuracy” was used as the metric to compute the quality of a feature subset which gives overall correctness of the model. It gives reliable results for balanced datasets and fails to handle false negatives which are extremely critical especially in cyber security use cases.

2. The datasets used to train and evaluate are very small in size and synthetically generated, which leads to likelihood of noise being captured by the models to achieve higher results. However, in large size datasets which are generated in real-time, the quality of data is richer in terms of diversity, which enables us to train the models that better handle unseen data and thus are more reliable and useful for critical and practical business problems.

3. As the number of features used to train the model increases, the complexity of model also increases. The prior research papers do not emphasize on usage of heuristic algorithms for reducing number of features in feature selection while achieving outcomes similar to large number of features. The number of iterations to generate optimal subset of features was between 100 to 1000. Thus, with high computational cost and time required, the researchers tried to fetch optimal feature subset.

4. The machine learning algorithms used in prior research are computationally intensive. In heuristic algorithms, we need to recurringly use the selected ML algorithm for computing fitness of a given solution. As the result, the cost to compute high quality results is very high.

5. In all research papers where Artificial Bee Colony optimization algorithm was used, in Onlooker bee phase, the probability of a food source was computed by taking ratio of fitness of the given food source and sum of the fitness of all food sources. However, as per the original literature, the probability of a given food source is computed using Equation (2). The reason for divergence to compute the probability was not covered in any prior research paper. Moreover, while computing the ratio, taking sum of all fitness values in the denominator leads to floating point underflow, because the computed ratio may be too small to be represented accurately.

List of Symbols and Abbreviations

Sr No	Abbreviation	Definition
1.	IOC	Indicators of Compromise
2.	UEBA	User Entity and Behavior Analytics
3.	IDD	Inadvertent Data Disclosure
4.	IP	Internet Protocol
5.	URL	Uniform Resource Locator
6.	SVM	Support Vector Machine
7.	prob(i)	Probability of ith solution
8.	fit(i)	Fitness value of ith solution
9.	max(fit)	Maximum fitness among all solutions
10.	r	Random number
11.	p	Switch probability
12.	λ	Step size in Levy flight
13.	γ	Pulse emission rate
14	L()	Levy distribution
15.	g*	Current best solution
16.	ε	Normal distribution
17.	ROC	Reverse Operating Characteristics
18.	AUC	Area Under the Curve
19.	MCC	Matthews Correlation Coefficient
20.	NPV	Negative Predictive Value
21.	FDR	False Discovery Rate
22.	K-NN / KNN	K-Nearest Neighbors
23.	ABC	Artificial Bee Colony
24.	FPA	Flower Pollination Algorithm
25.	ACO	Ant Colony Optimization
26.	PSO	Particle Swarm Optimization
27.	RIDOR	Ripple Down Rule Learner
28.	ML	Machine Learning

List of Tables

List of Figures

List of Equations

Equation number	Equation	Description of variables in the equation	Page number
Equation (1)	Robust scaling = x – median/IQR	Computation of Robust scaling	56
Equation (2)	Prob(i) =0.9*(fit(i)/max(fit))+0.1	Computing probability of each food source prior Onlooker bee phase starts in ABC algorithm Prob(i) = Probability of ith food source fit(i) = Fitness of the ith food source max(fit) = Maximum fitness among all food sources	133
Equation (3)	fit=1/1+f, if f >= 0 fit=1+|f|, if f< 0	Fitness function for ABC algorithm f = objective value of a given solution fit = fitness value of a given solution	134
Equation (4)	Accuracy = (True Positive + True Negative)/(True Positive + True Negative + False Positive + False Negative)	Computation of Accuracy for a model	142
Equation (5)	Precision = True Positive /(True Positive + False Positive)	Computation of Precision for a model	142
Equation (6)	Recall = True Positive /(True Positive + False Negative)	Computation of Recall for a model	142
Equation (7)	F1-Score=2*(Precision * Recall)/(Precision + Recall)	Computation of F1-Score	143
Equation (8)	True Positive Rate = True Positive /(True Positive + False Negative)	Computation of True Positive Rate	145
Equation (9)	False Positive Rate = False Positive/(True Negative + False Positive)	Computation of False Positive Rate	145
Equation (10)	Balanced accuracy = (True Positive Rate +True Negative Rate)/2	Computation of Balanced accuracy	145

Equation (11)	True Negative Rate = True Negative/(True Negative + False Positive)	Computation of True Negative Rate	145
Equation (12)	MCC=(True Negative * True Positive) – (False Negative * False Positive) / sqrt((True Positive + False Positive)* (True Positive + False Negative) * (True Negative + False Positive) * (True Negative + False Negative))	Computation of Matthews Correlation Coefficient (MCC)	145
Equation (13)	Negative Predictive Value = True Negative/(True Negative + False Negative)	Computation of Negative Predictive Value	146
Equation (14)	False Discovery rate = False Positive/(True Positive + False Positive)	Computation of False Discovery Rate	146
Equation (15)	k = (p0-pe)/(1-pe)	Computation of Cohen Kappa score k = Kappa Score p0 = Relative measured among models pe = Hypothetical probability of chance agreement	146
Equation (16)	Detection rate = Number of normal correctly classified/Total number of normal	Computation of Detection rate	181
Equation (17)	Error rate = Number of normal incorrectly classified/Total number of normal	Computation of Error rate	181
Equation (18)	X is an outlier if: - X< Q1 – 1.5 * IQR or X>Q3 + 1.5 * IQR	Detection of Outliers using IQR IQR is Inter Quartile Range Q1 = 25th percentile Q3 = 75^{th} percentile	52
Equation (19)	new_solution=current_solution+ γ*L(λ)*(g*-current_solution)	Global pollination in Flower Pollination Algorithm	136
Equation (20)	new_solution=current_solution +ε * (xj-xk)	Local pollination in Flower Pollination Algorithm	136

Table of Content

Chapter 1

List of Objectives

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 heuristic optimization algorithms for feature selection
5. Research about different classification algorithms
6. Research about different evaluation metrics used to evaluate results of classification model
7. Feature selection and training of models
8. Evaluation of models

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)

2.1 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.

2.2 Analyzing 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 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.

2.3 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.

2.4 UEBA for anomaly detection: -

1. Here the focus is on homogenous population in the organization that have similar and repetitive patterns of behaviour.
2. For creating baseline of users, we try to find users who are similar and then form a baseline based on their behaviour pattern.
3. 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.

2.5 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

3.1 Summary of open-source datasets: -

We collected 7 open-source datasets from different sources, and they were analysed to compare their characteristics, to gather information and finalize the dataset for the project.

Table 3.1.1 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 3.1.2 Comparison of all datasets

Sr No	**Dataset name**	**Advantages**
1	BETH	1. Large number of records. 2. No duplicates and no null records.
2	BrakTooth	1. No null values. 2. Moderate number of records 3. Allows to build model for multi-class classification.
3	Mil-STD- 1553	1. Large number of records with very high dimensionality. 2. Allows to build model for multi-class classification.
4	ServerLogs	1. Large number of records. 2. No duplicates or null records.
5	UNR-IDD	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.

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

1. UNR-IDD
2. cic

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

3.2 UNR-IDD dataset: -

Table 3.2.1 List of fields in UNR-IDD dataset

Sr no	Field name	Description	Type of field	Comments
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	Scale	Port statistics
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	Scale	Delta port statistics
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.	Scale	Flow entry and Flow table
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 Attack: Network intrusion	Binary classification	Target: Binary

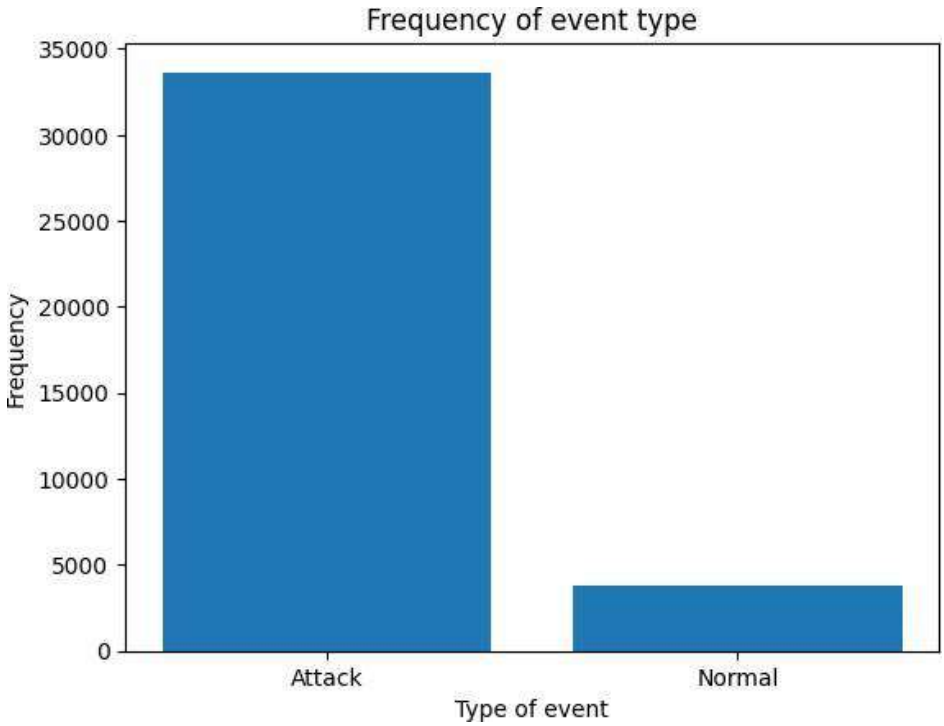


Figure 3.2.1 Bar chart of events in UNR-IDD dataset based on Binay label.

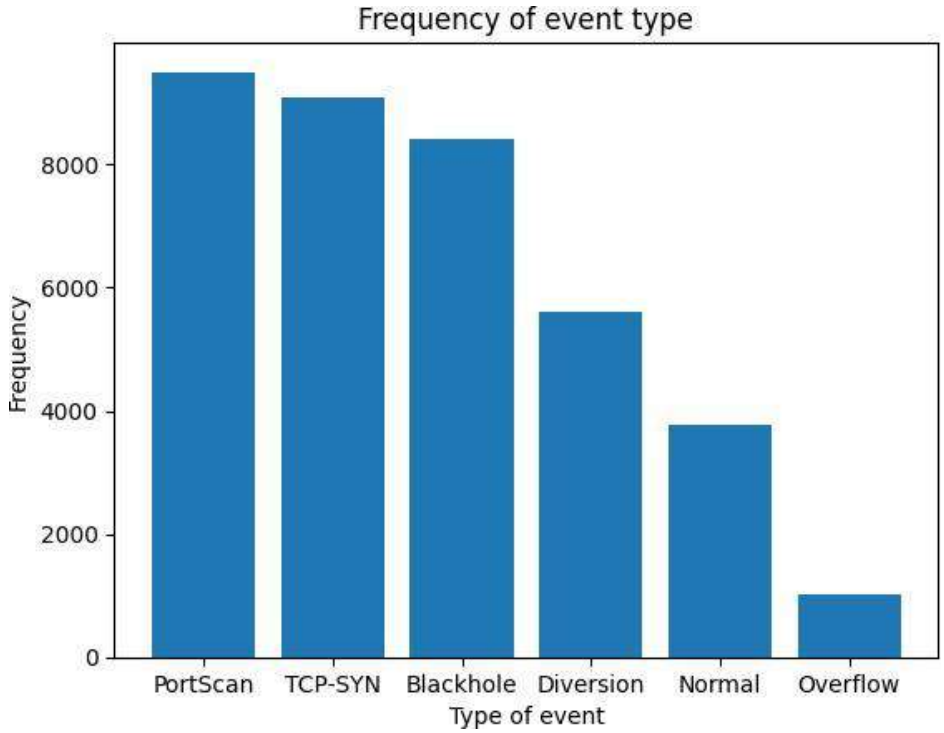


Figure 3.2.2 Bar chart of events in UNR-IDD dataset based on Label.

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.

3.3 CIC dataset: -

Table 3.3.1 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	Targetclass: Multi- class classification
59	ClassLabel	Subtype of intrusion	Targetclass: Multi- class classification

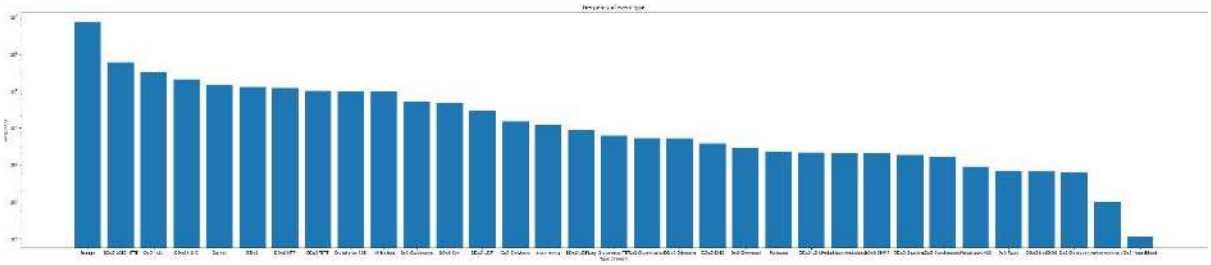


Figure 3.3.1 Bar chart of events in CIC dataset based on Label

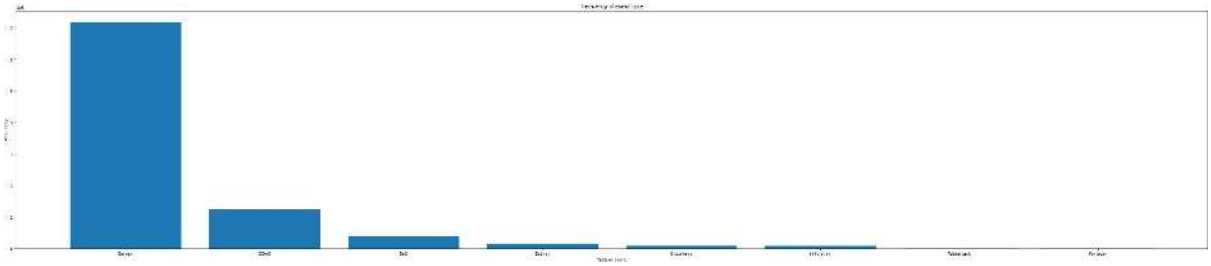


Figure 3.3.2 Bar chart of events in CIC dataset based on ClassLabel

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: -

Imbalanced nature of target features.

3.4 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 analyze.

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

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

4.2 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.
- 0.0016% 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.

4.3 Summarized view of distribution of data plotted on log scale for 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.

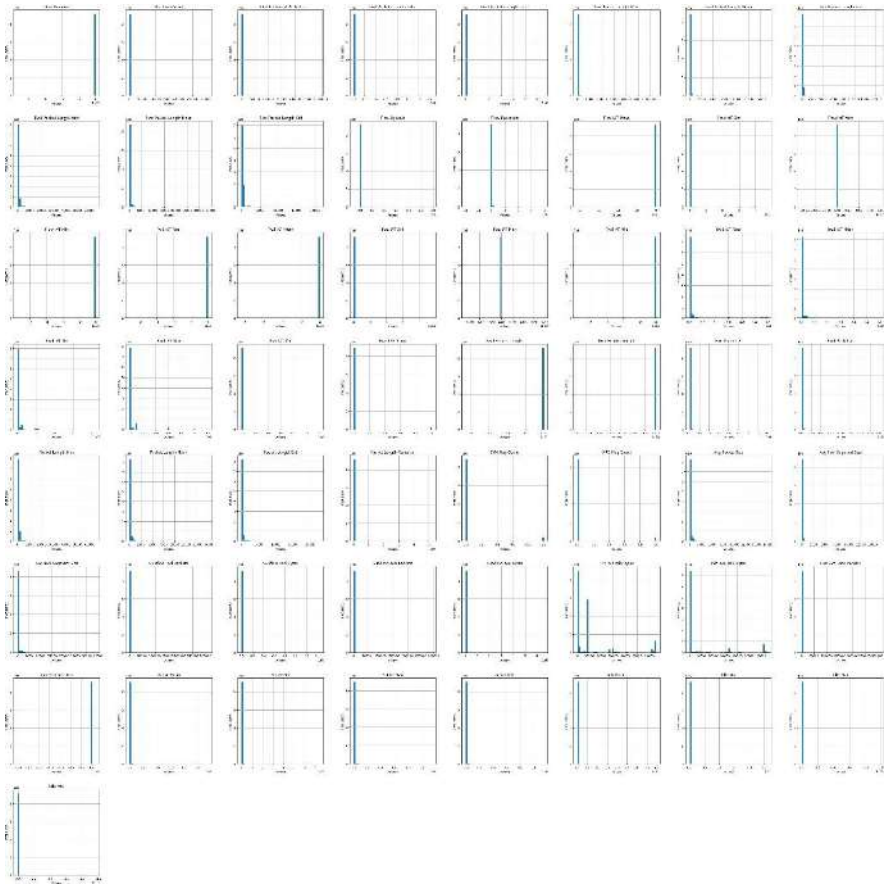


Figure 4.3.1 Histogram of all independent features plotted on normal scale.

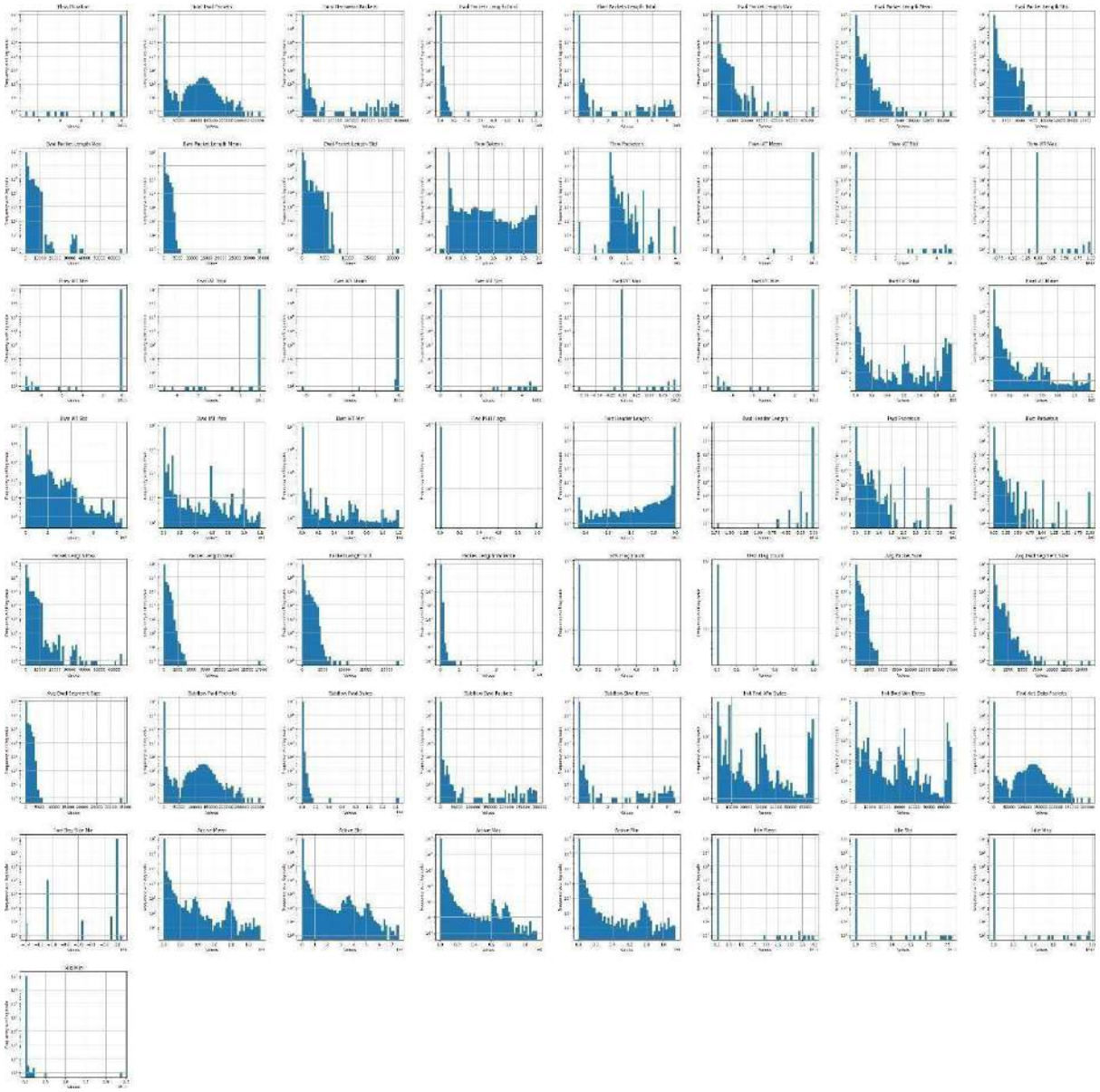


Figure 4.3.2 Histogram of all independent features plotted on log scale.

4.4 Distribution of datapoints plotted on log scale for each independent feature: -

Flow Duration: The duration of the flow

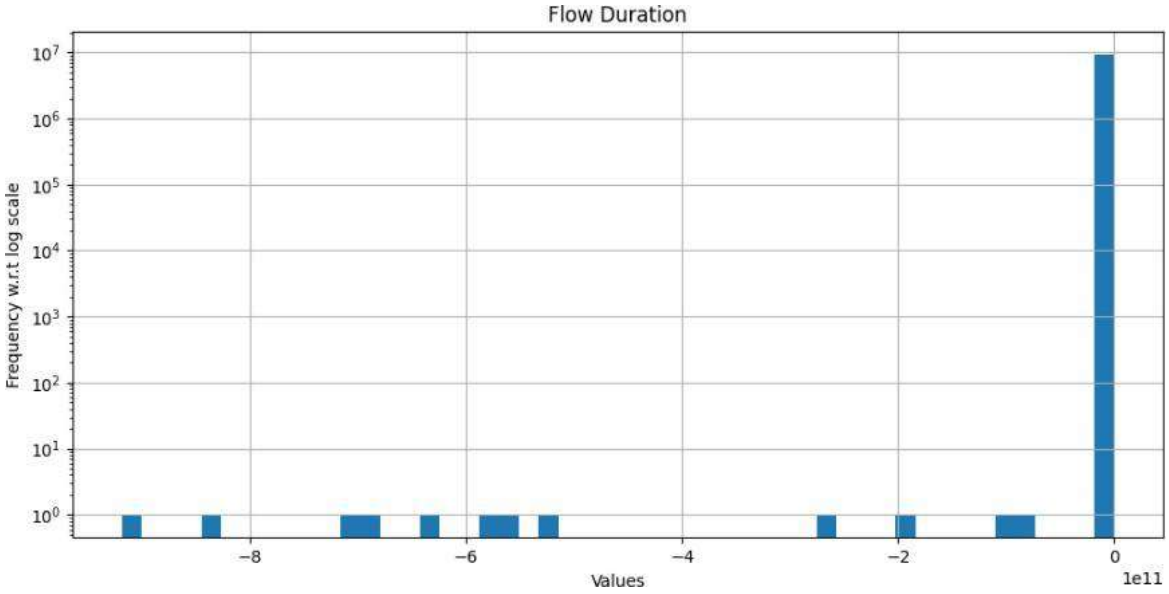


Figure 4.4.1 Histogram of Flow Duration plotted on log scale

- 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

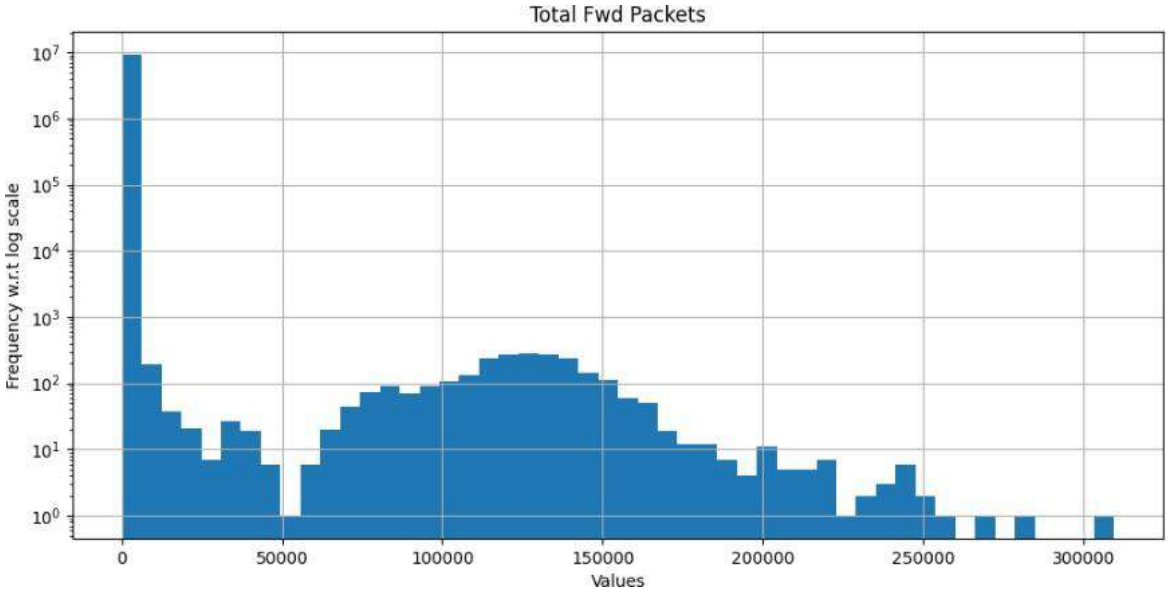


Figure 4.4.2 Histogram of Total Fwd Packets plotted on log scale

- 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

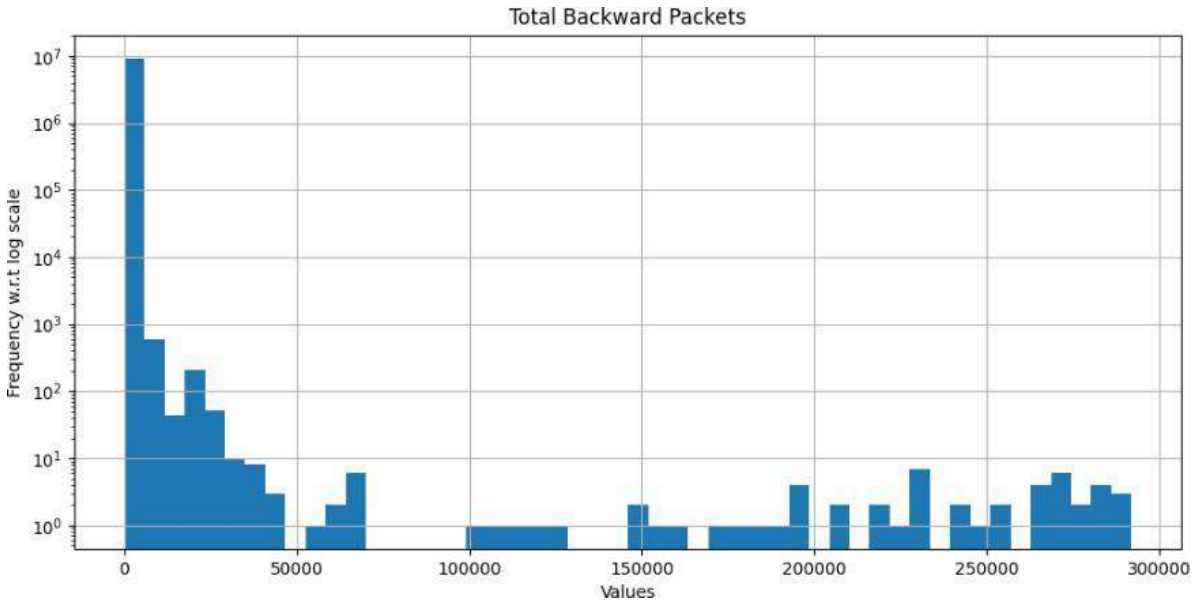


Figure 4.4.3 Histogram of Total Backward Packets plotted on log scale

- 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

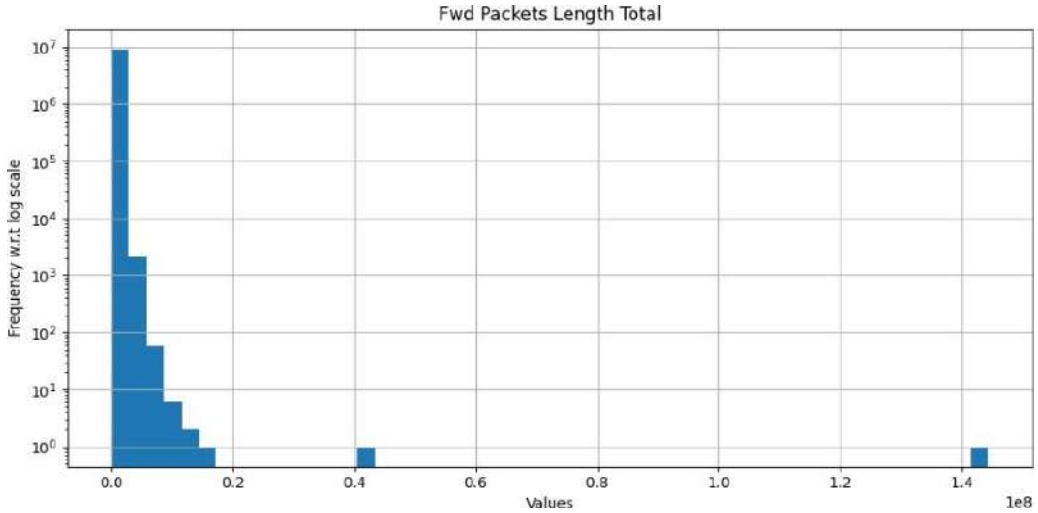


Figure 4.4.4 Histogram of Fwd Packets Length Total plotted on log scale

- 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

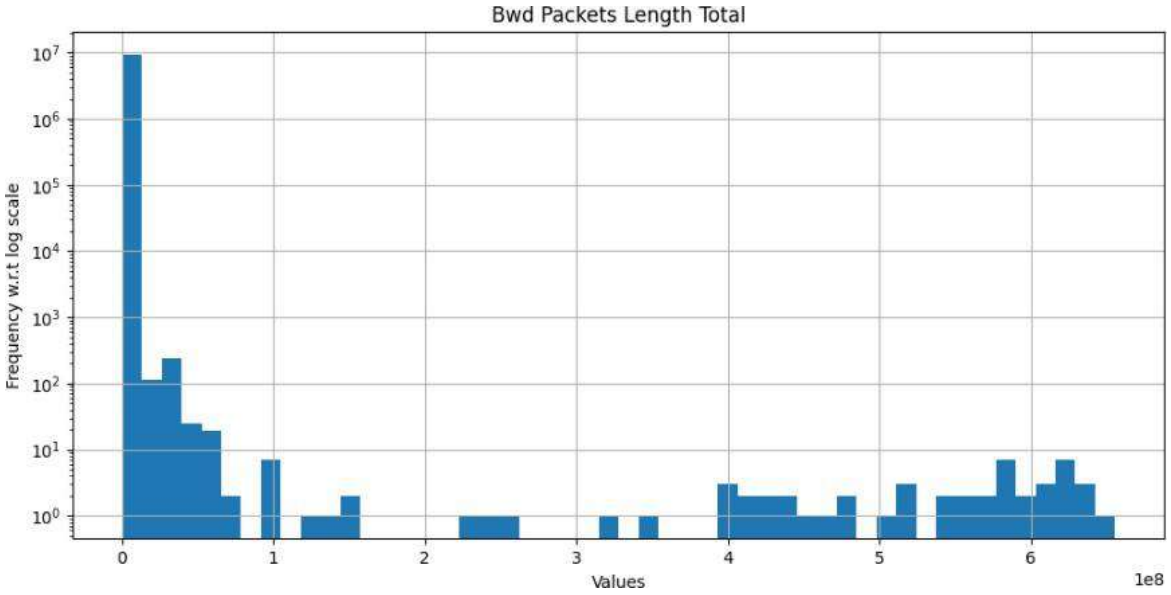


Figure 4.4.5 Histogram of Bwd Packets Length Total plotted on log scale

- 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

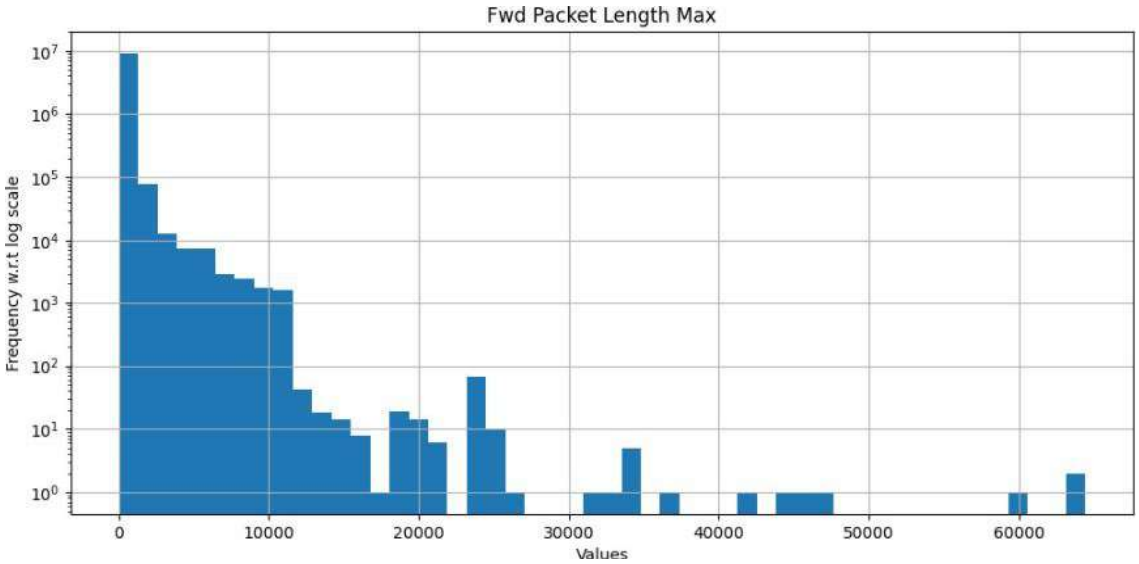


Figure 4.4.6 Histogram of Fwd Packet Length Max plotted on log scale

- 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 ofobservations 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

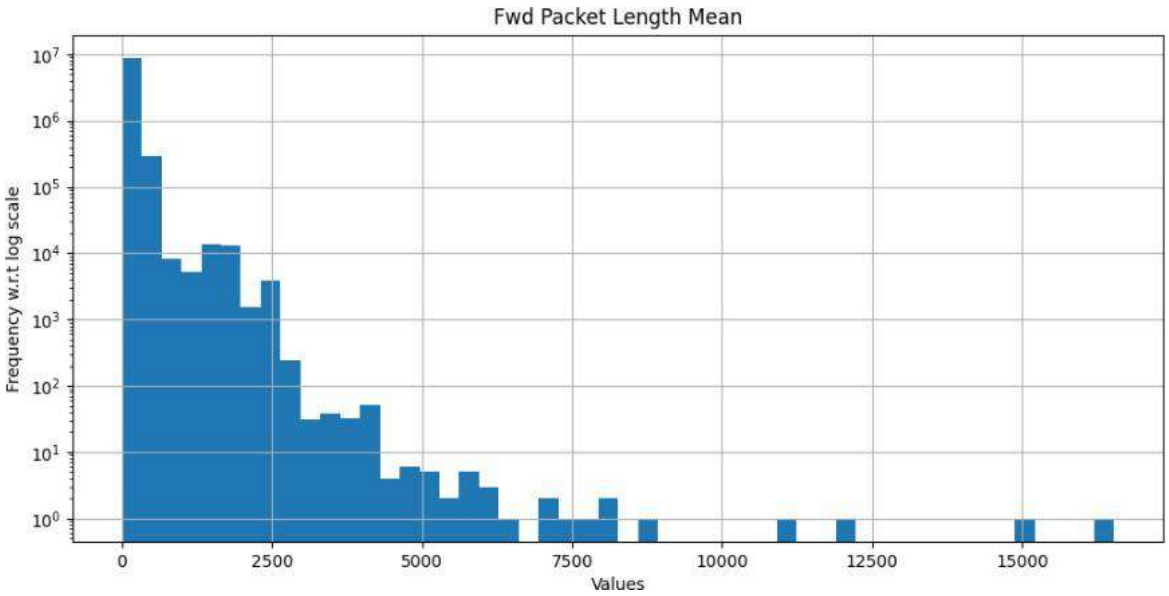


Figure 4.4.7 Histogram of Fwd Packet Length Mean plotted on log scale

- 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>=0 and Fwd Packet Length Mean<=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

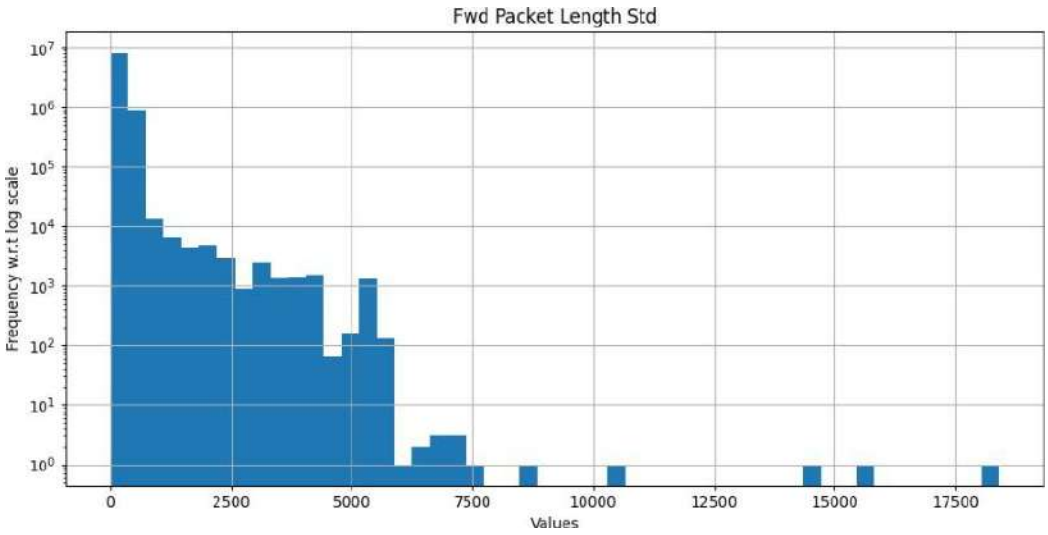


Figure 4.4.8 Histogram of Fwd Packet Length Std plotted on log scale

- 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>=0 and Fwd Packet Length Std<=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

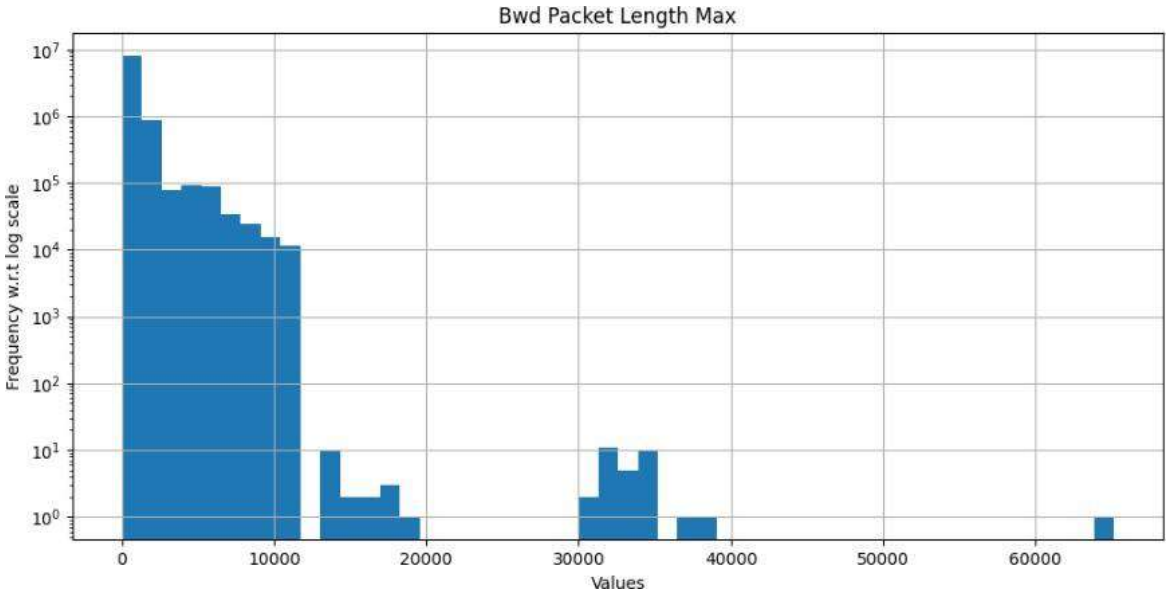


Figure 4.4.9 Histogram of Bwd Packet Length Max plotted on log scale

- The distribution is skewed towards right: Positively skewed.
- Peak was observed on first bin from left. Peak lies around Bwd Packet Length Max>=0 and Bwd Packet Length Max<=1250.

- Most number of observations lie between Bwd Packet Length Max>=0 and Bwd Packet Length Max<=10000.
- There are few observations in the range: - Bwd Packet Length Max>=11250 and Bwd Packet Length Max<=20000, Bwd Packet Length Max>=30000 and Bwd Packet Length Max<=35000.
- There is an observation at Bwd Packet Length Max>60000. This may indicate outliers in the data.

Bwd Packet Length Mean: Mean length of backward packet

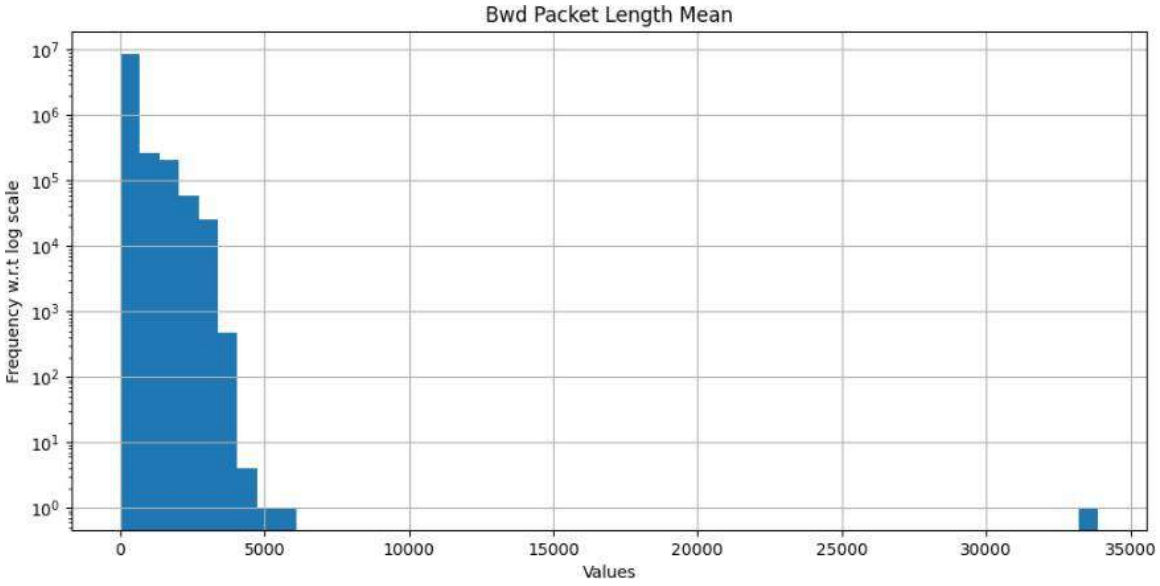


Figure 4.4.10 Histogram of Bwd Packet Length Mean plotted on log scale

- The distribution is skewed towards right: Positively skewed.
- Peak was observed on first bin from left. Peak lies around Bwd Packet Length Mean>=0 and Bwd Packet Length Mean<=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

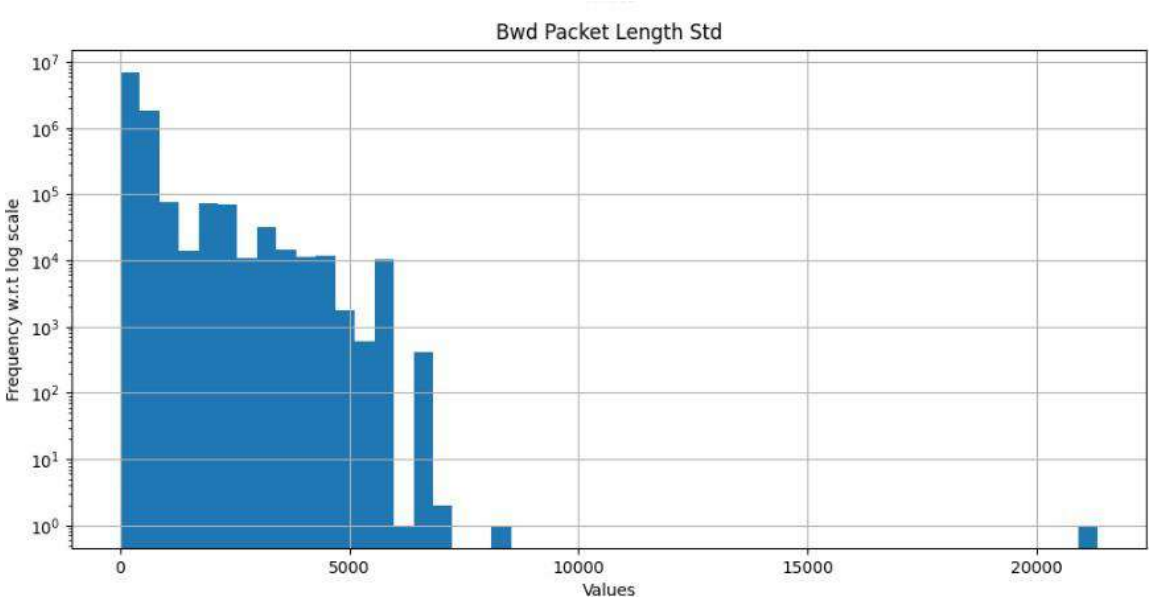


Figure 4.4.11 Histogram of Bwd Packet Length Std plotted on log scale

- The distribution is skewed towards right: Positively skewed.
- Peak was observed on first bin from left. Peak lies around Bwd Packet Length Std>=0 and Bwd Packet Length Std<=416.67.
- After the peak, there is significant decline in results.
- There is plateau region observed around Bwd Packet Length Std>=2083 and Bwd Packet Length Std<=2500.
- There is another plateau region observed (smaller than the above) around Bwd Packet Length Std>=3750 and Bwd Packet Length Std<=4166.
- There is an observation at Bwd Packet Length Std>20000. This may indicate outliers in the data.

Flow Bytes/s: Flow bytes per second

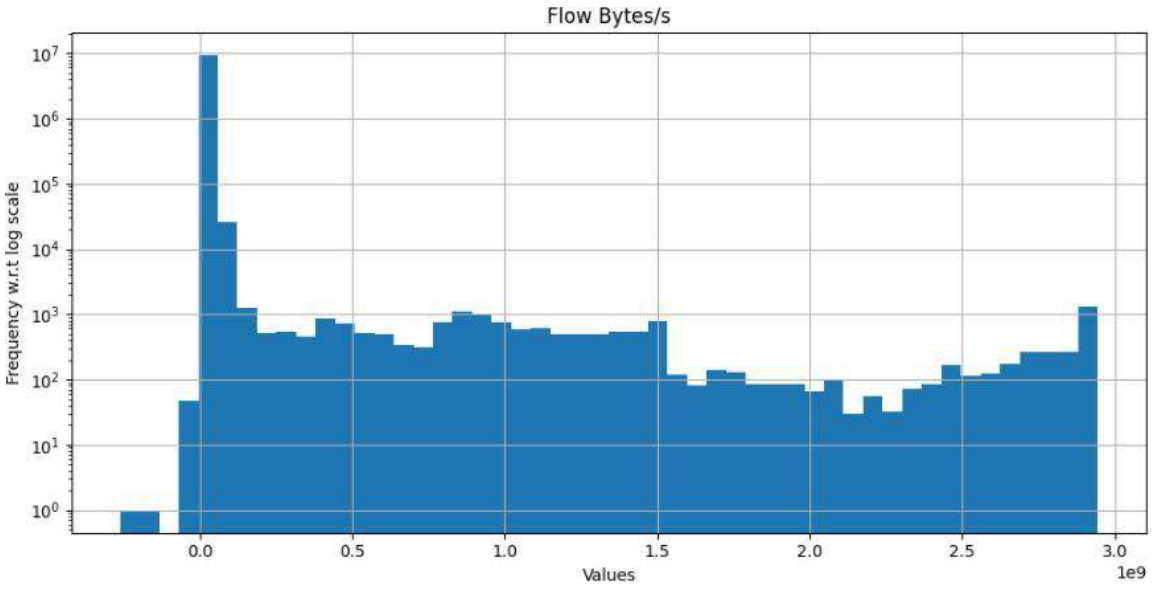


Figure 4.4.12 Histogram of Flow Bytes/s plotted on log scale

- 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

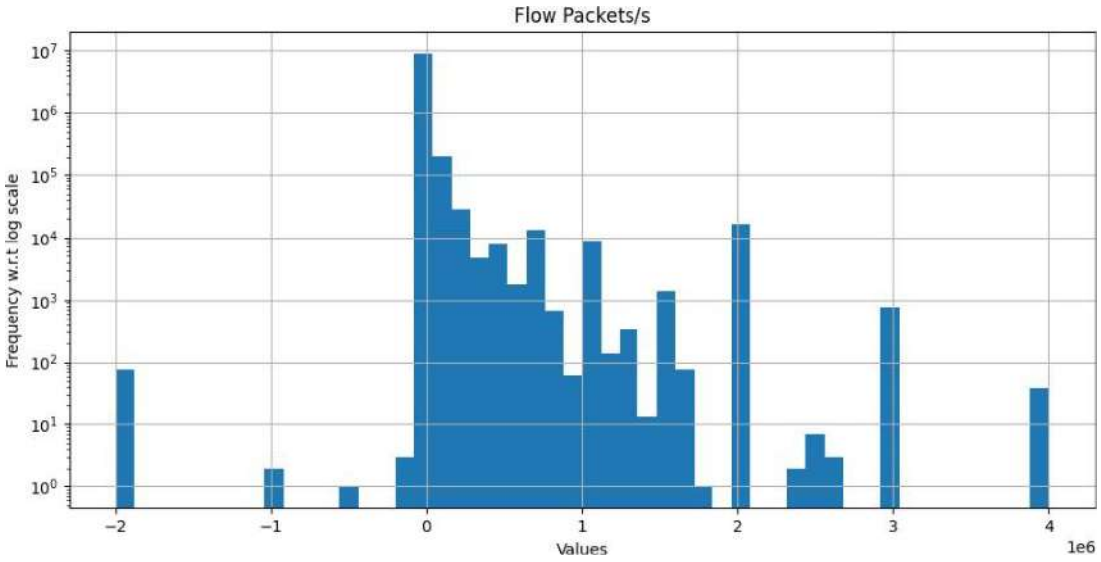


Figure 4.4.13 Histogram of Flow Packets/s plotted on log scale

- 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

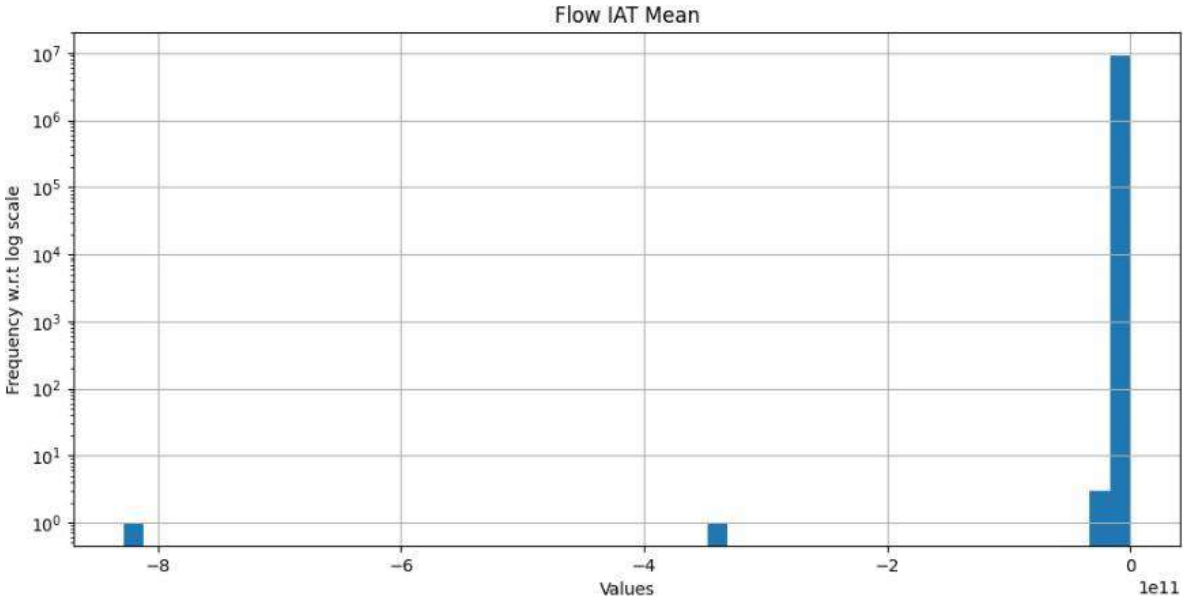


Figure 4.4.14 Histogram of Flow IAT Mean plotted on log scale

- 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

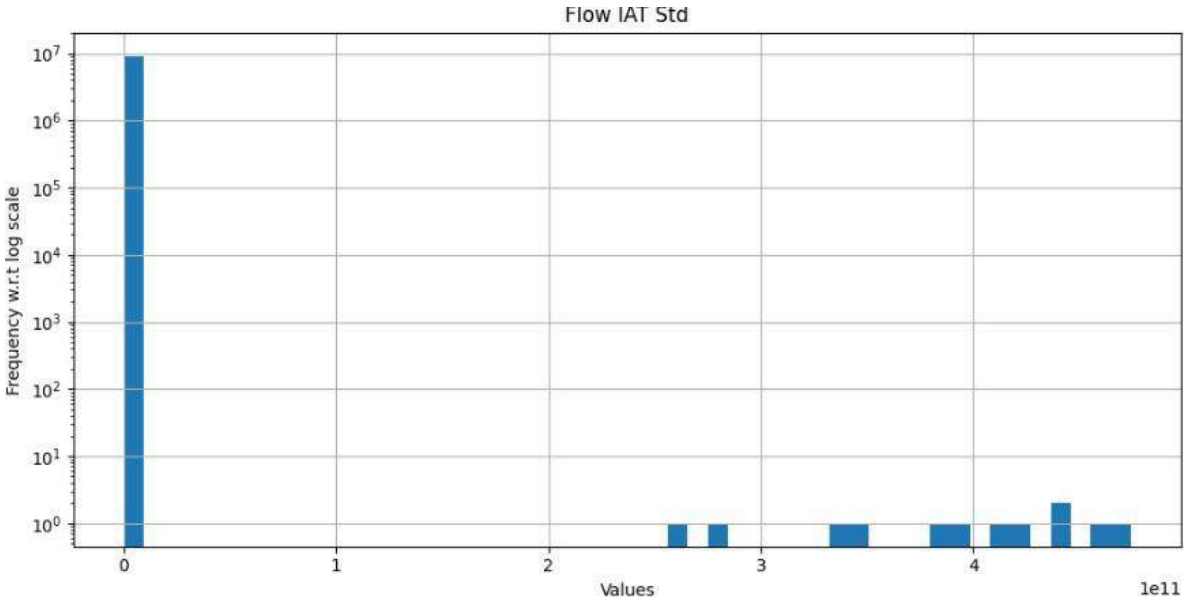


Figure 4.4.15 Histogram of Flow IAT Std plotted on log scale

- 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

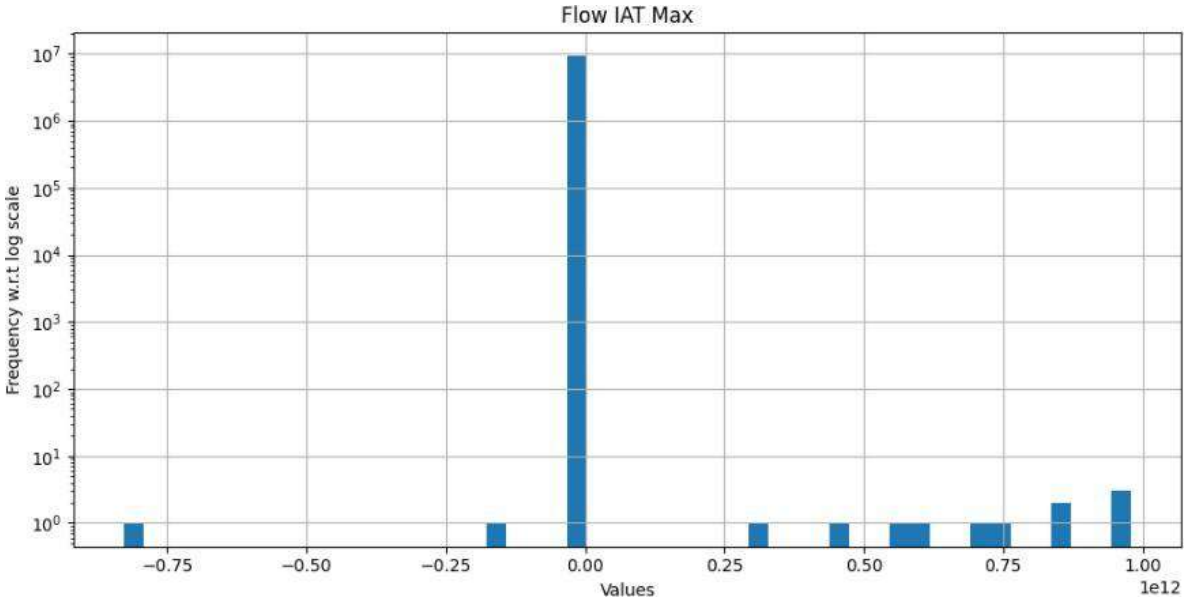


Figure 4.4.16 Histogram of Flow IAT Max plotted on log scale

- 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

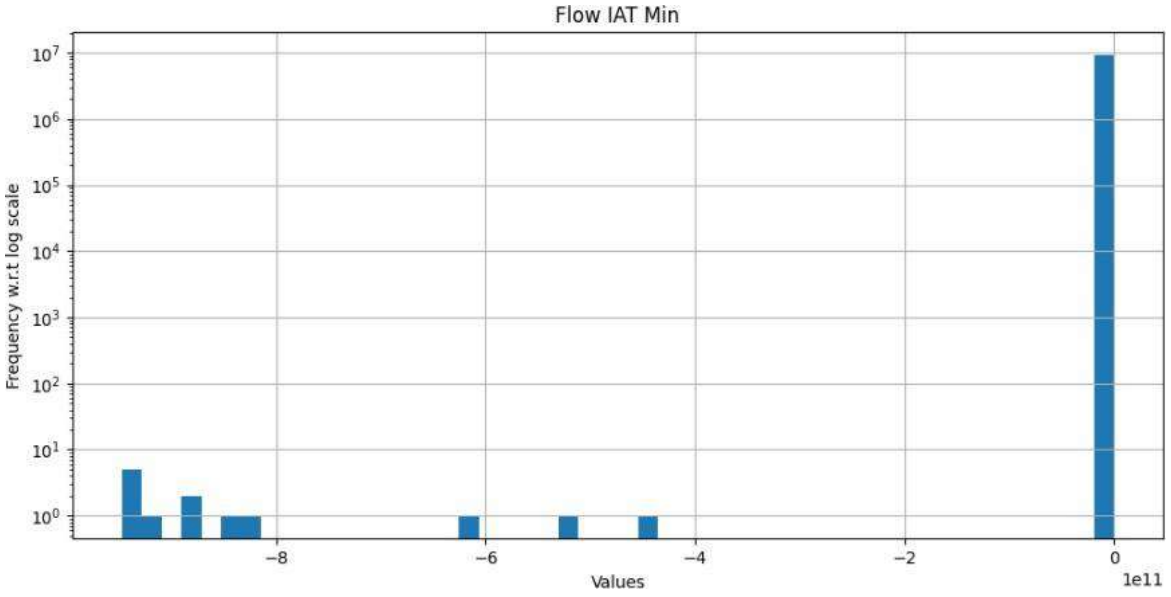


Figure 4.4.17 Histogram of Flow IAT Min plotted on log scale

- Peak was observed around Flow 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 Total: Total time between forward packets

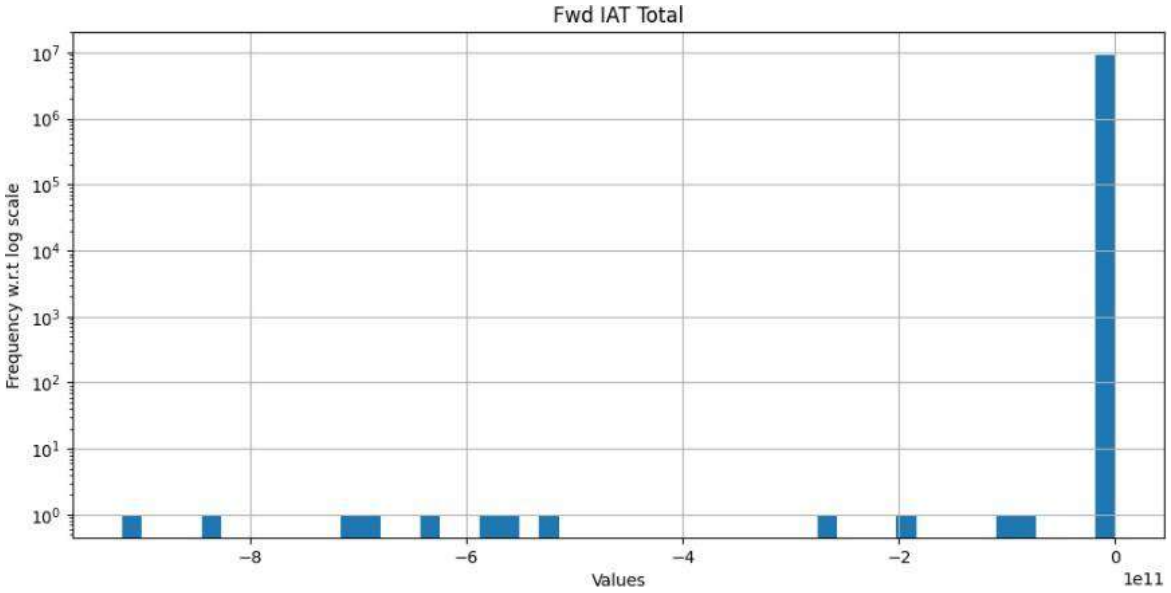


Figure 4.4.18 Histogram of Fwd IAT Total plotted on log scale

- 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

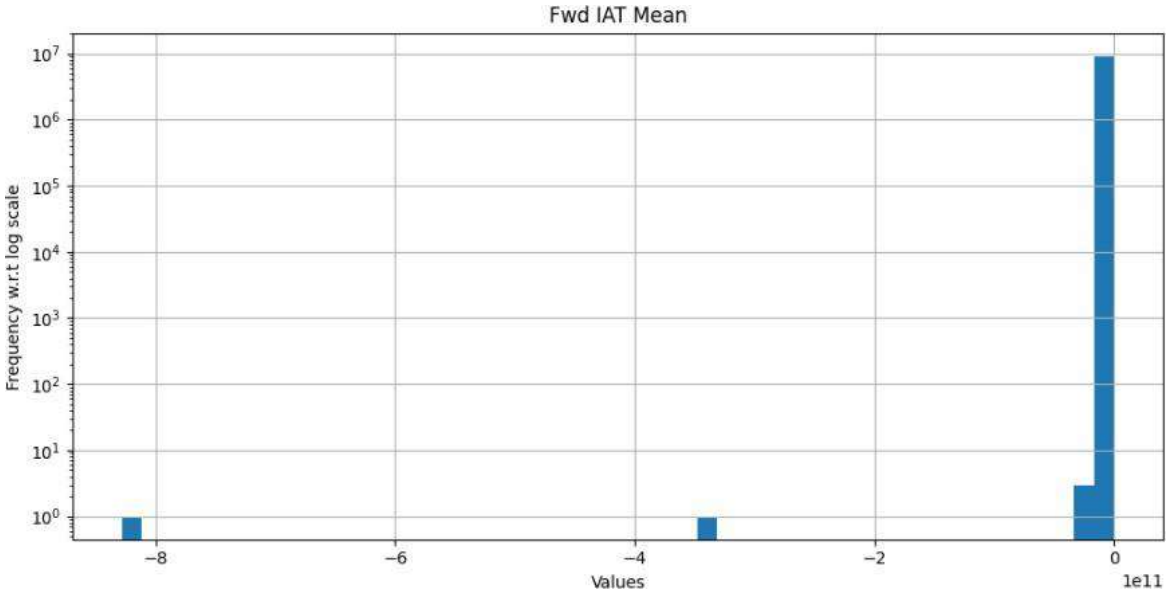


Figure 4.4.19 Histogram of Fwd IAT Mean plotted on log scale

- 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

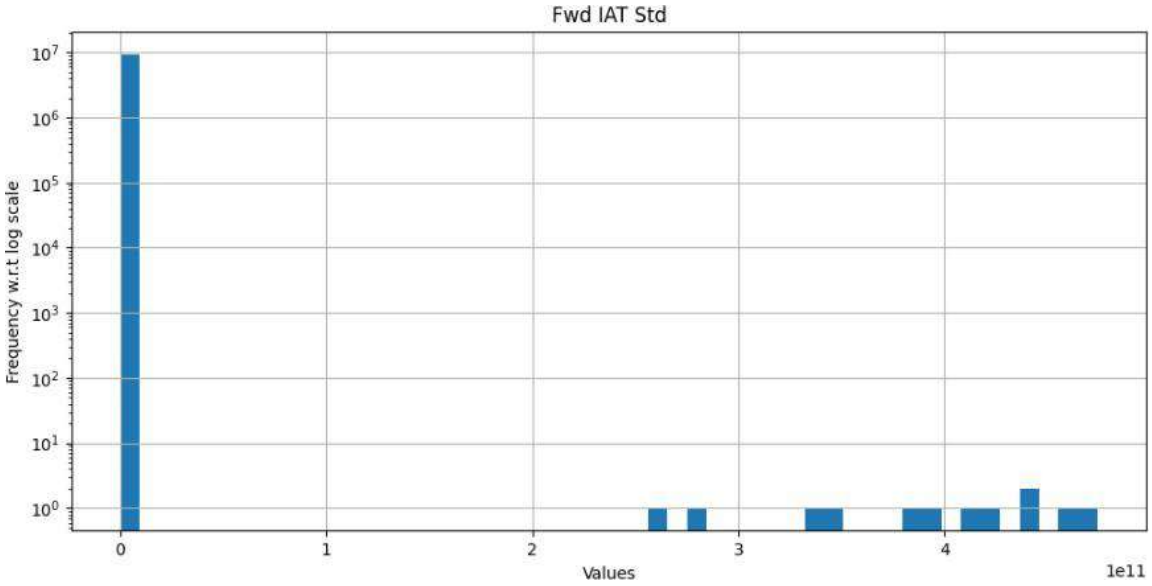


Figure 4.4.20 Histogram of Fwd IAT Std plotted on log scale

- Peak was observed around Fwd IAT Std=0.
- There are small number of observations in the range: Fwd IAT Std>=2 and Fwd IAT Std<=3, Fwd IAT Std>=3 and Fwd IAT Std<=4, Fwd IAT Std>4.

Fwd IAT Max: Maximum time between forward packets

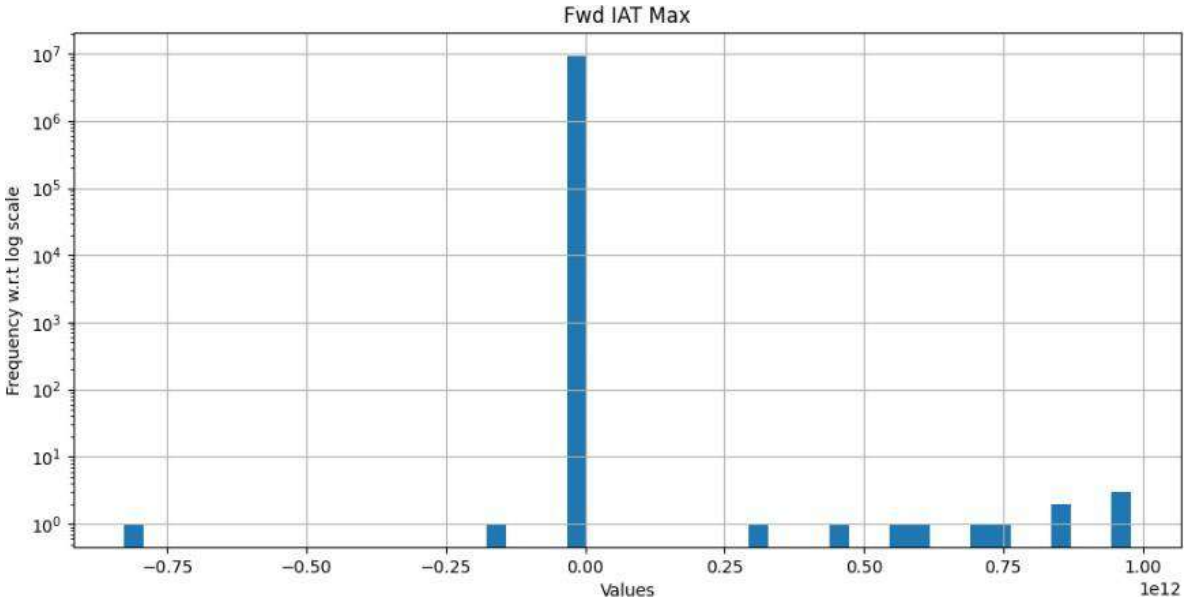


Figure 4.4.21 Histogram of Fwd IAT Max plotted on log scale

- 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

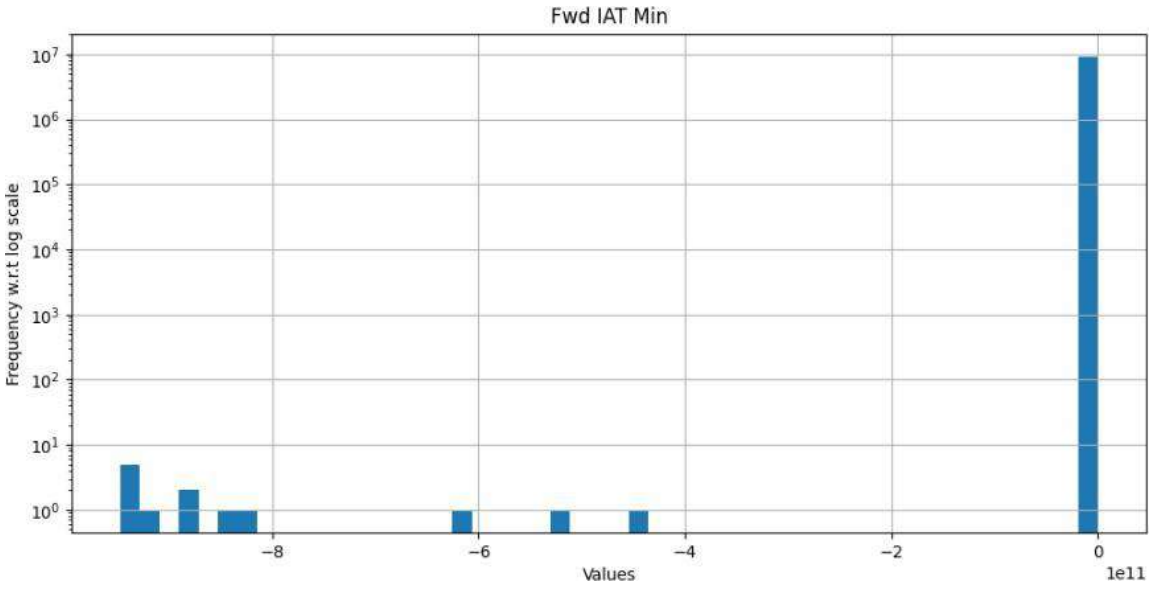


Figure 4.4.22 Histogram of Fwd IAT Min plotted on log scale

- Peak was observed around Fwd 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

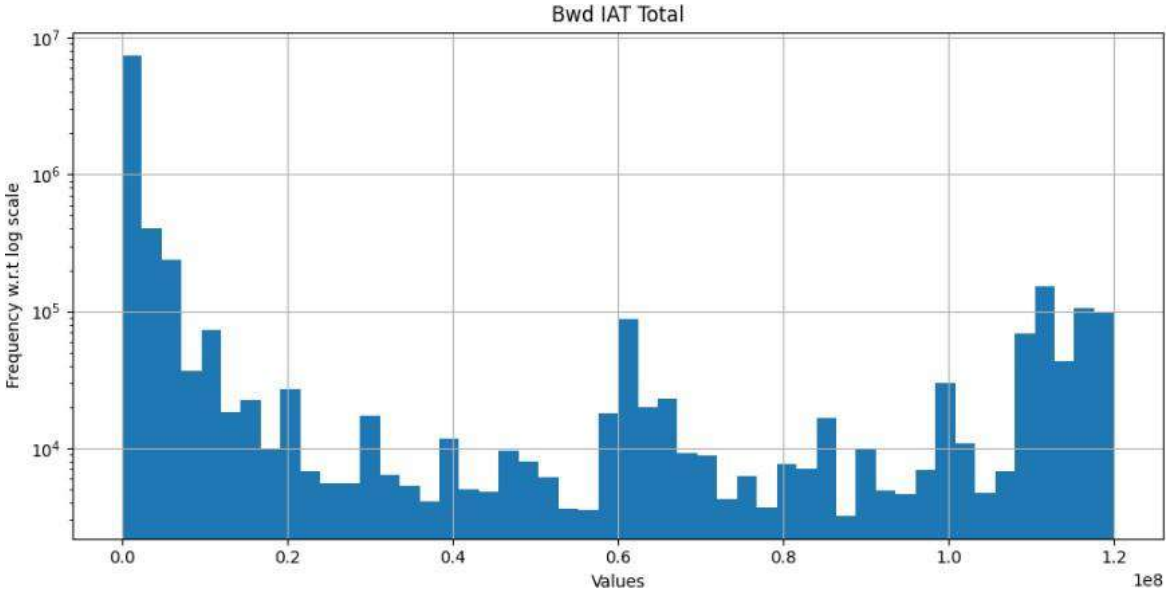


Figure 4.4.23 Histogram of Bwd IAT Total plotted on log scale

- 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

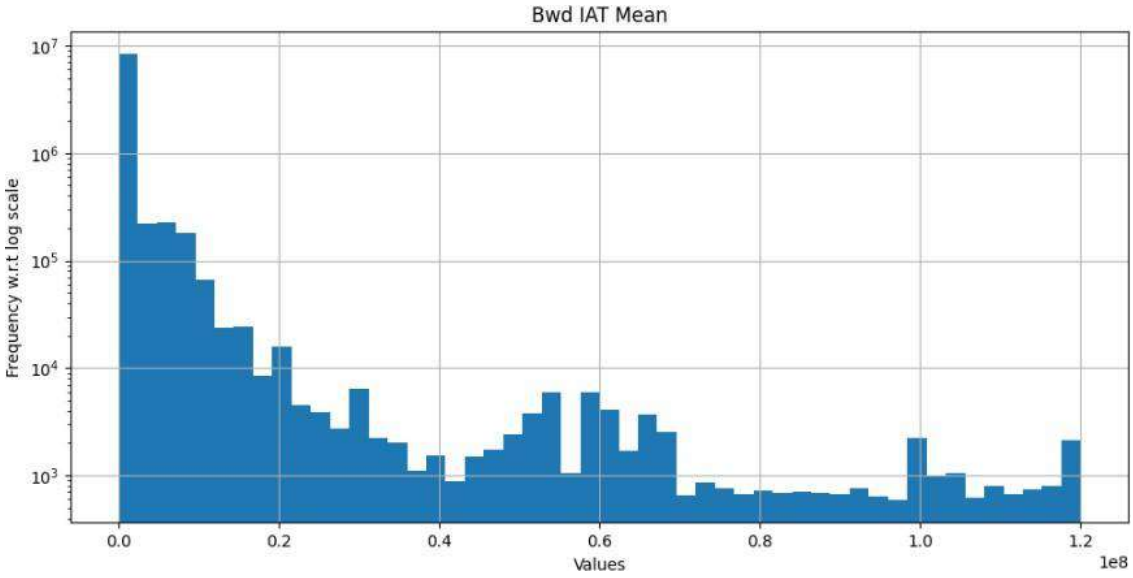


Figure 4.4.24 Histogram of Bwd IAT Mean plotted on log scale

- 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

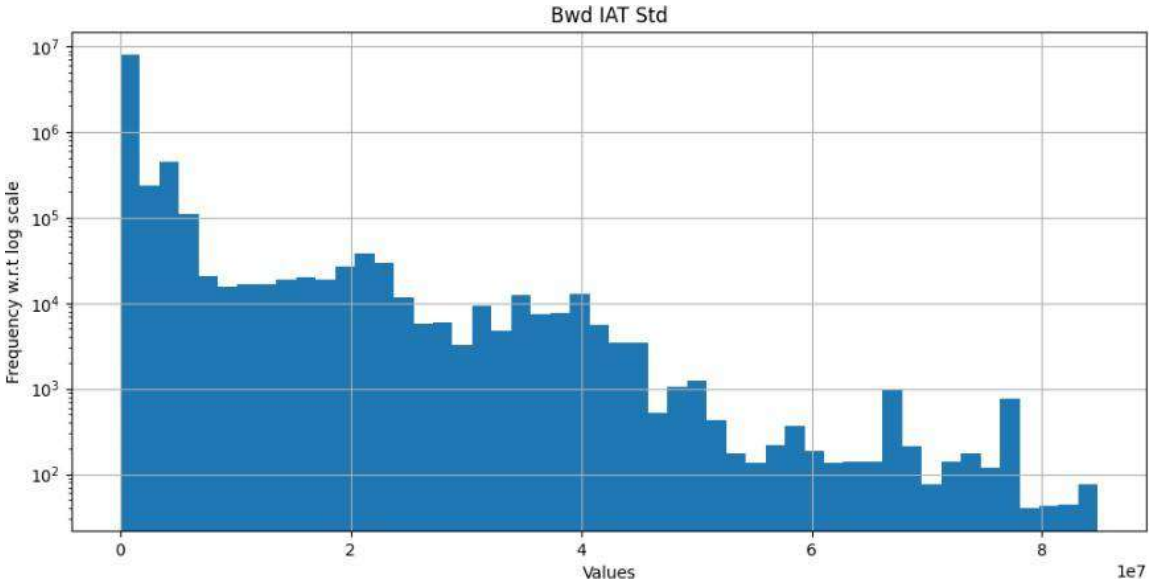


Figure 4.4.25 Histogram of Bwd IAT Std plotted on log scale

- 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>=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

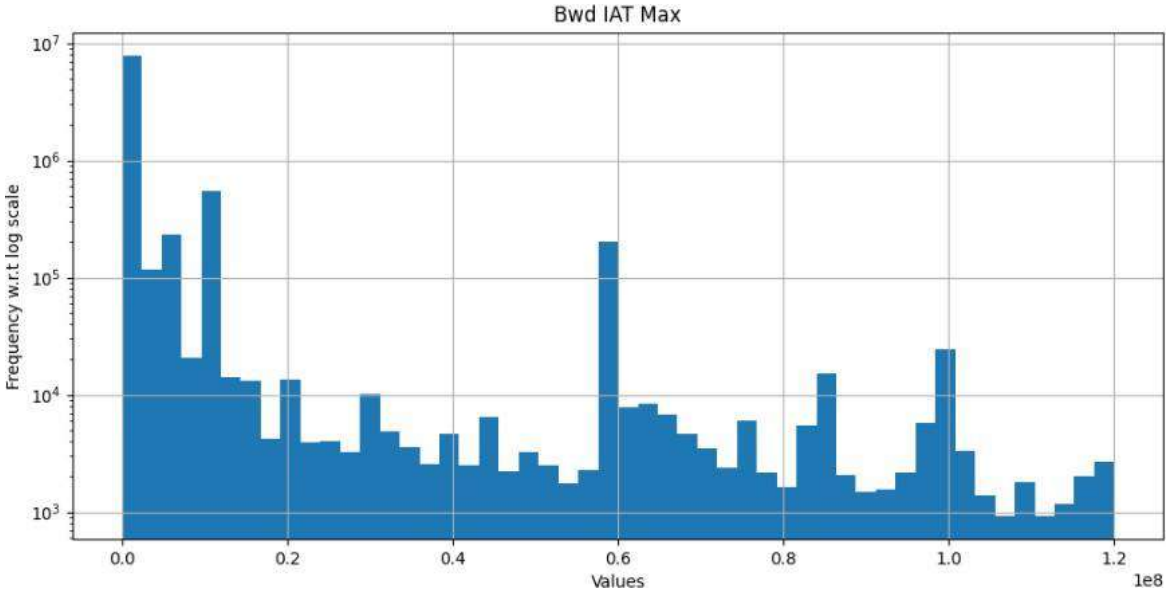


Figure 4.4.26 Histogram of Bwd IAT Max plotted on log scale

- 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

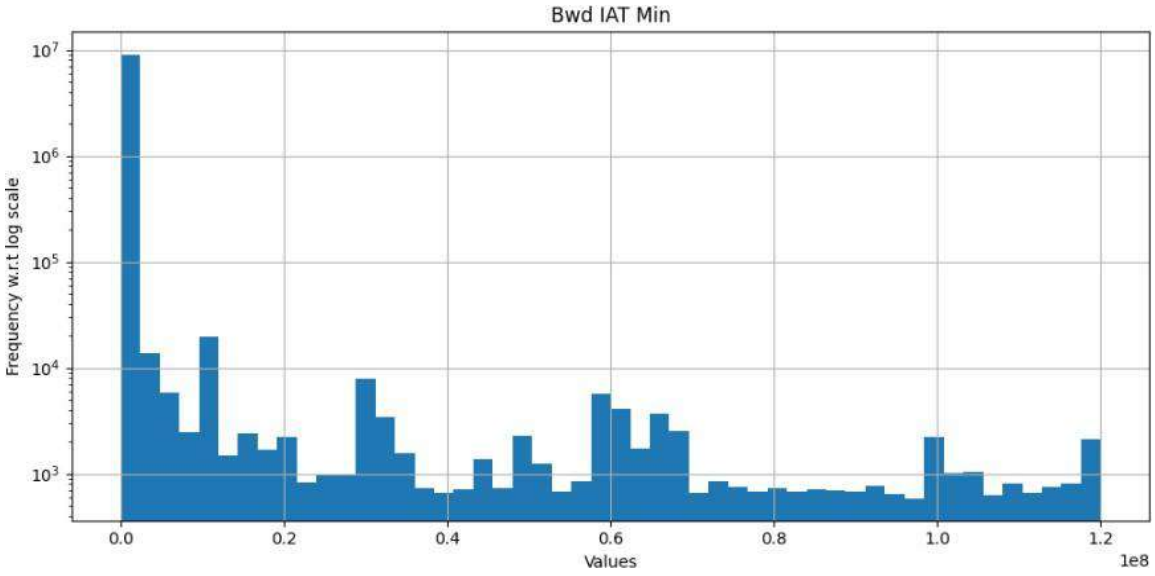


Figure 4.4.27 Histogram of Bwd IAT Min plotted on log scale

- 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

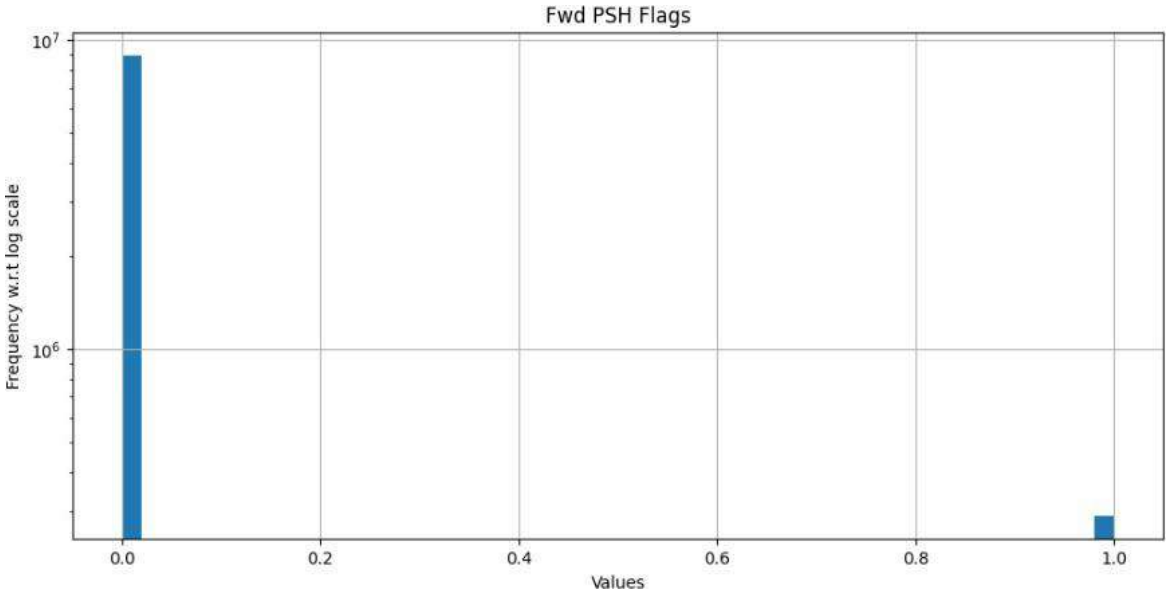


Figure 4.4.28 Histogram of Fwd PSH Flags plotted on log scale

- 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

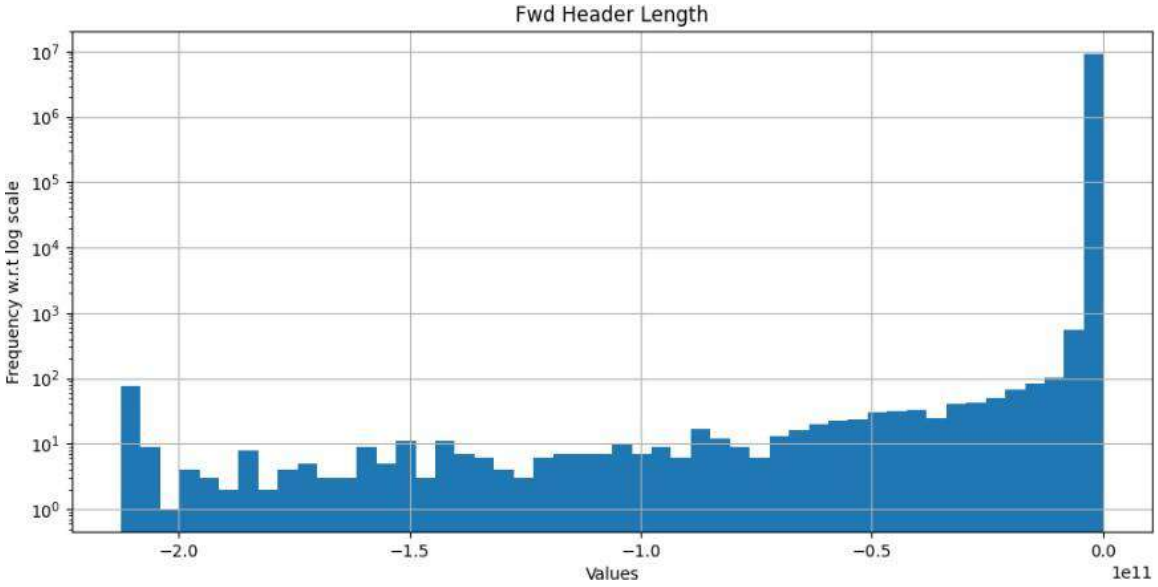


Figure 4.4.29 Histogram of Fwd Header Length plotted on log scale

- 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

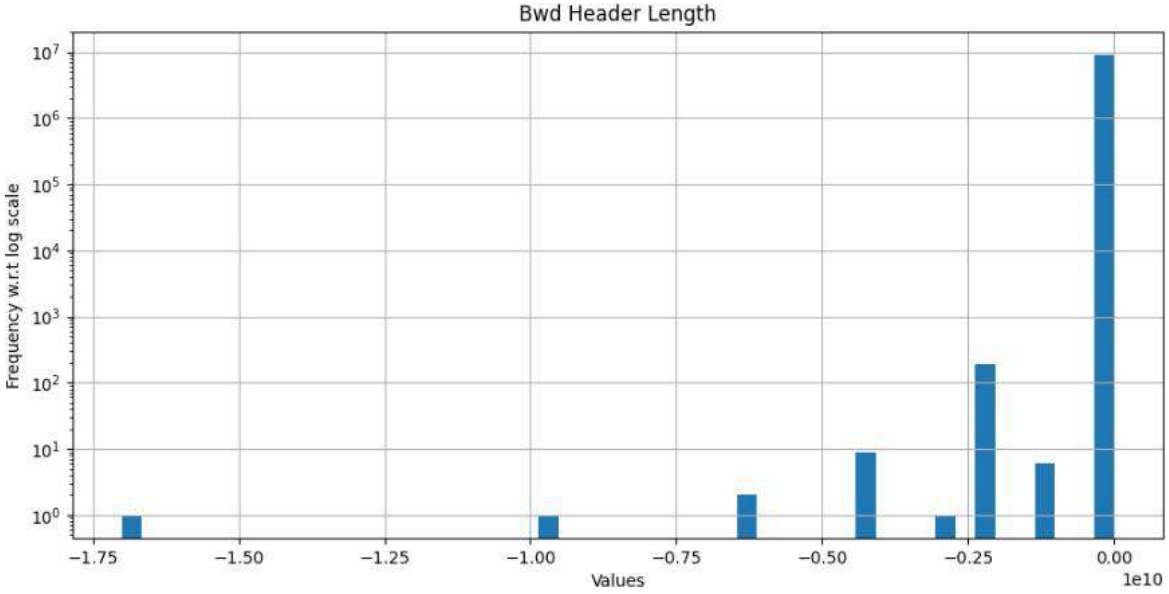


Figure 4.4.30 Histogram of Bwd Header Length plotted on log scale

- 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

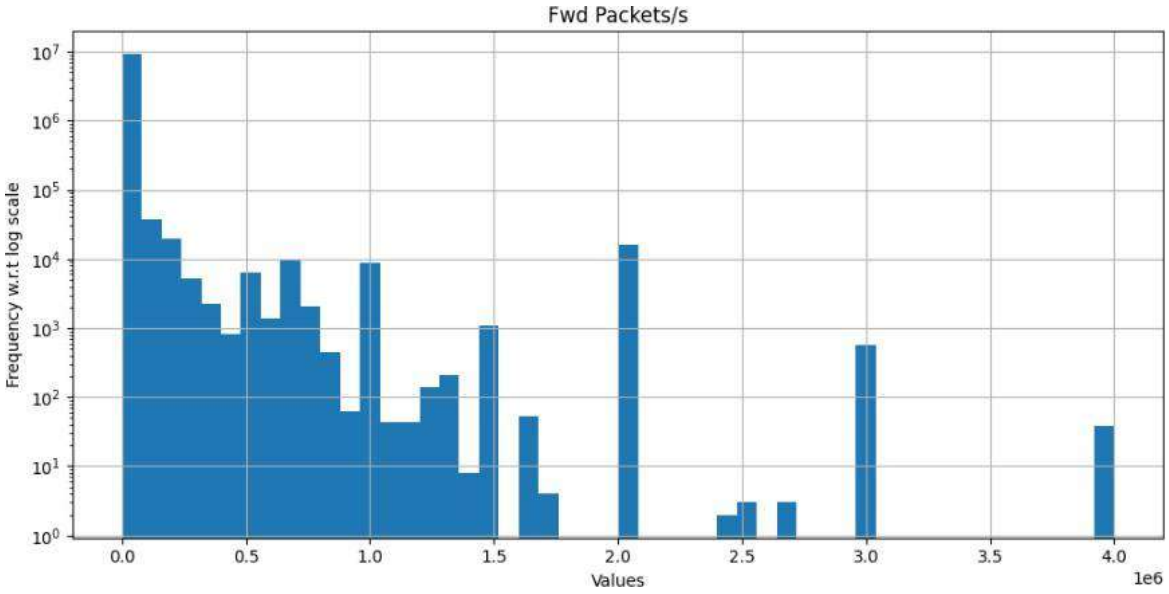


Figure 4.4.31 Histogram of Fwd Packets/s plotted on log scale

- 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

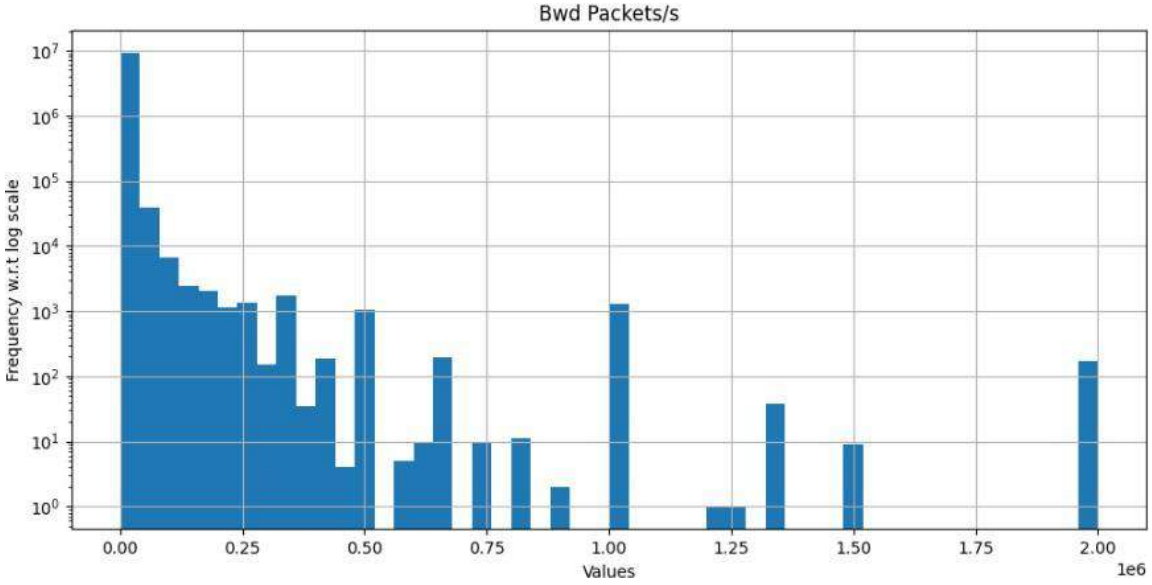


Figure 4.4.32 Histogram of Bwd Packets/s plotted on log scale

- 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

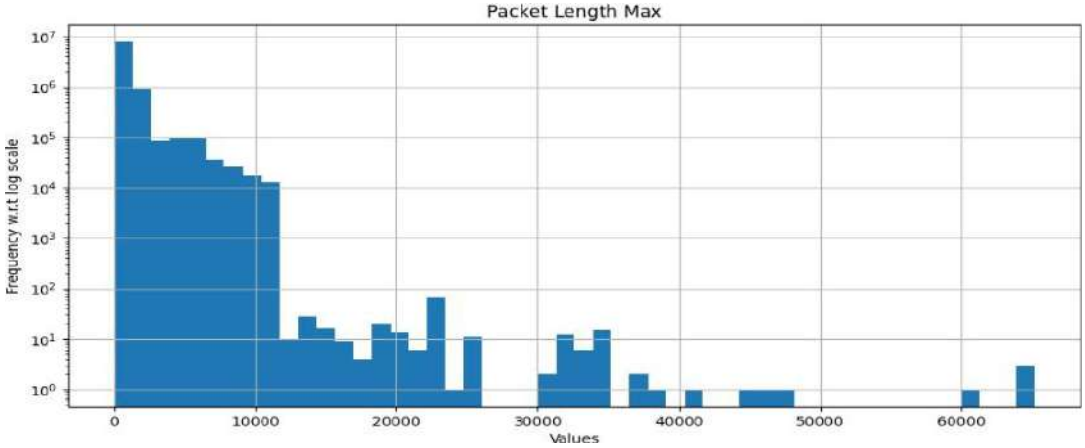


Figure 4.4.33 Histogram of Packet Length Max plotted on log scale

- 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>=0 and Packet Length Max<=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

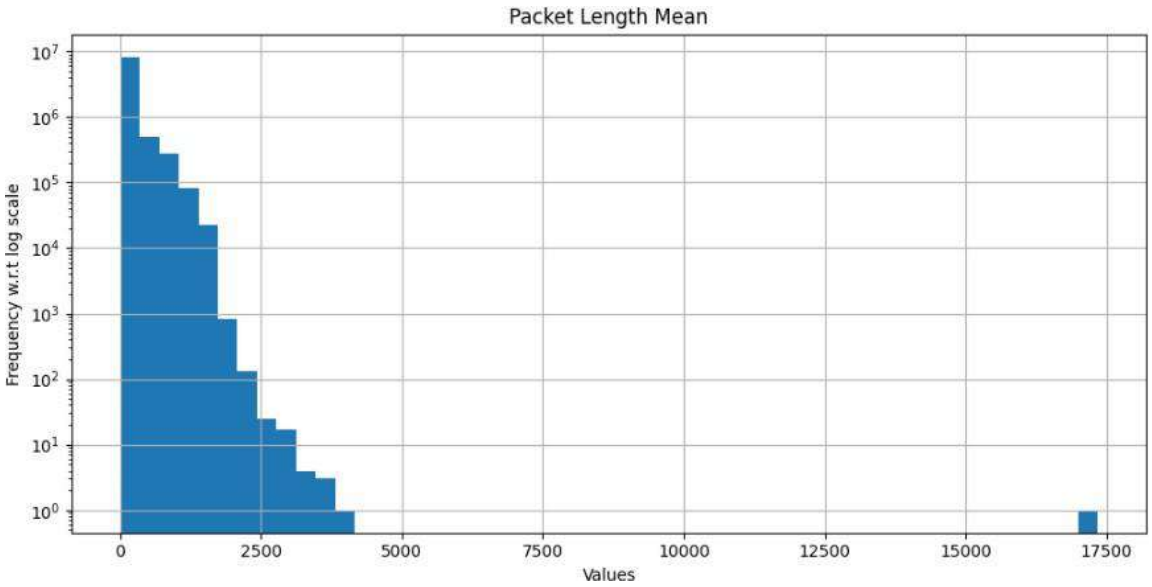


Figure 4.4.34 Histogram of Packet Length Mean plotted on log scale

- 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>=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

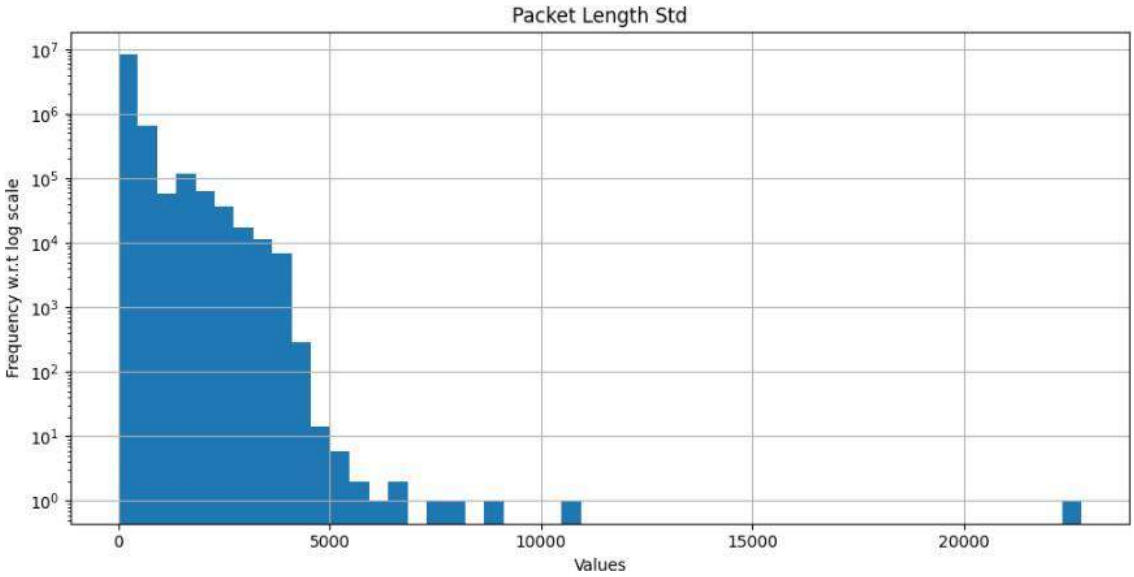


Figure 4.4.35 Histogram of Packet Length Std plotted on log scale

- 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

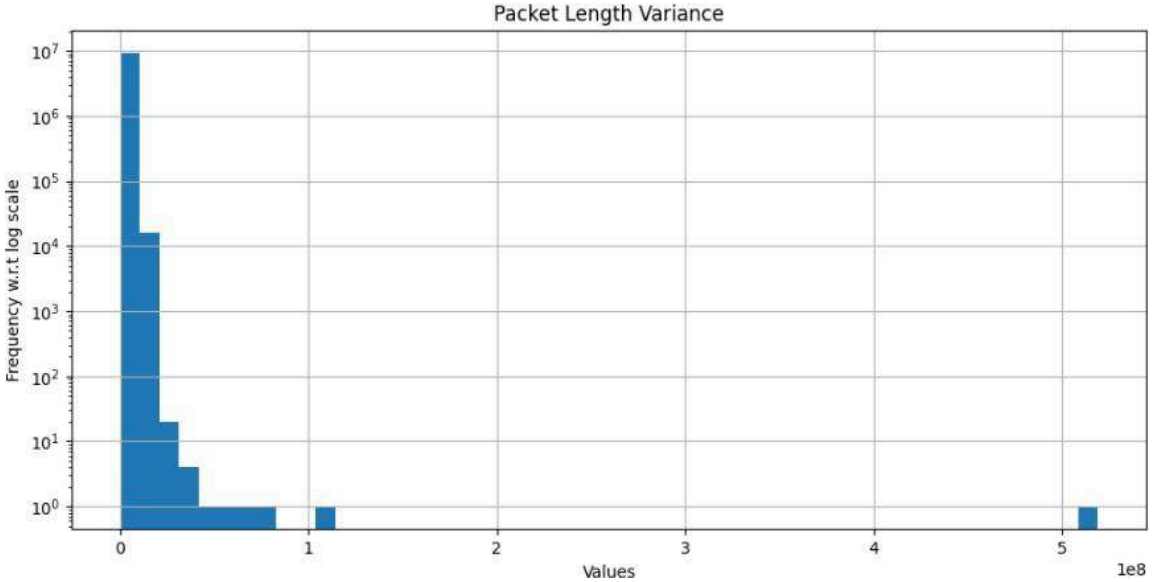


Figure 4.4.36 Histogram of Packet Length Variance plotted on log scale

- 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

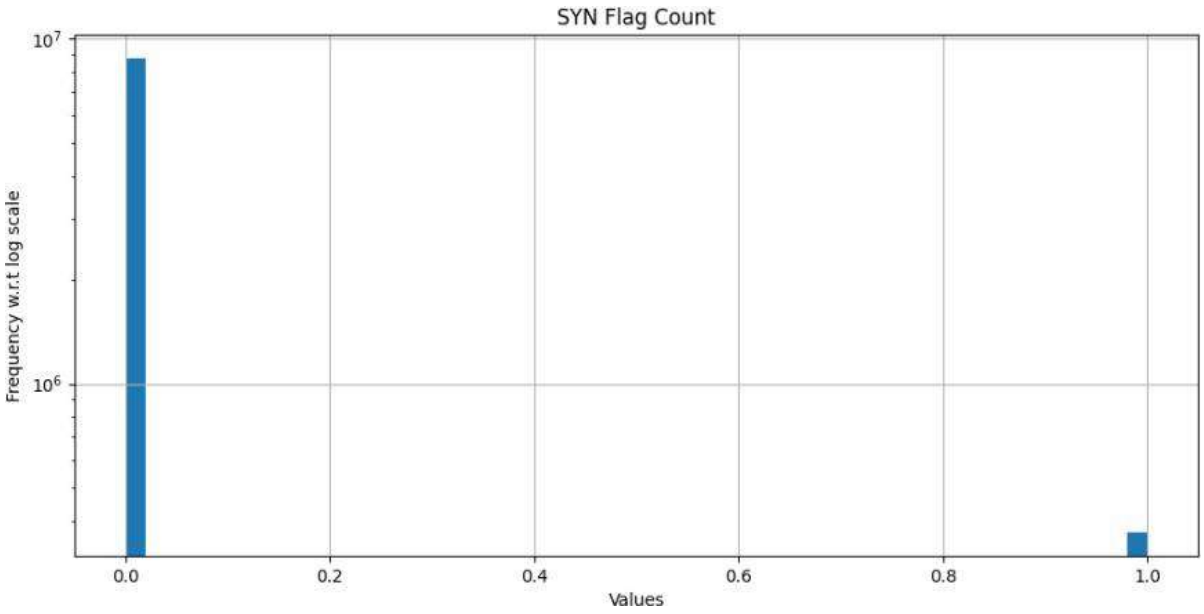


Figure 4.4.37 Histogram of SYN Flag Count plotted on log scale

- 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

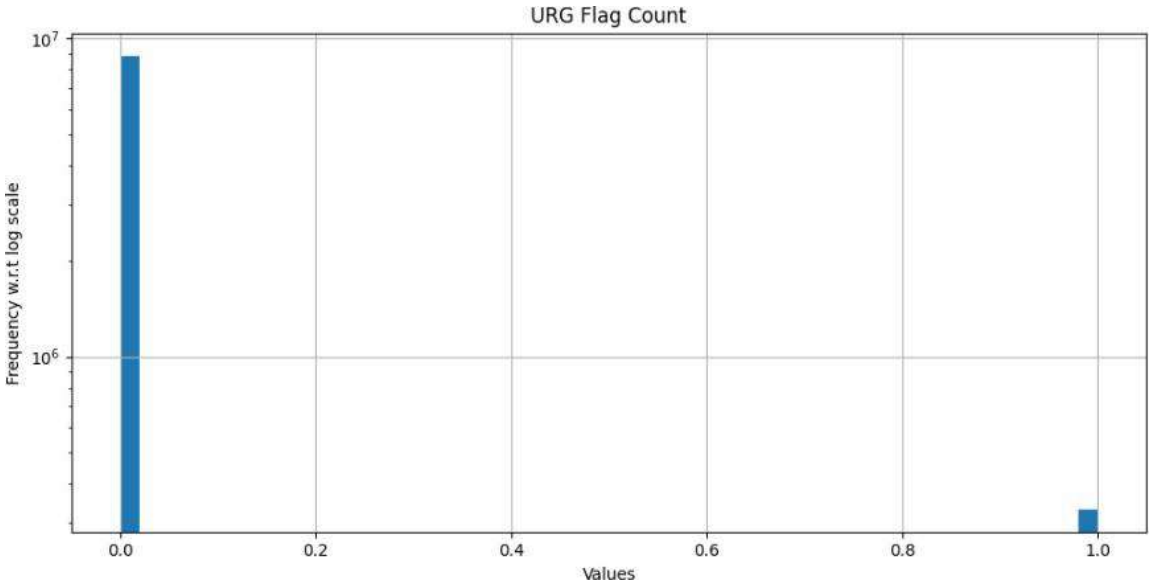


Figure 4.4.38 Histogram of URG Flag Count plotted on log scale

- 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

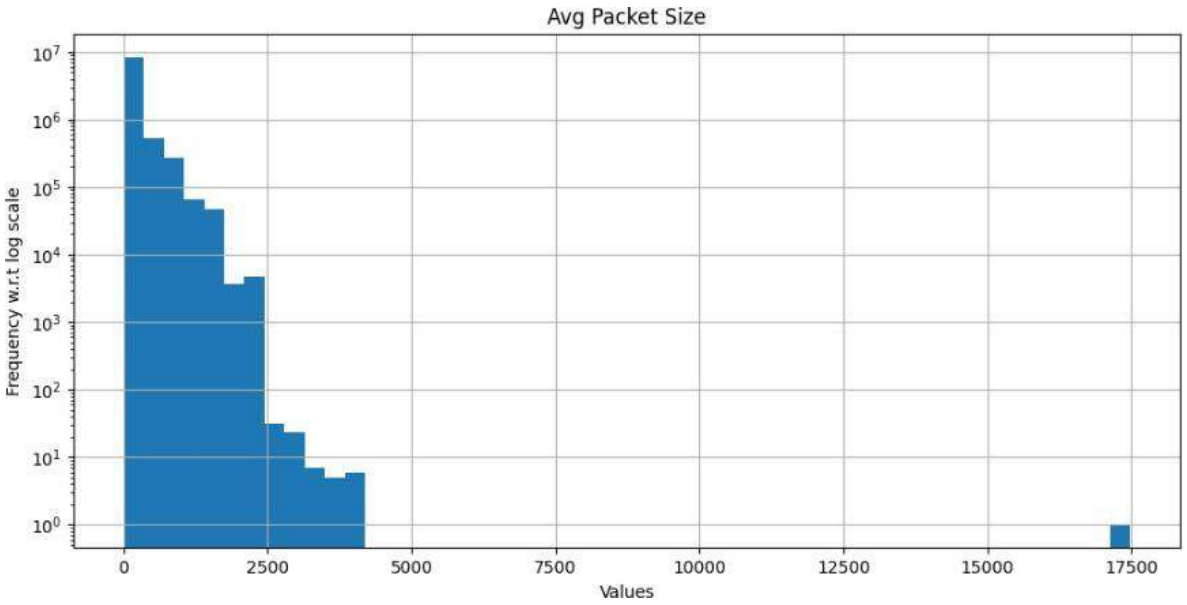


Figure 4.4.39 Histogram of Avg Packet Size plotted on log scale

- 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

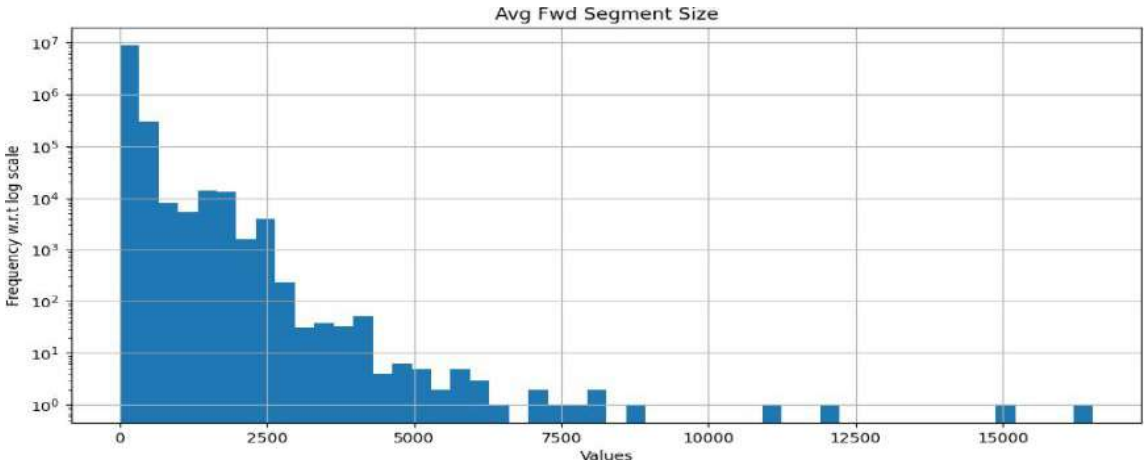


Figure 4.4.40 Histogram of Avg Fwd Segment Size plotted on log scale

- 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 Segment Size>=15000. This may indicate outlier in the data.

Avg Bwd Segment Size: Average backward segment size

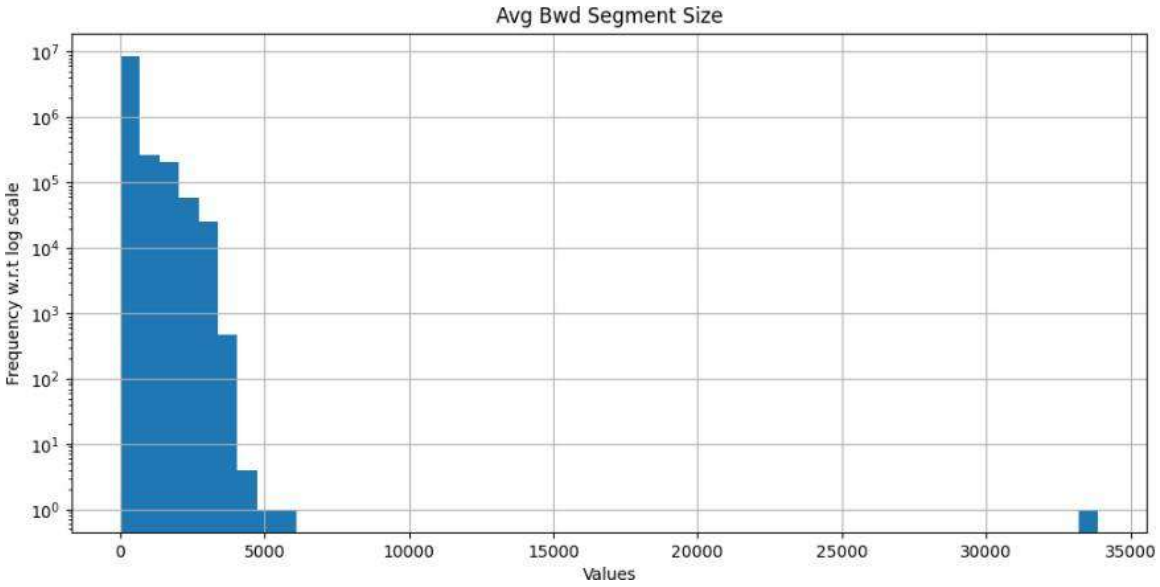


Figure 4.4.41 Histogram of Avg Bwd Segment Size plotted on log scale

- 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

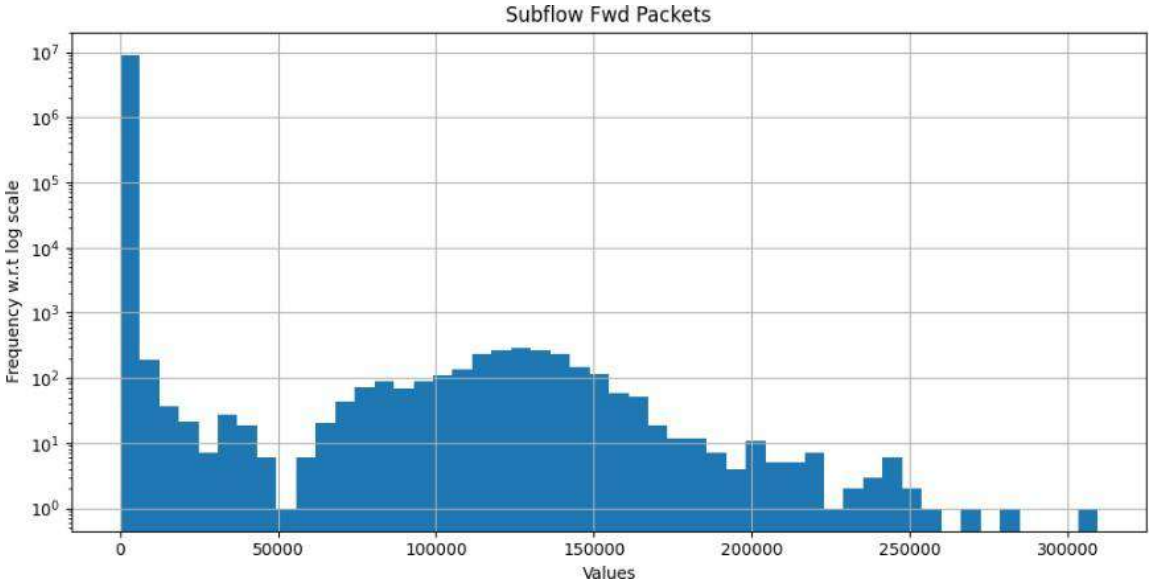


Figure 4.4.42 Histogram of Subflow Fwd Packets plotted on log scale

- 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 plateau region observed between Subflow Fwd Packets>=100000 and Subflow Fwd Packets<=150000.
- There were decline in the number of results observed after Subflow Fwd Packets>=150000.
- There are many values between Subflow Fwd Packets>=50000 and Subflow Fwd Packets<=150000.
- There is a value after Subflow Fwd Packets>300000. This may indicate outlier in the data.

Subflow Fwd Bytes: Subflow forward bytes

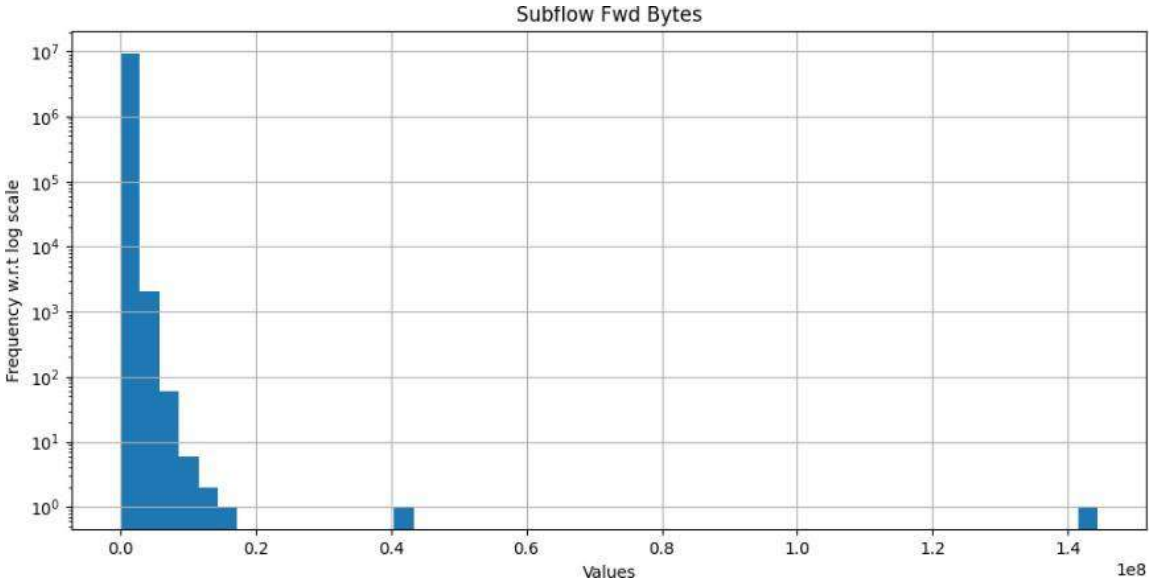


Figure 4.4.43 Histogram of Subflow Fwd Bytes plotted on log scale

- 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>=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

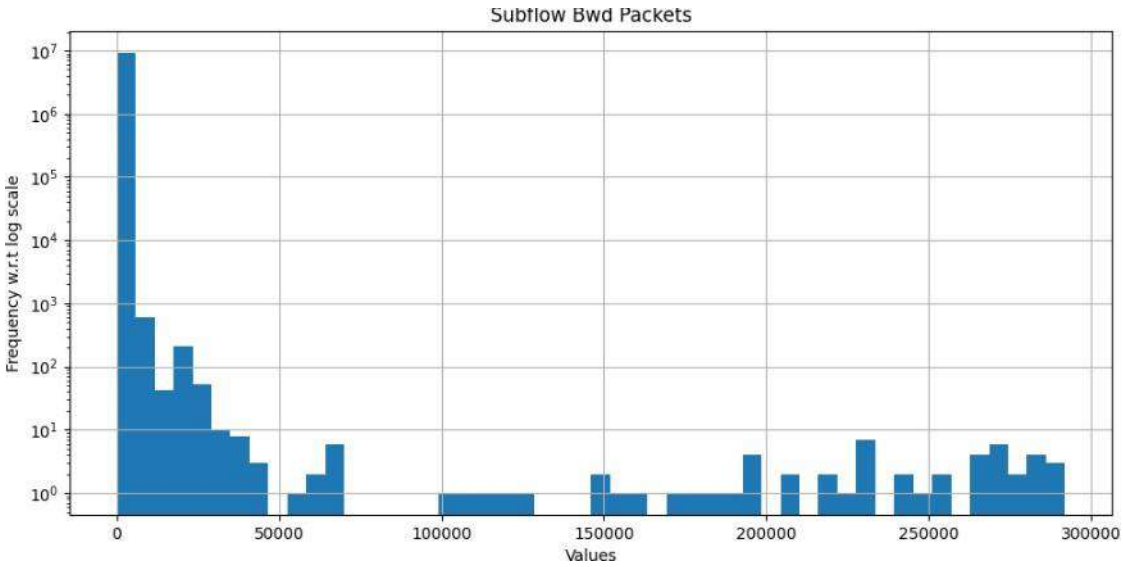


Figure 4.4.44 Histogram of Subflow Bwd Packets plotted on log scale

- 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

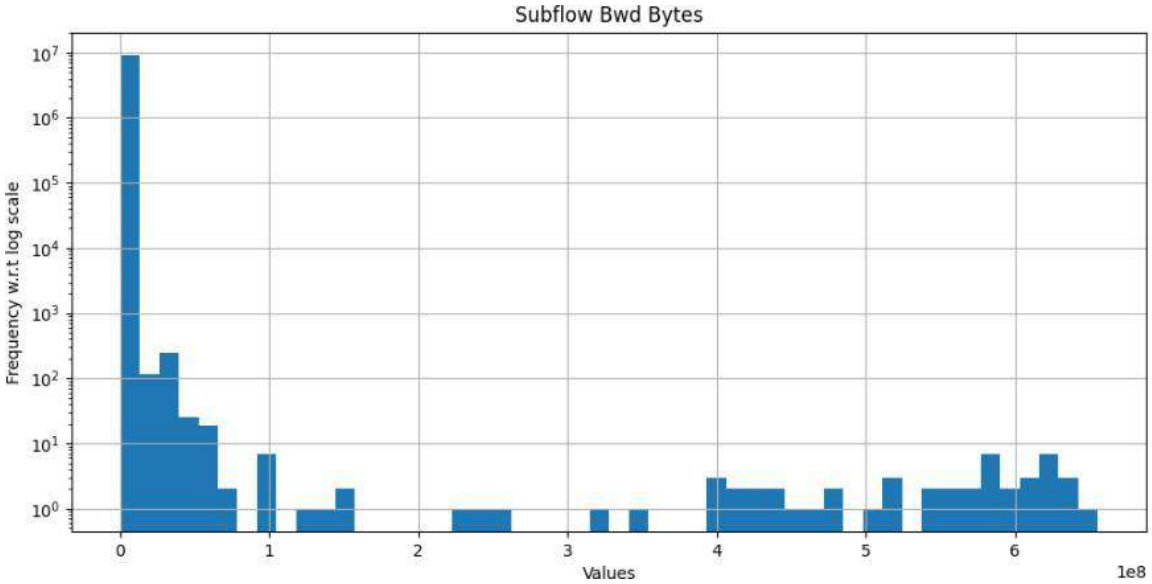


Figure 4.4.45 Histogram of Subflow Bwd Bytes plotted on log scale

- 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

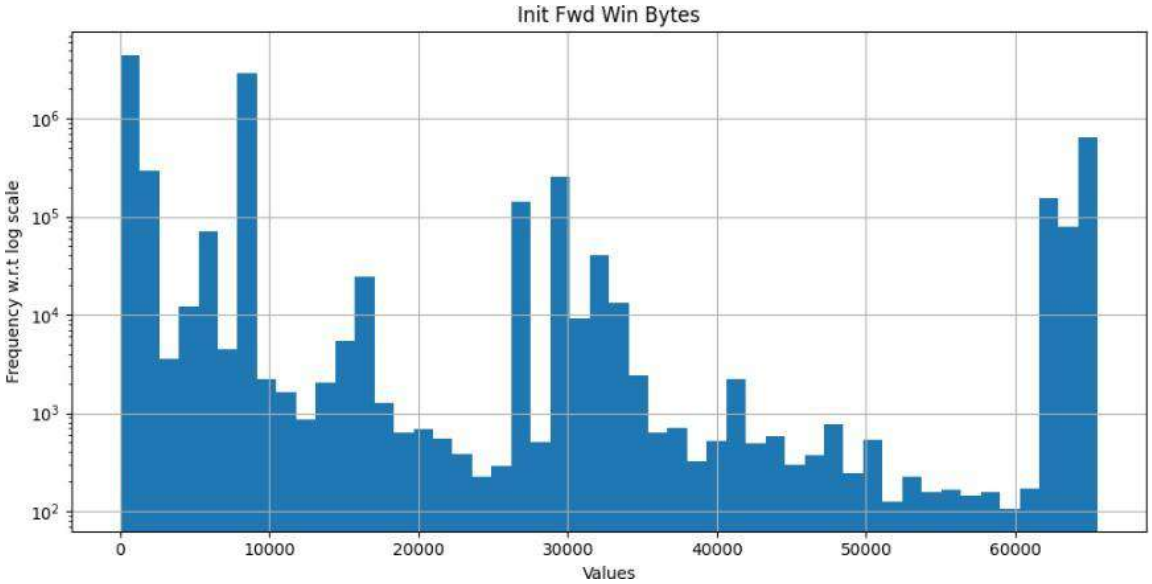


Figure 4.4.46 Histogram of Init Fwd Win Bytes plotted on log scale

- 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

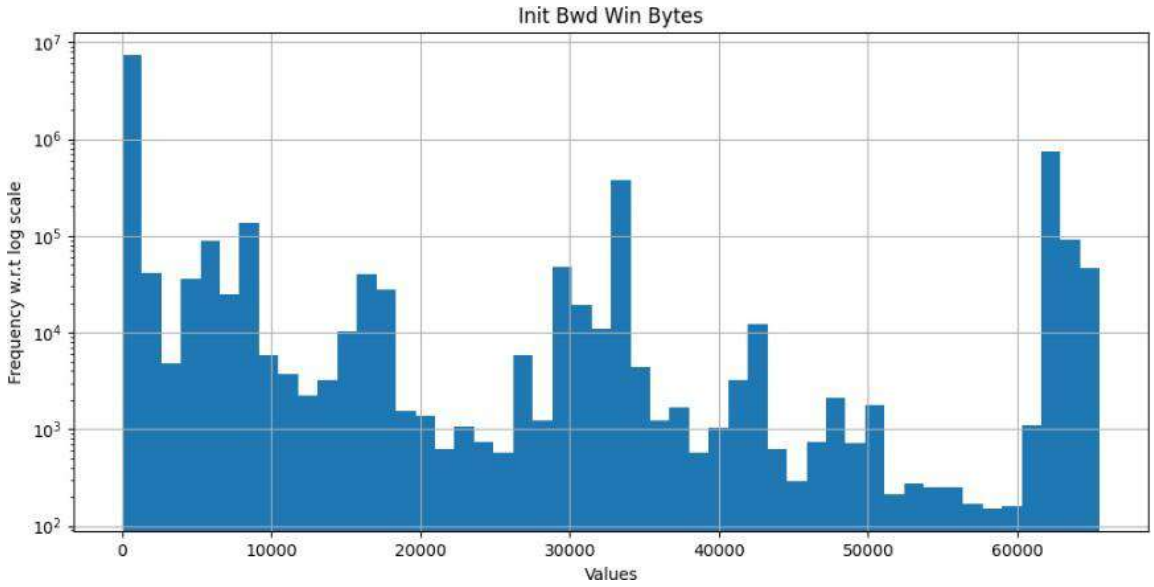


Figure 4.4.47 Histogram of Init Bwd Win Bytes plotted on log scale

- 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

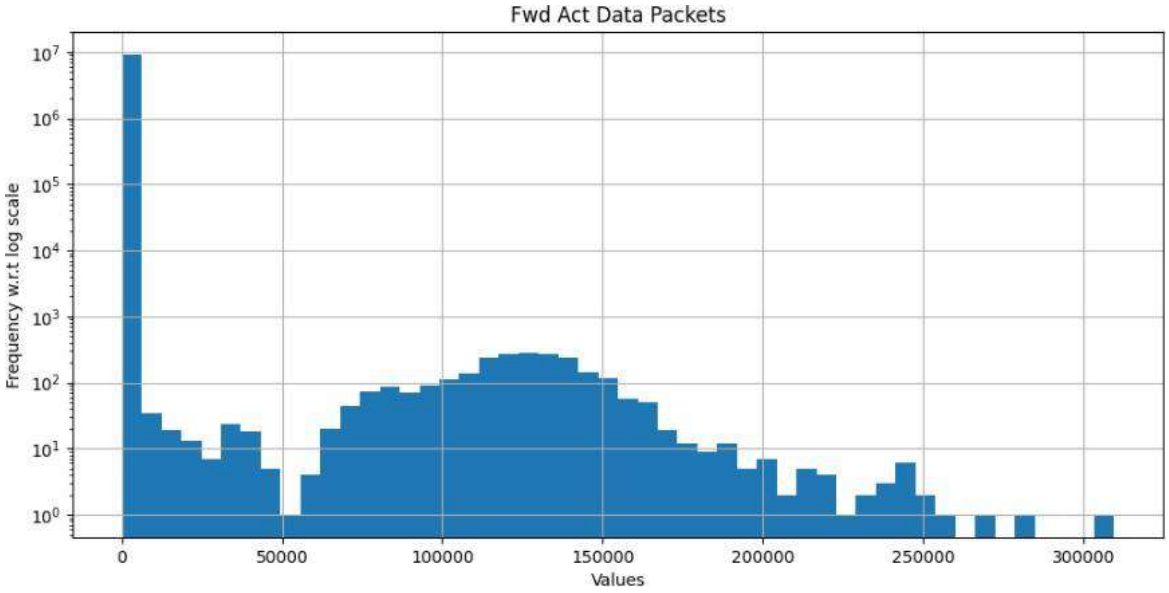


Figure 4.4.48 Histogram of Fwd Act Data Packets plotted on log scale

- 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>=100000 and Fwd Act Data Packets<=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

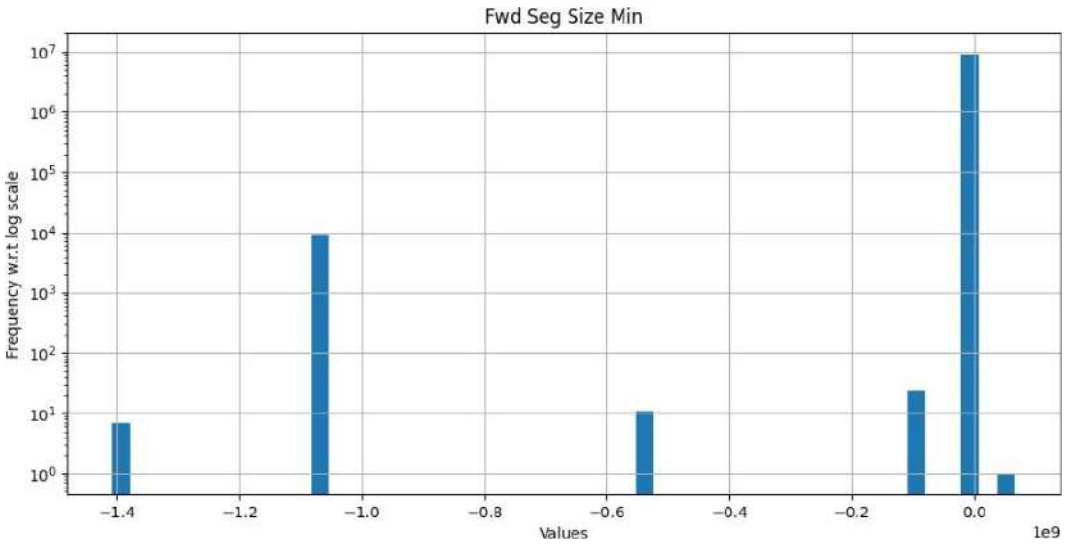


Figure 4.4.49 Histogram of Fwd Seg Size Min plotted on log scale

- 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

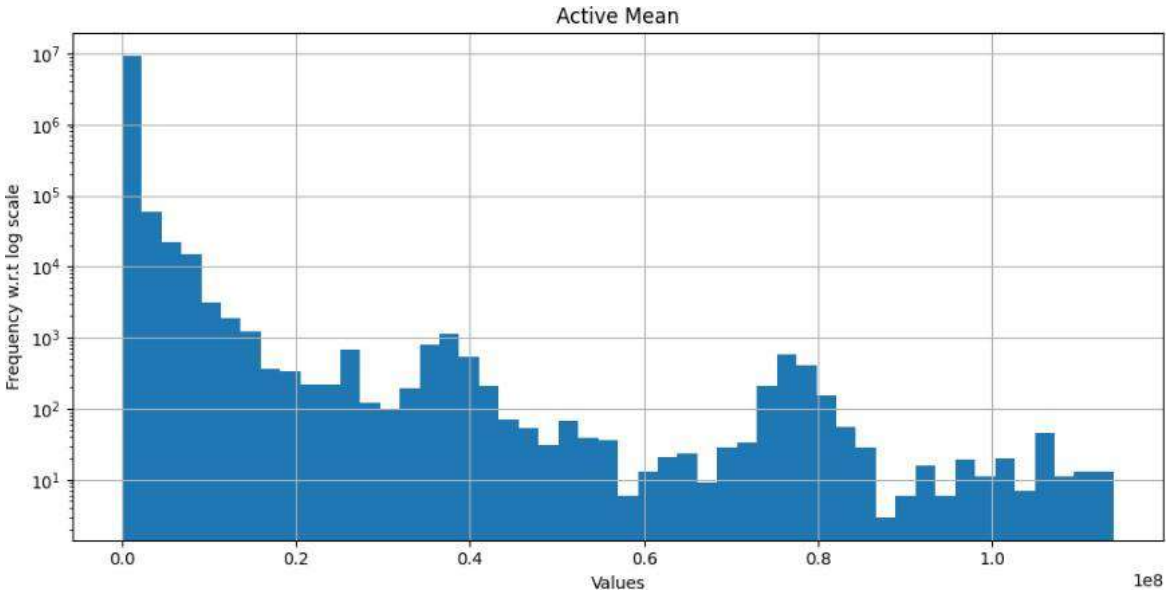


Figure 4.4.50 Histogram of Active Mean plotted on log scale

- 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

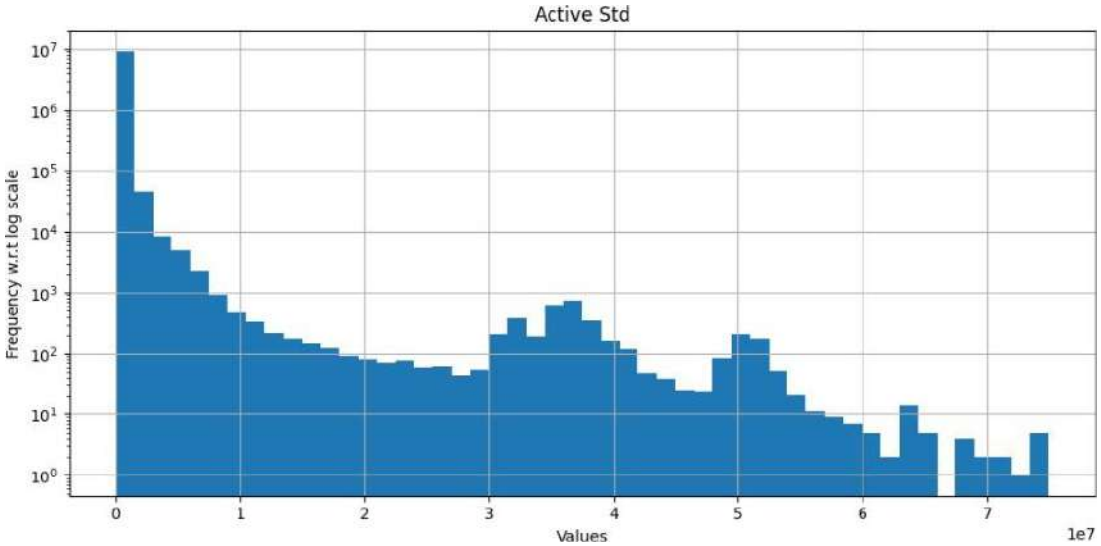


Figure 4.4.51 Histogram of Active Std plotted on log scale

- 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

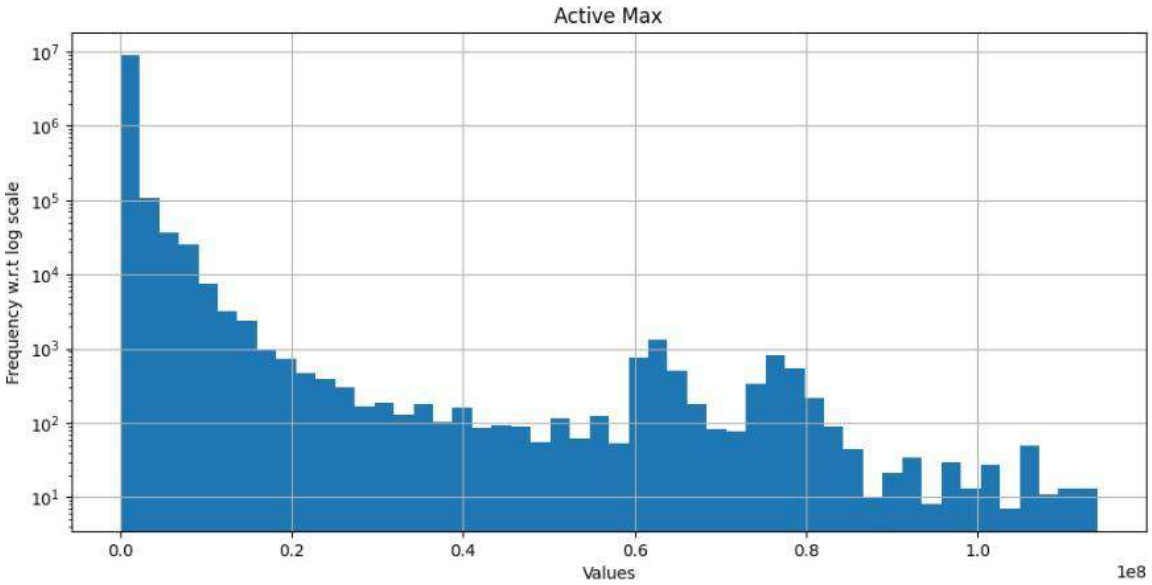


Figure 4.4.52 Histogram of Active Max plotted on log scale

- 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

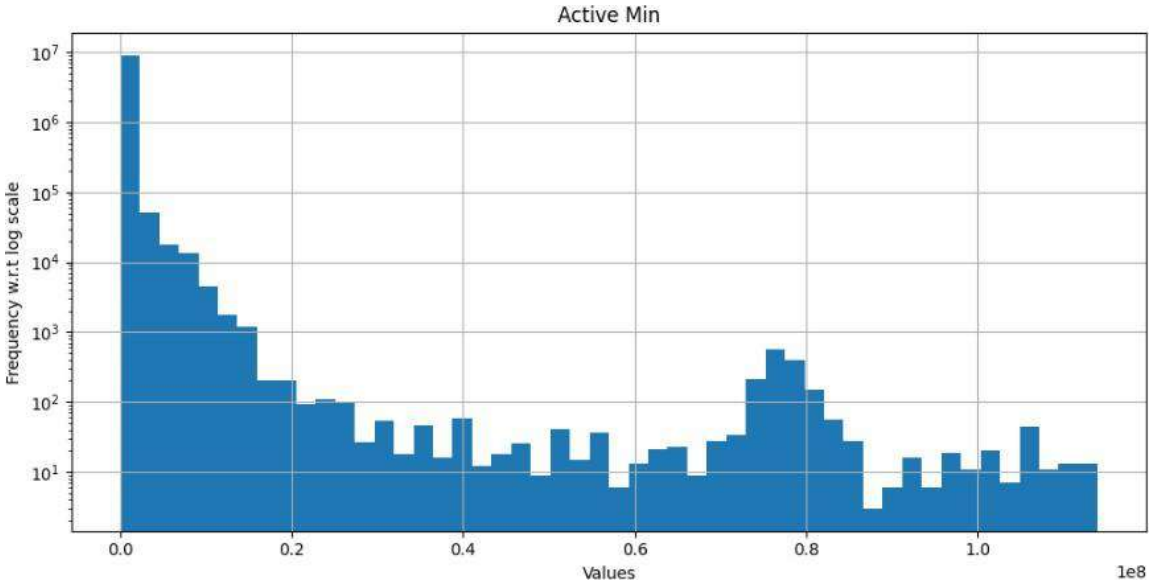


Figure 4.4.53 Histogram of Active Min plotted on log scale

- 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

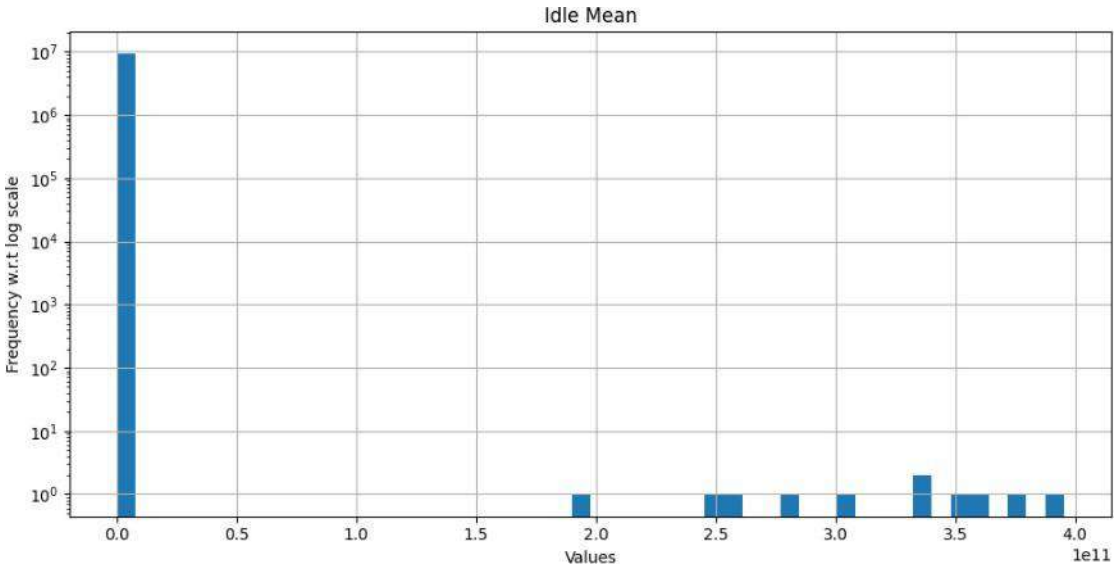


Figure 4.4.54 Histogram of Idle Mean plotted on log scale

- 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

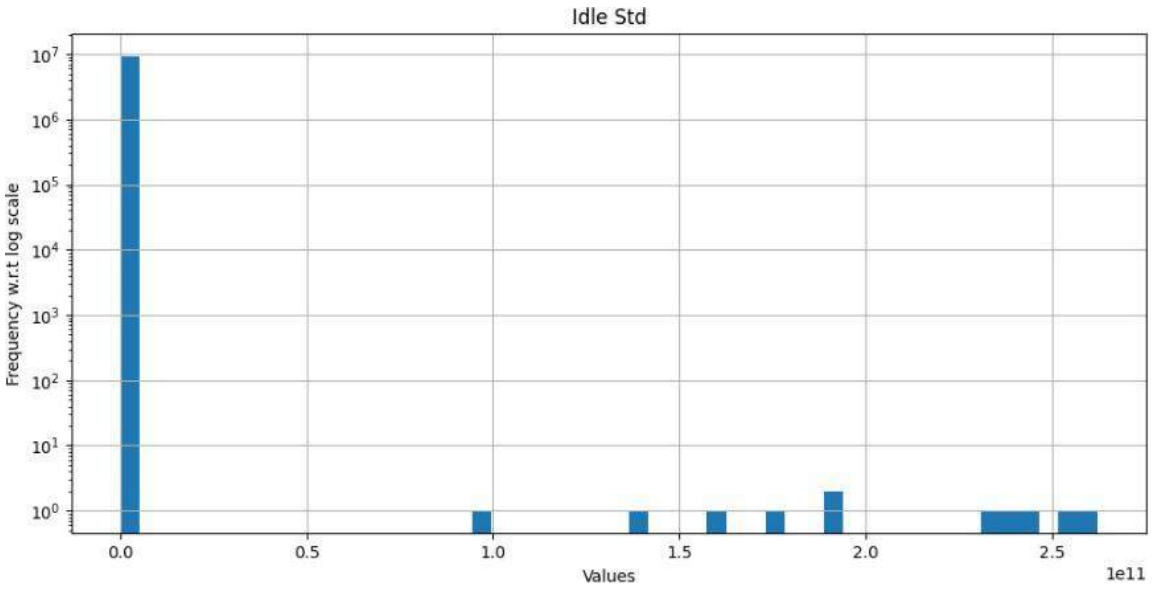


Figure 4.4.55 Histogram of Idle Std plotted on log scale

- 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

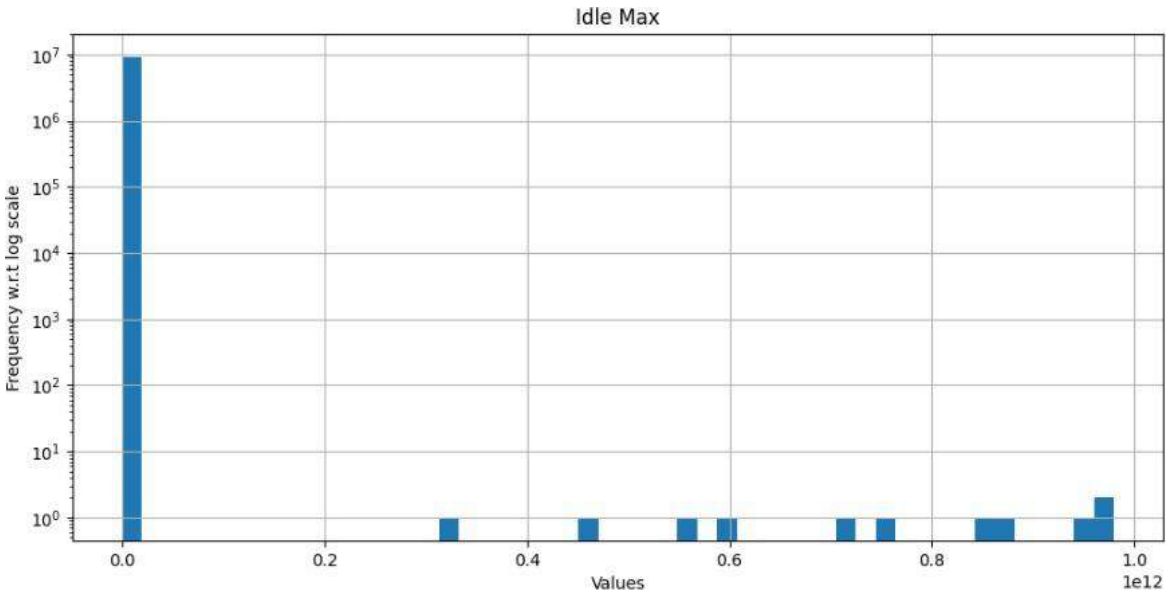


Figure 4.4.56 Histogram of Idle Max plotted on log scale

- 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

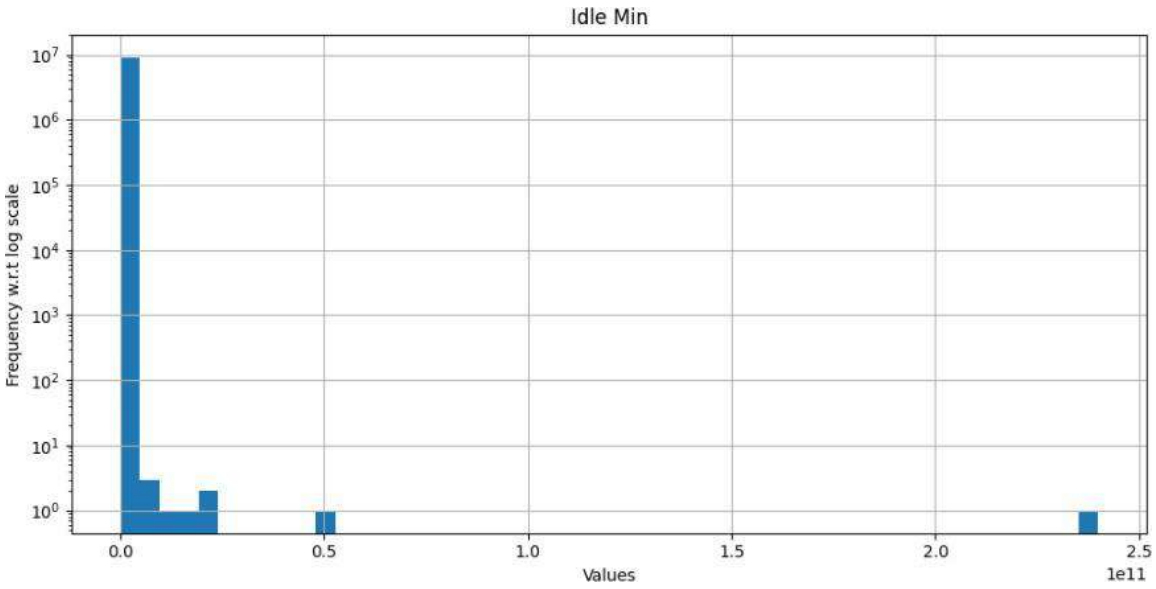


Figure 4.4.57 Histogram of Idle Min plotted on log scale

- 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.

4.5 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.

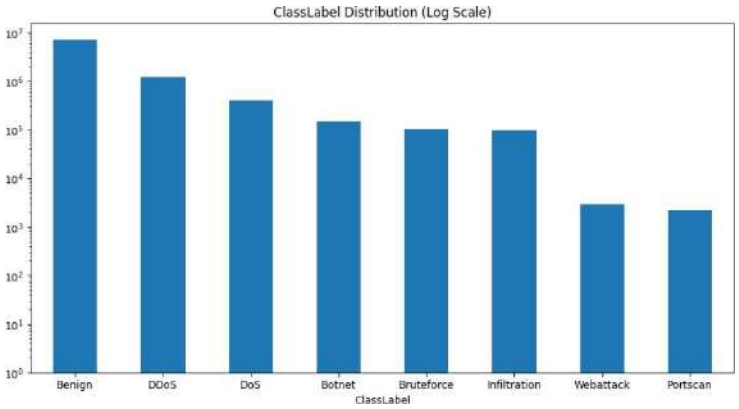


Figure 4.5.1 Bar chart of ClassLabel plotted on log scale

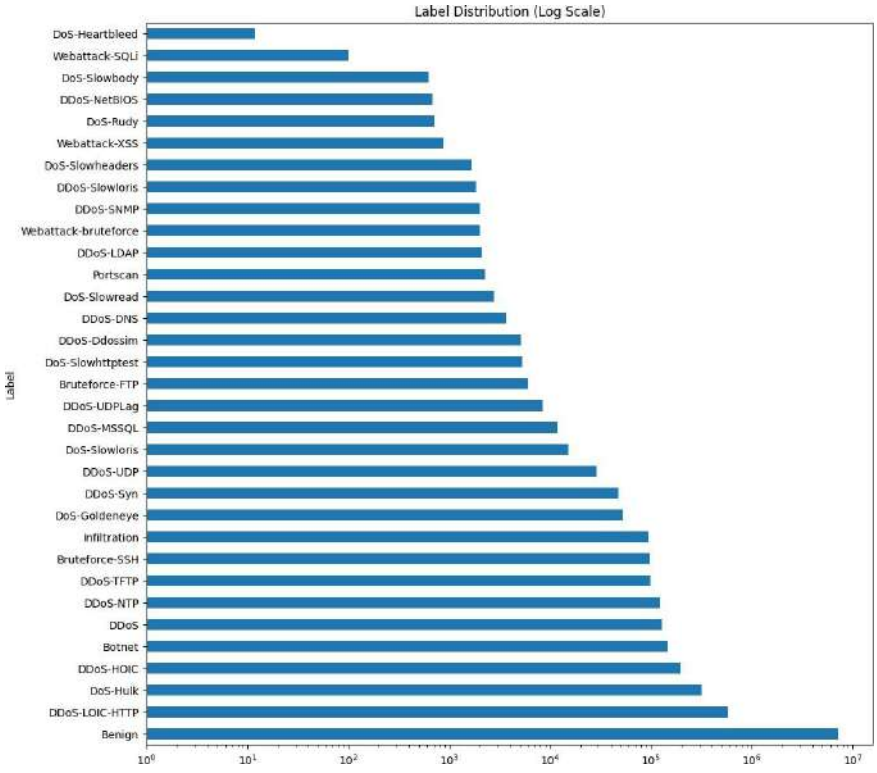


Figure 4.5.2 Bar chart of Label plotted on log scale

4.6 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: -

1. Init Fwd Win Bytes: 29%
2. 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.

4.7 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 isMaliious=1 : 1981390

4.8 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

4.9 Handling large size of the 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.

4.10 Analyzing 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 Equation (18) is satisfied, then x is an outlier.

Table 4.10.1 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

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: -

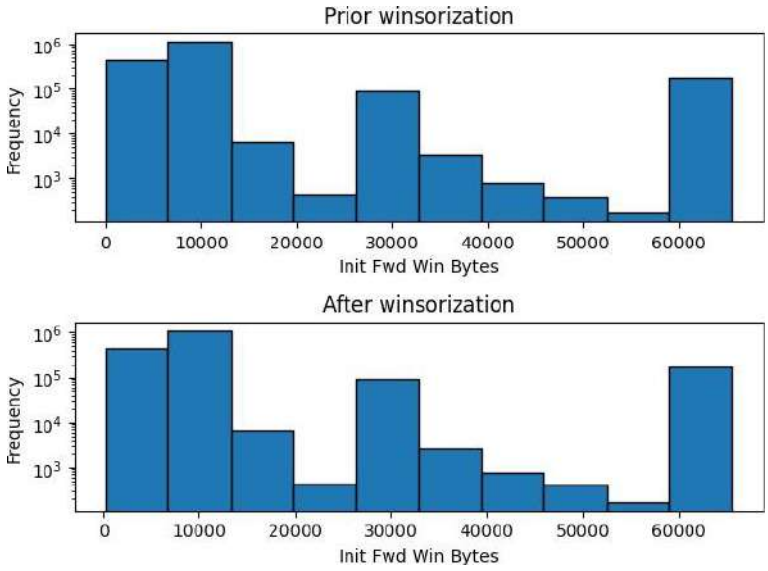


Figure4.10.1Histogram to compare impact of winsorization on InitFwdWinBytes

- 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: -

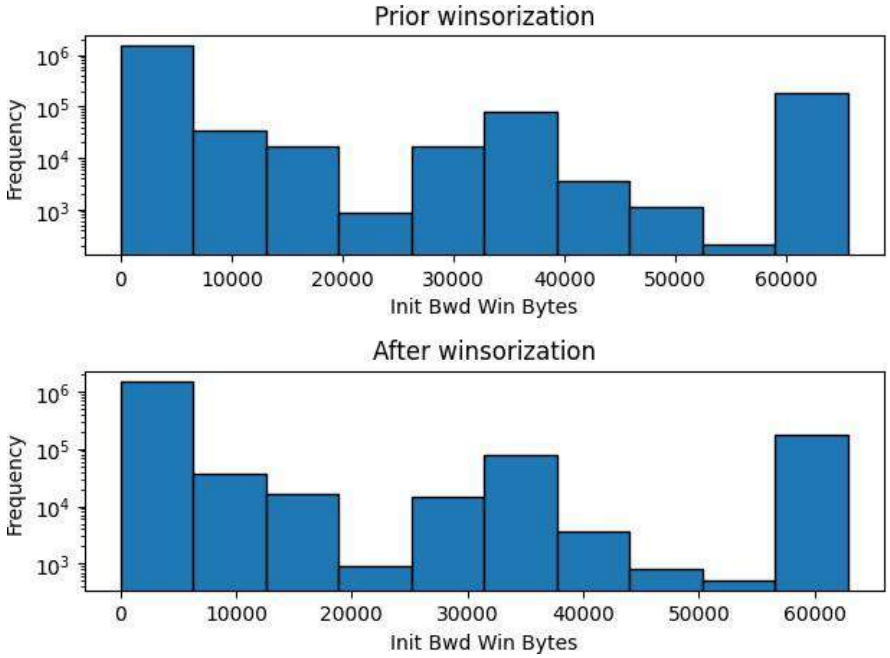


Figure 4.10.2 Histogram to compare impact of winsorization on Init Bwd Win Bytes

- 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: -

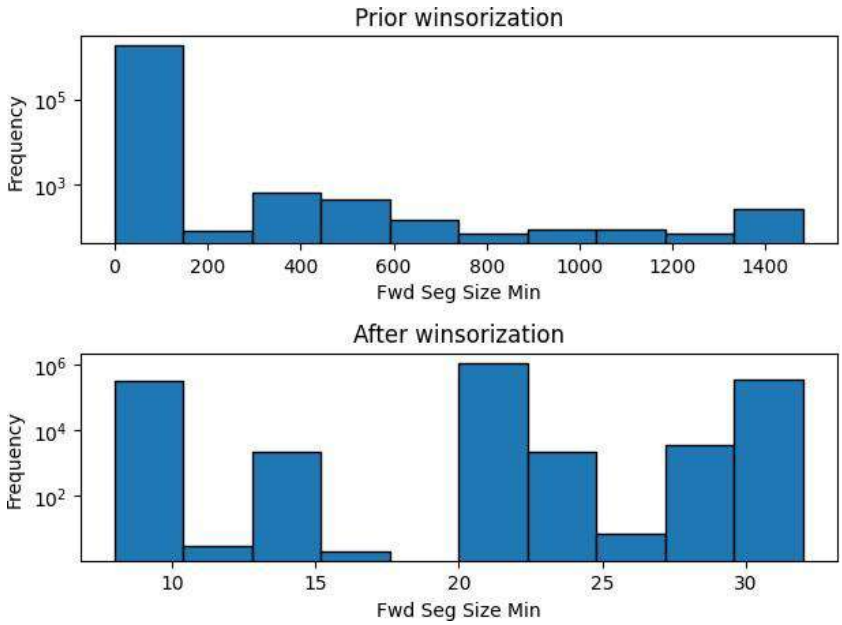


Figure 4.10.3 Histogram to compare impact of winsorization on Fwd Seg Size Min

- 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: -

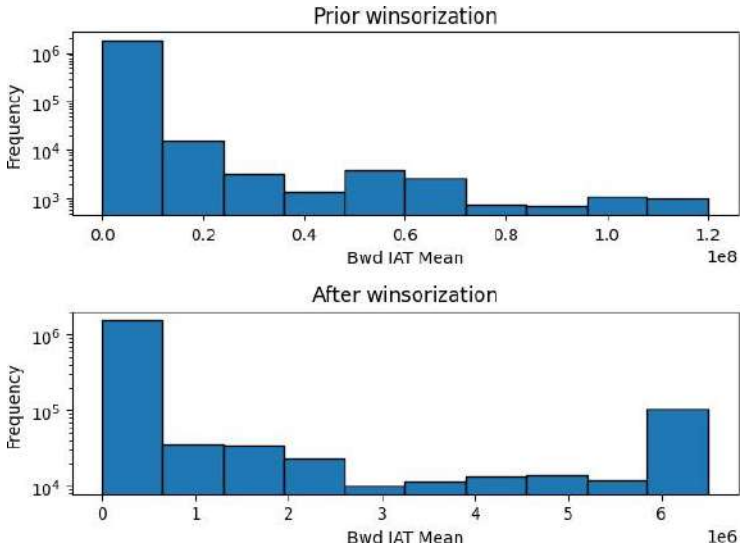


Figure 4.10.4Histogram to compare impact of winsorization on Bwd IATMean

- 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 is computed using Equation (1).

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: -

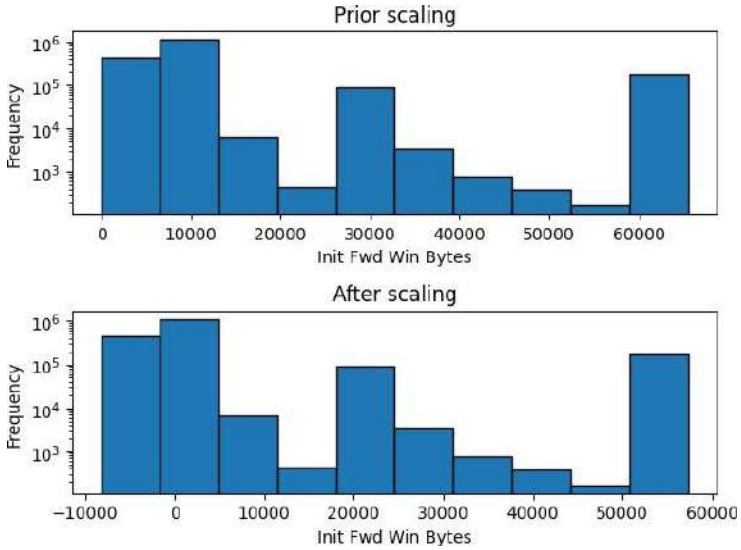


Figure 4.10.5 Histogram to compare impact of Robust Scaling on Init Fwd Win Bytes

Init Bwd Win Bytes: -

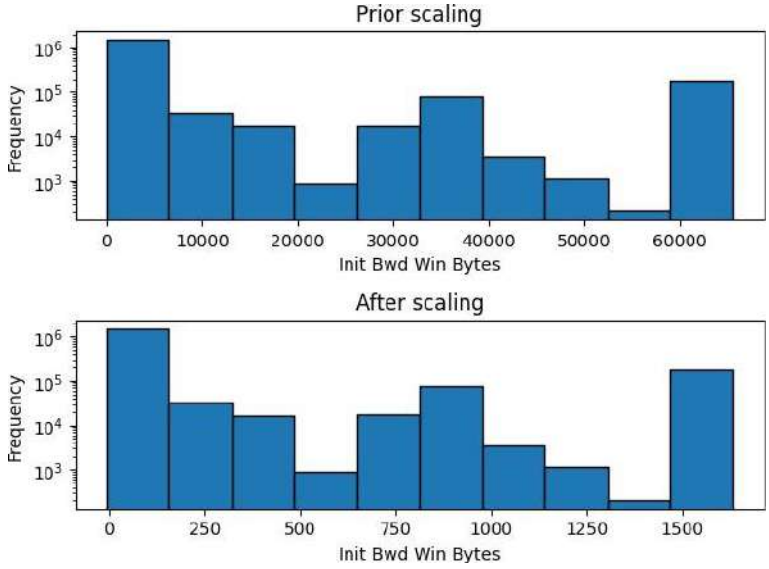


Figure 4.10.6 Histogram to compare impact of Robust Scaling on Init Bwd Win Bytes

Fwd Seg Size Min: -

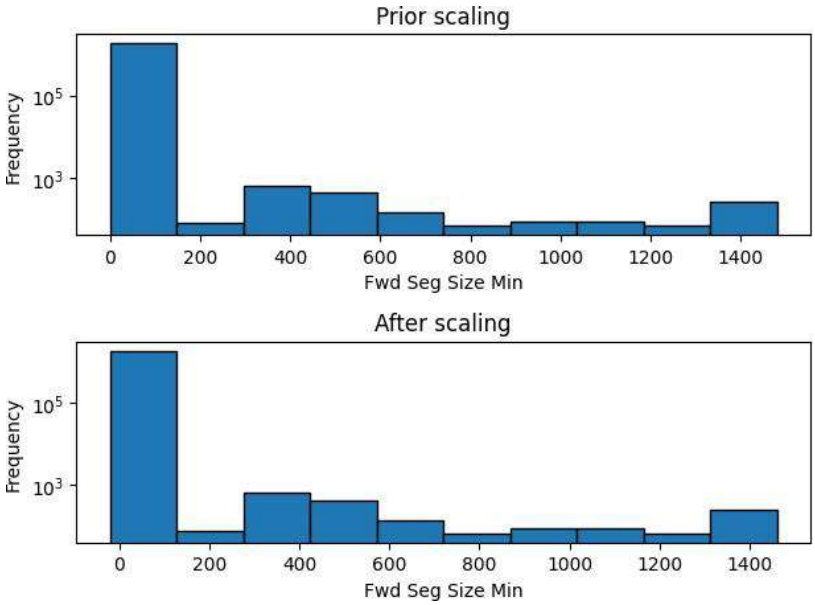


Figure 4.10.7 Histogram to compare impact of Robust Scaling on Fwd Seg Size Min

Bwd IAT Mean: -

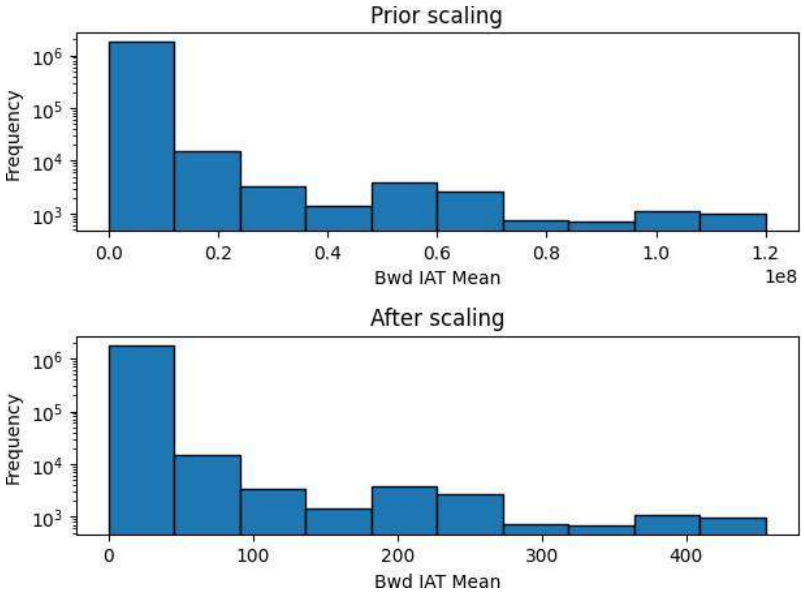


Figure 4.10.8 Histogram to compare impact of Robust Scaling on Bwd IAT Mean

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.

4.11 Distribution of data plotted on log scale for each independent feature after handling negative values and outliers: -

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.

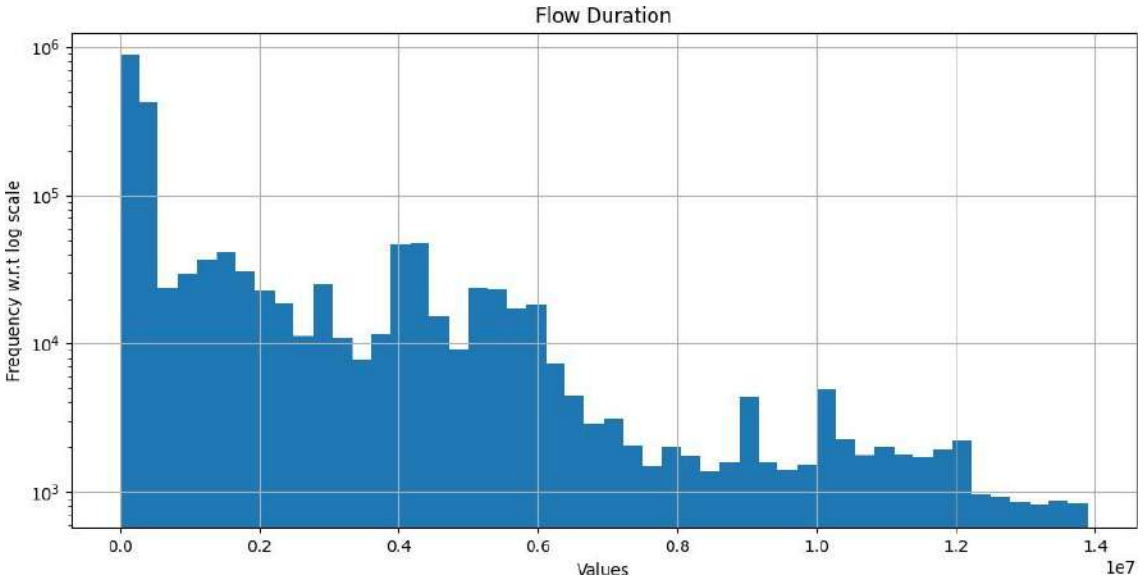


Figure 4.11.1 Histogram of Flow Duration plotted on log scale after handling negative values and outliers

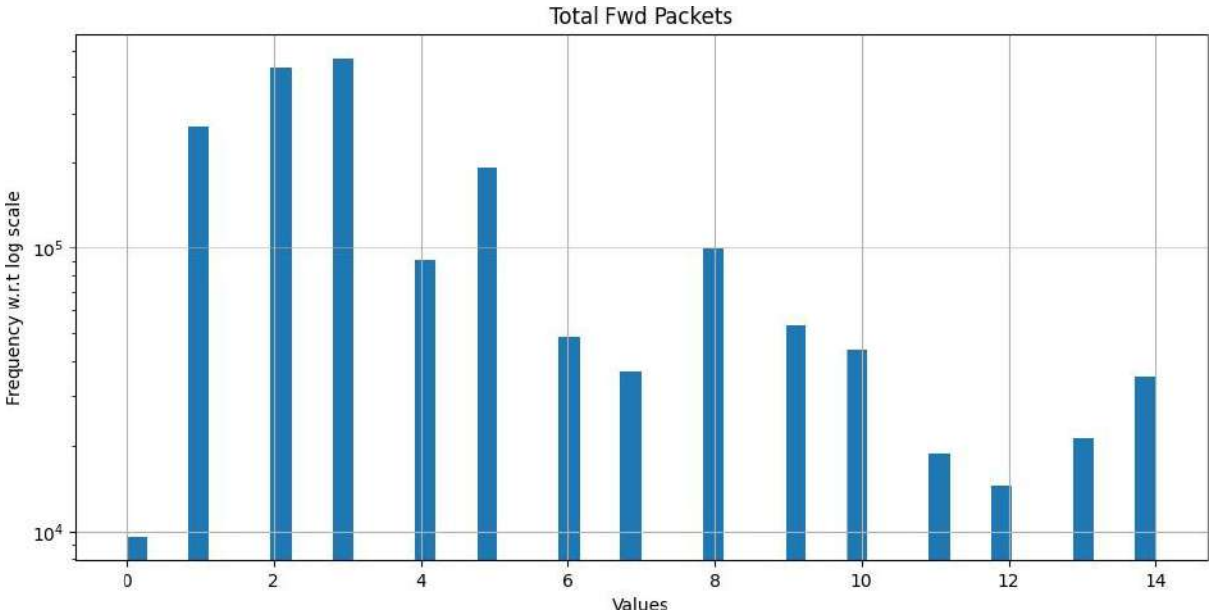


Figure 4.11.2 Histogram of Total Fwd Packets plotted on log scale after handling negative values and outliers

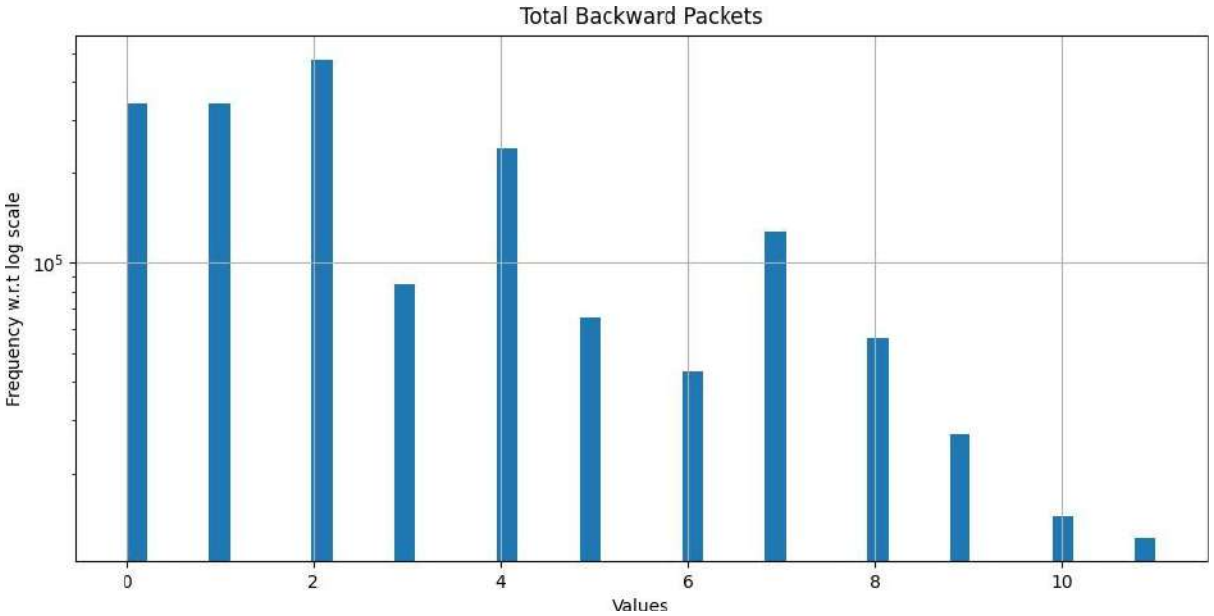


Figure 4.11.3 Histogram of Total Backward Packets plotted on log scale after handling negative values and outliers

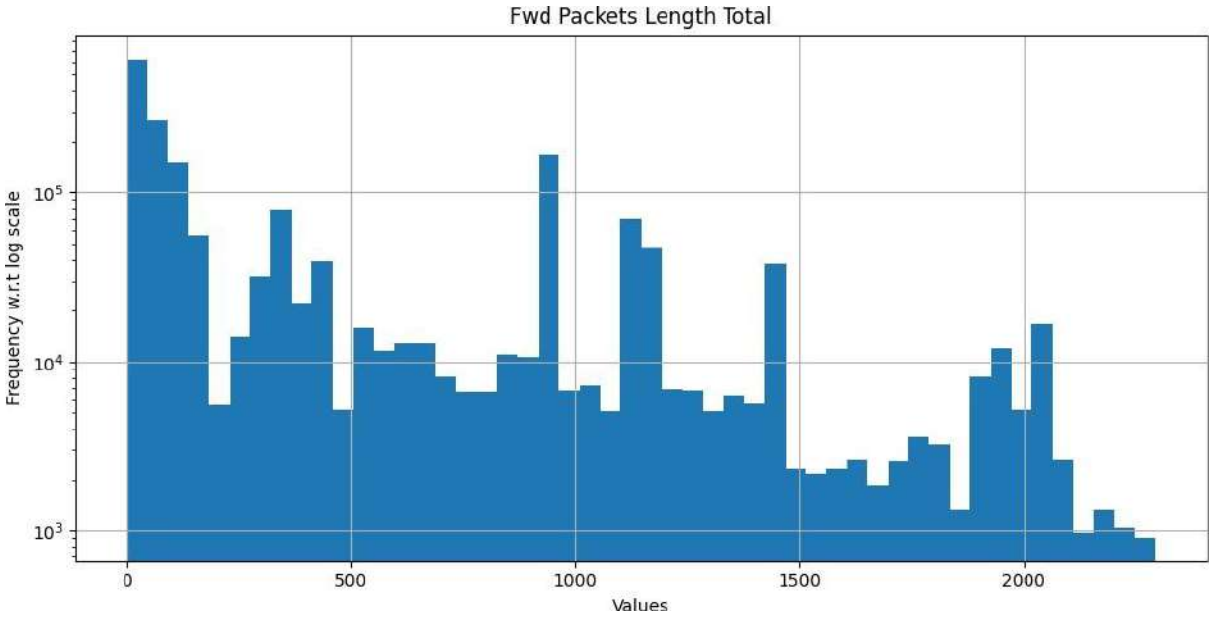


Figure 4.11.4 Histogram of Fwd Packets Length Total plotted on log scale after handling negative values and outliers

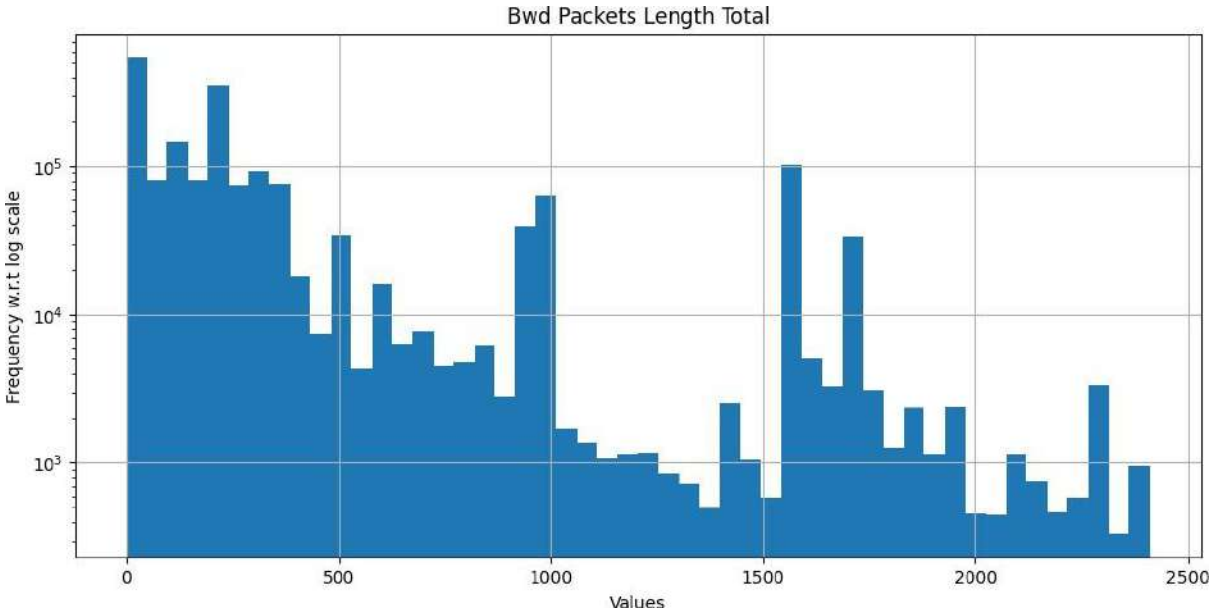


Figure 4.11.5 Histogram of Bwd Packets Length Total plotted on log scale after handling negative values and outliers

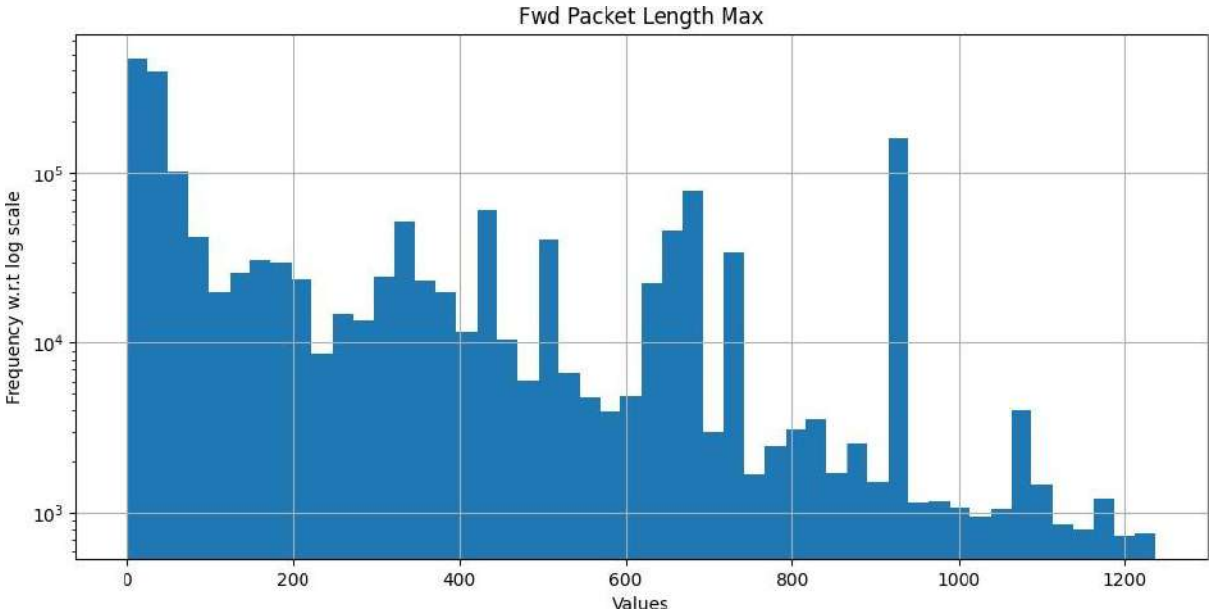


Figure 4.11.6 Histogram of Fwd Packet Length Max plotted on log scale after handling negative values and outliers

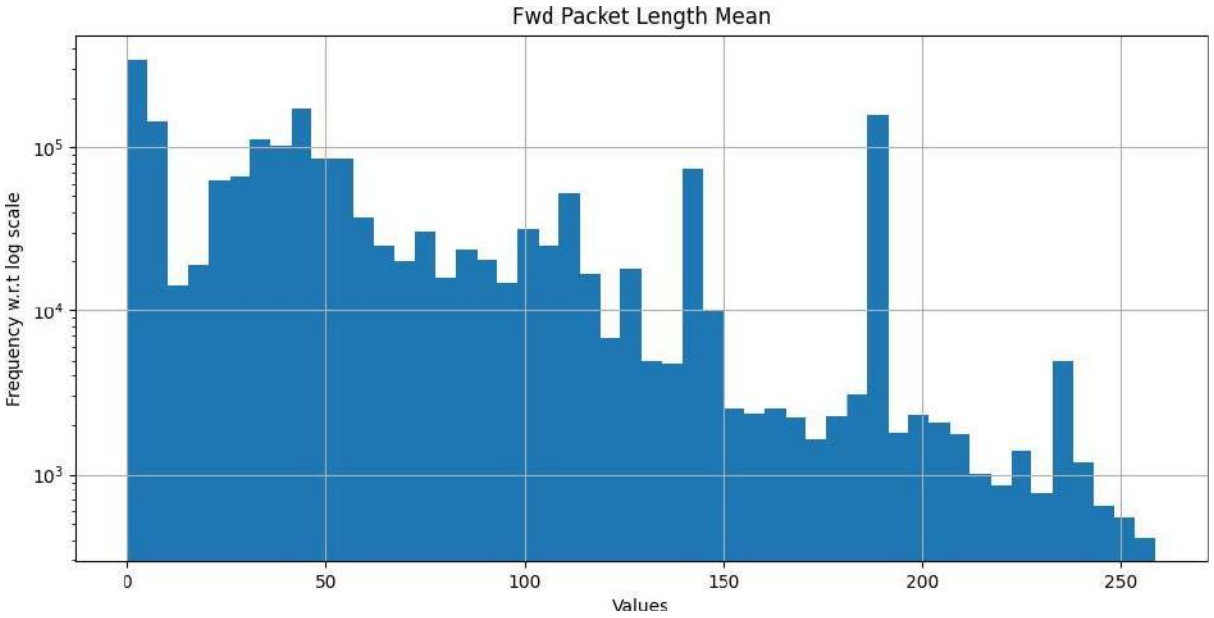


Figure 4.11.7 Histogram of Fwd Packet Length Mean plotted on log scale after handling negative values and outliers

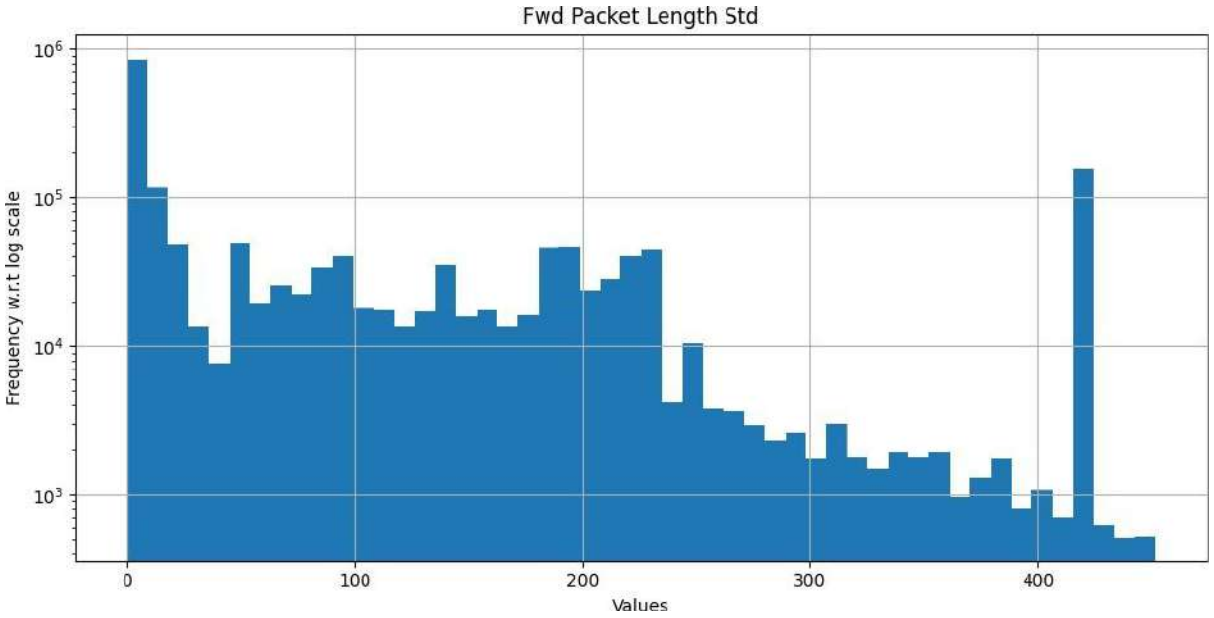


Figure 4.11.8 Histogram of Fwd Packet Length Std plotted on log scale after handling negative values and outliers

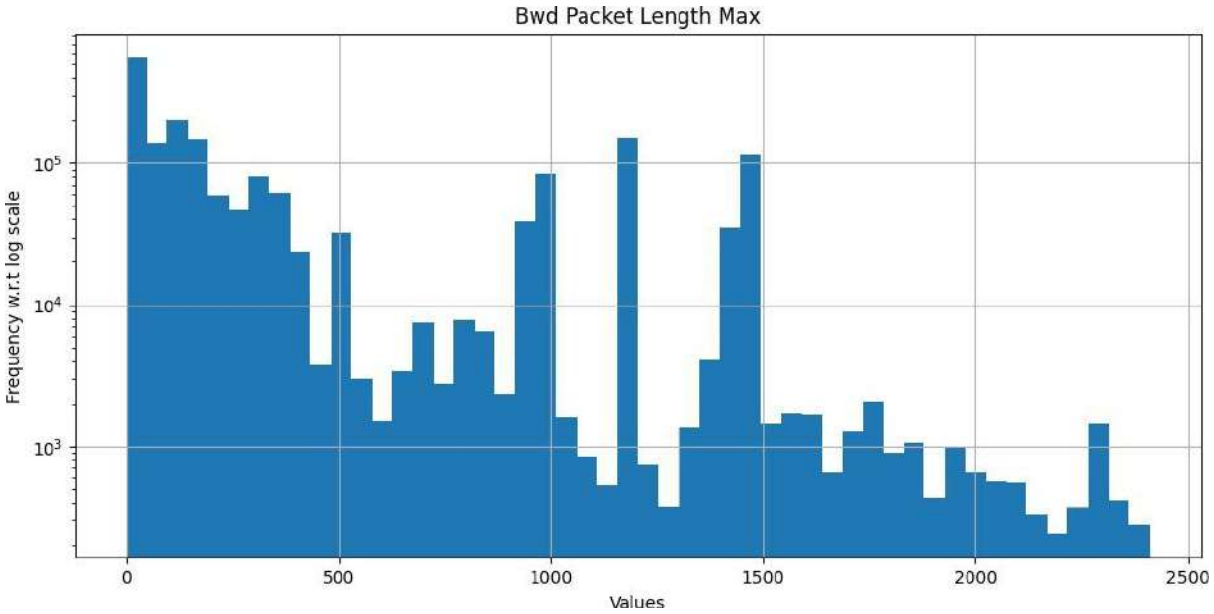


Figure 4.11.9 Histogram of Bwd Packet Length Max plotted on log scale after handling negative values and outliers

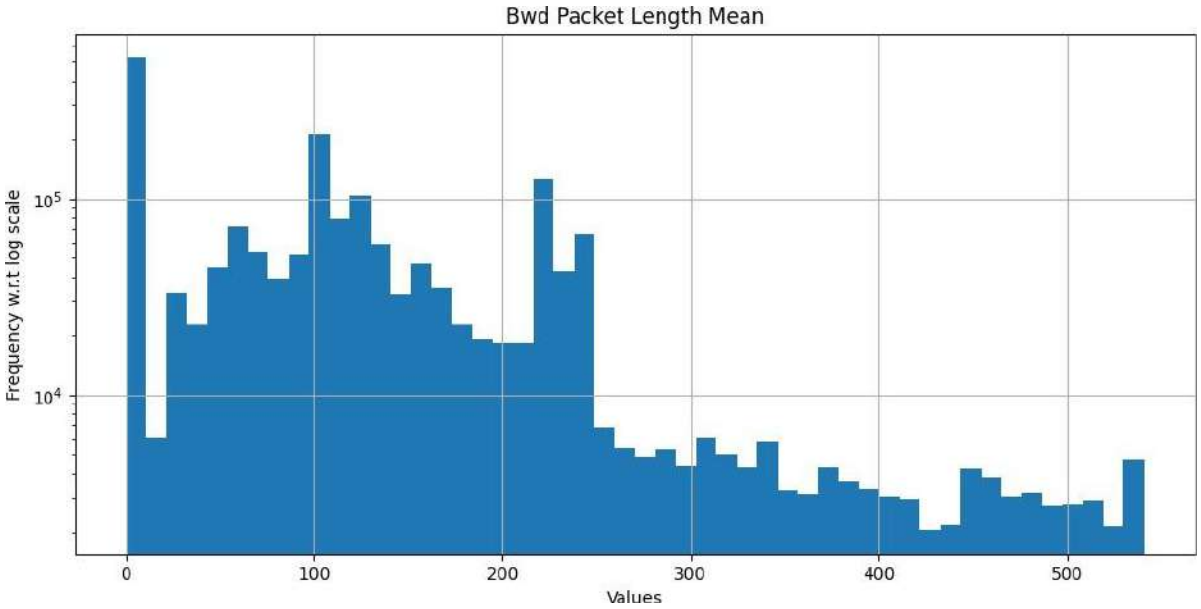


Figure 4.11.10 Histogram of Bwd Packet Length Mean plotted on log scale after handling negative values and outliers

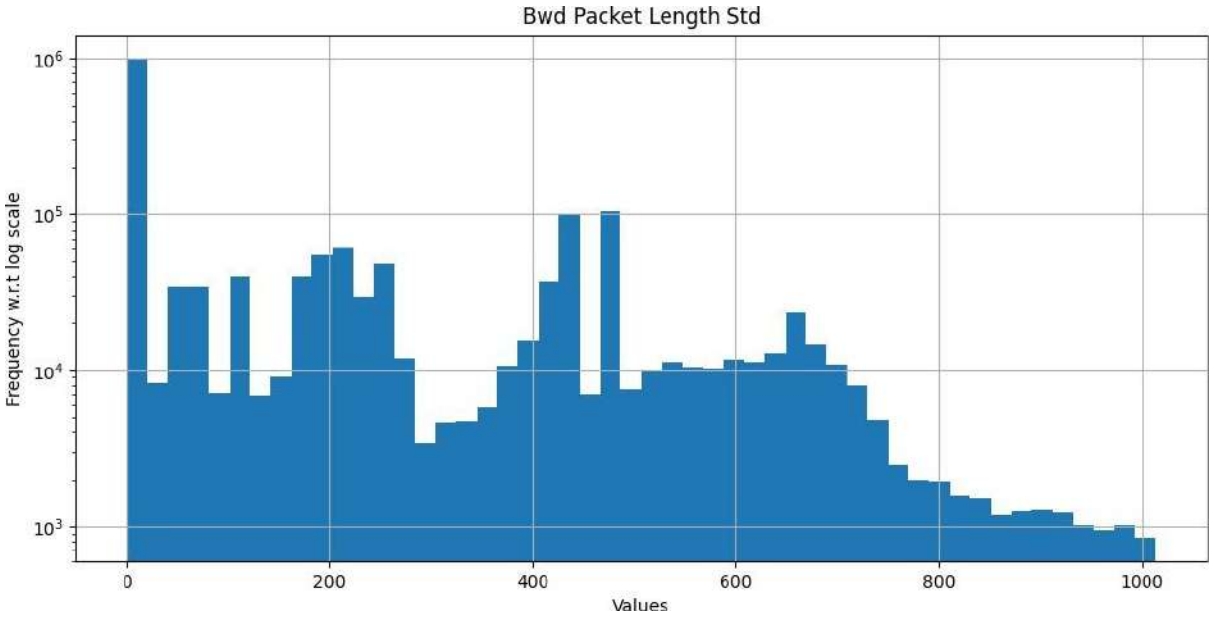


Figure 4.11.11 Histogram of Bwd Packet Length Std plotted on log scale after handling negative values and outliers

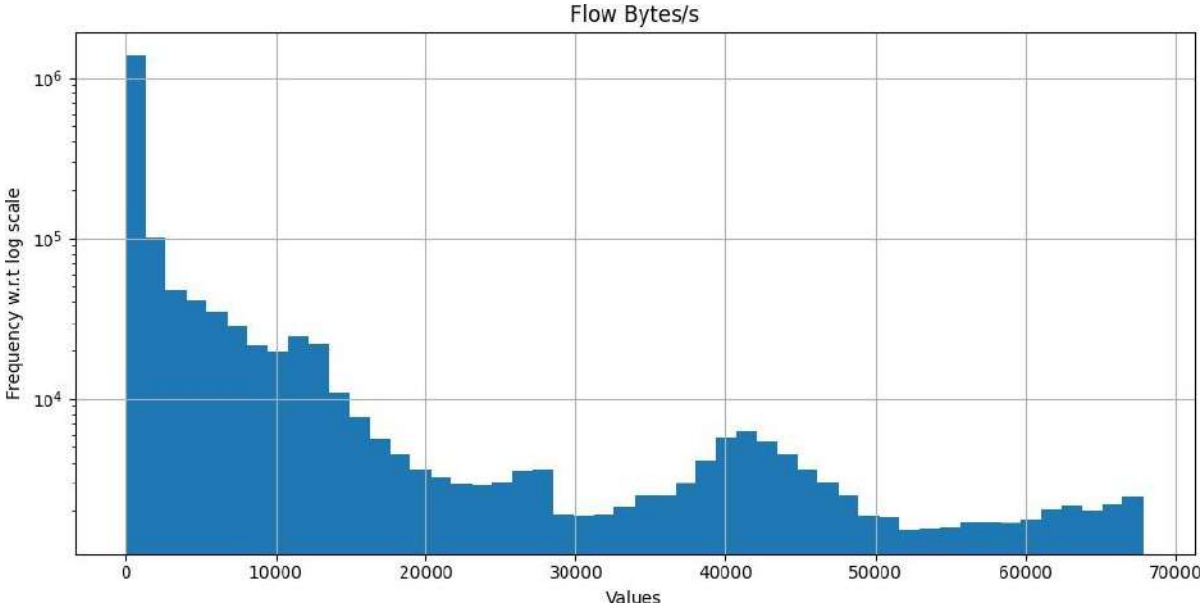


Figure 4.11.12 Histogram of Flow Bytes/s plotted on log scale after handling negative values and outliers

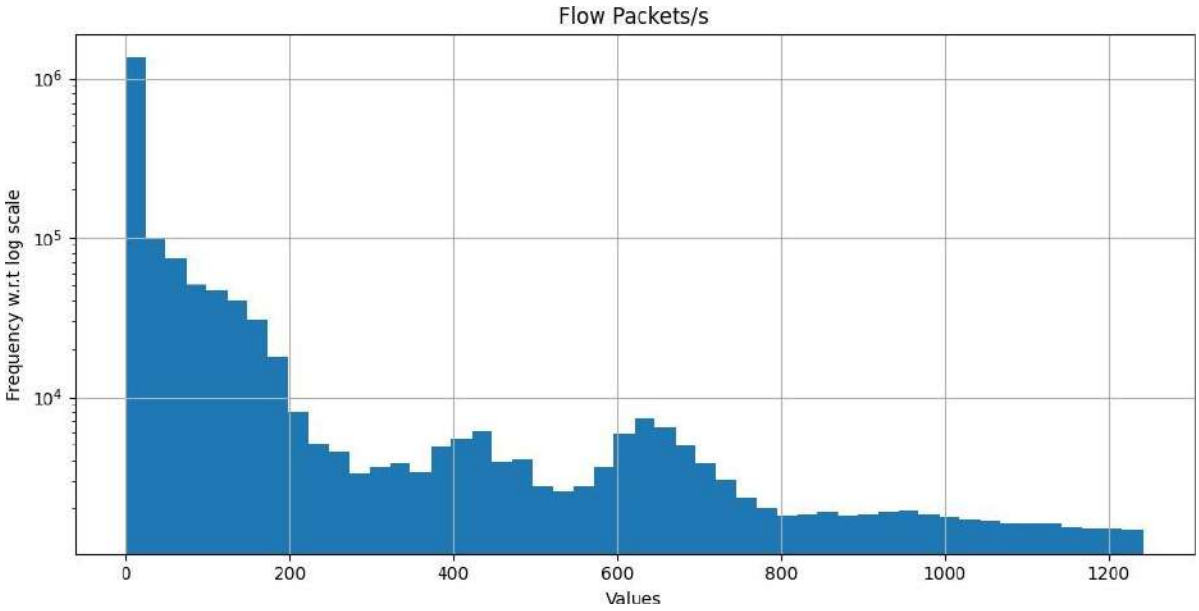


Figure 4.11.13 Histogram of Flow Packets/s plotted on log scale after handling negative values and outliers

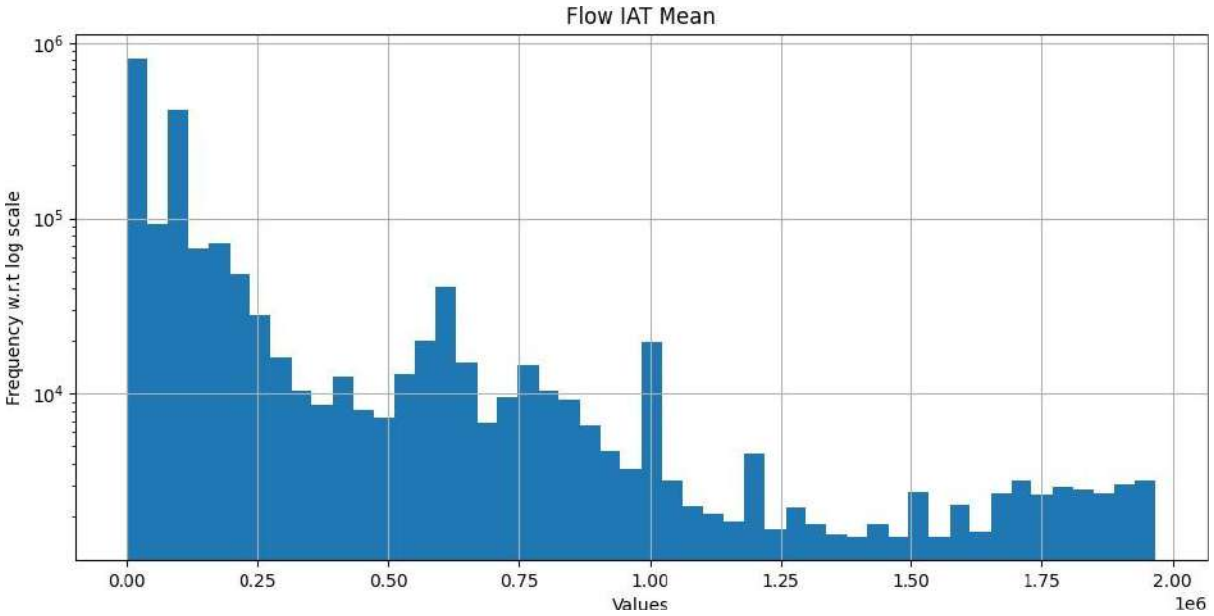


Figure 4.11.14 Histogram of Flow IAT Mean plotted on log scale after handling negative values and outliers

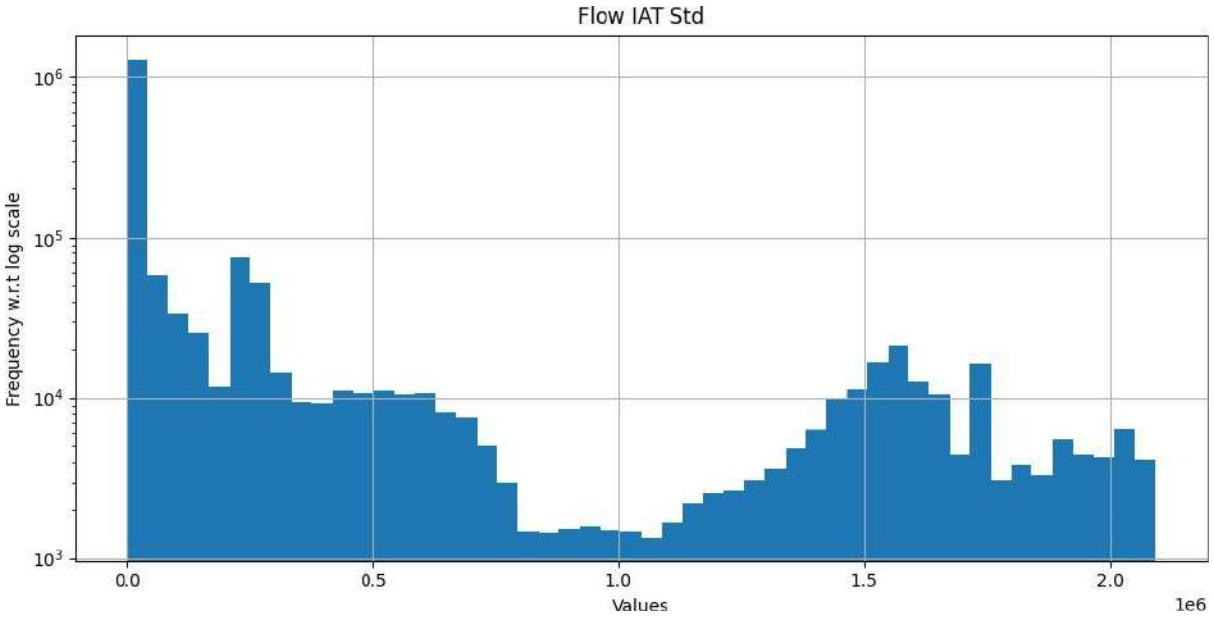


Figure 4.11.15 Histogram of Flow IAT Std plotted on log scale after handling negative values and outliers

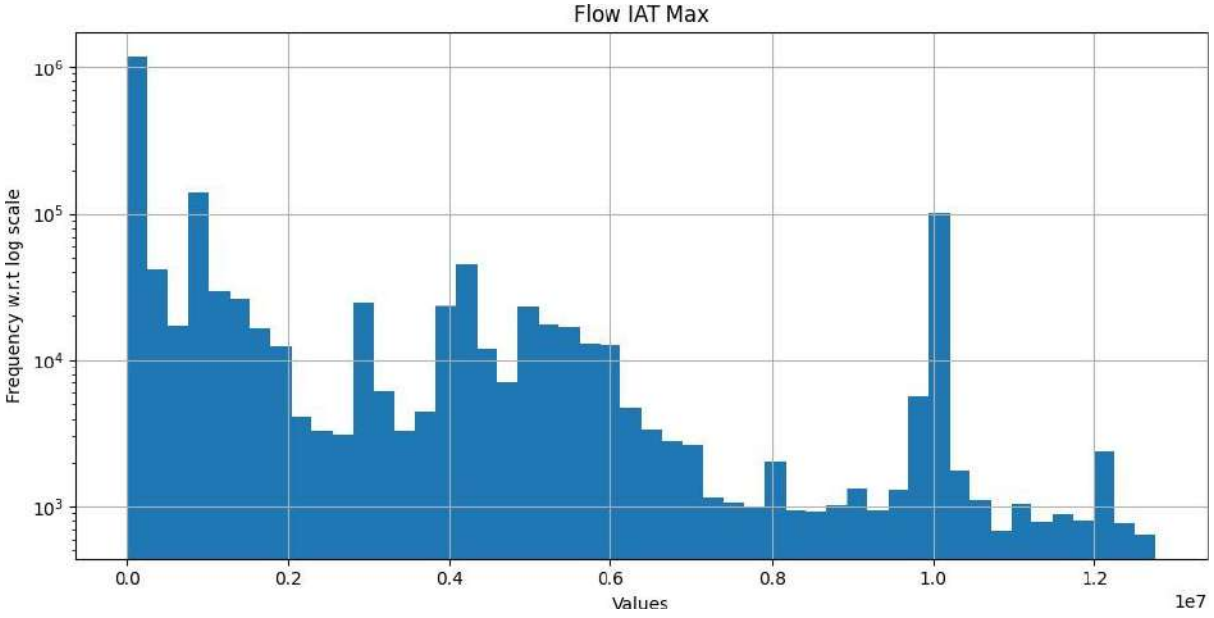


Figure 4.11.16 Histogram of Flow IAT Max plotted on log scale after handling negative values and outliers

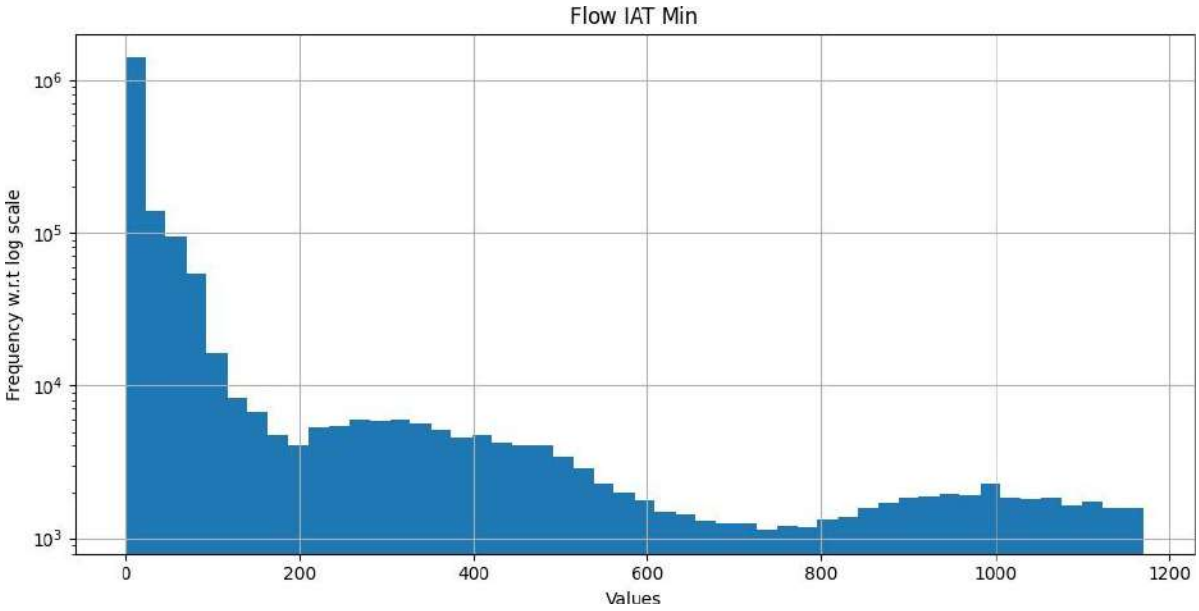


Figure 4.11.17 Histogram of Flow IAT Min plotted on log scale after handling negative values and outliers

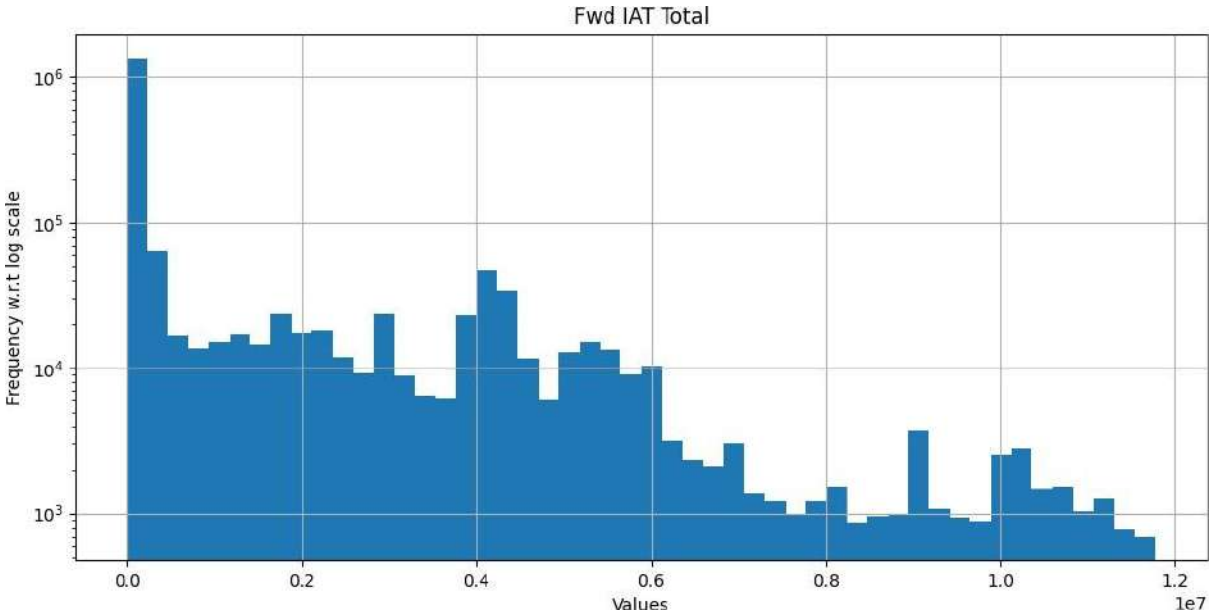


Figure 4.11.18 Histogram of Fwd IAT Total plotted on log scale after handling negative values and outliers

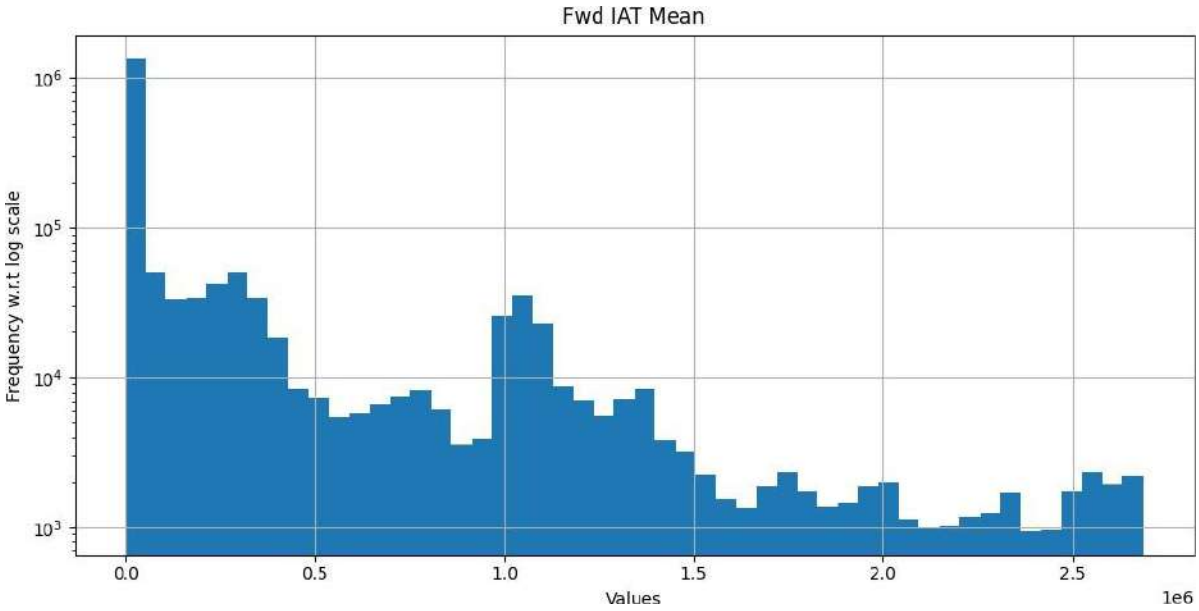


Figure 4.11.19 Histogram of Fwd IAT Mean plotted on log scale after handling negative values and outliers

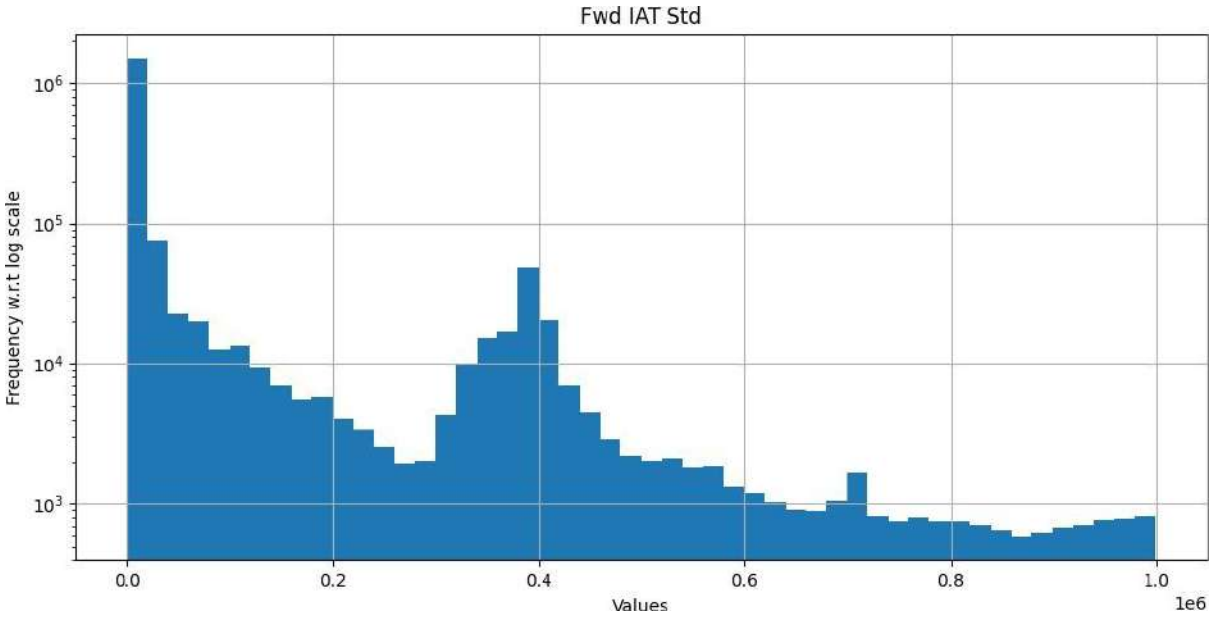


Figure 4.11.20 Histogram of Fwd IAT Std plotted on log scale after handling negative values and outliers

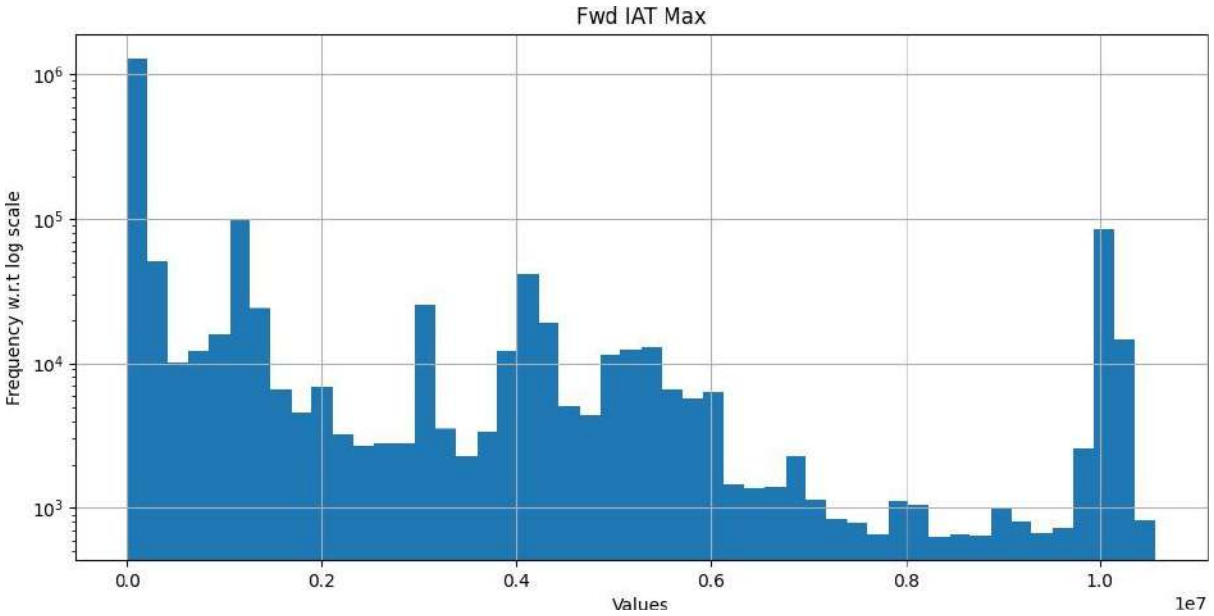


Figure 4.11.21 Histogram of Fwd IAT Max plotted on log scale after handling negative values and outliers

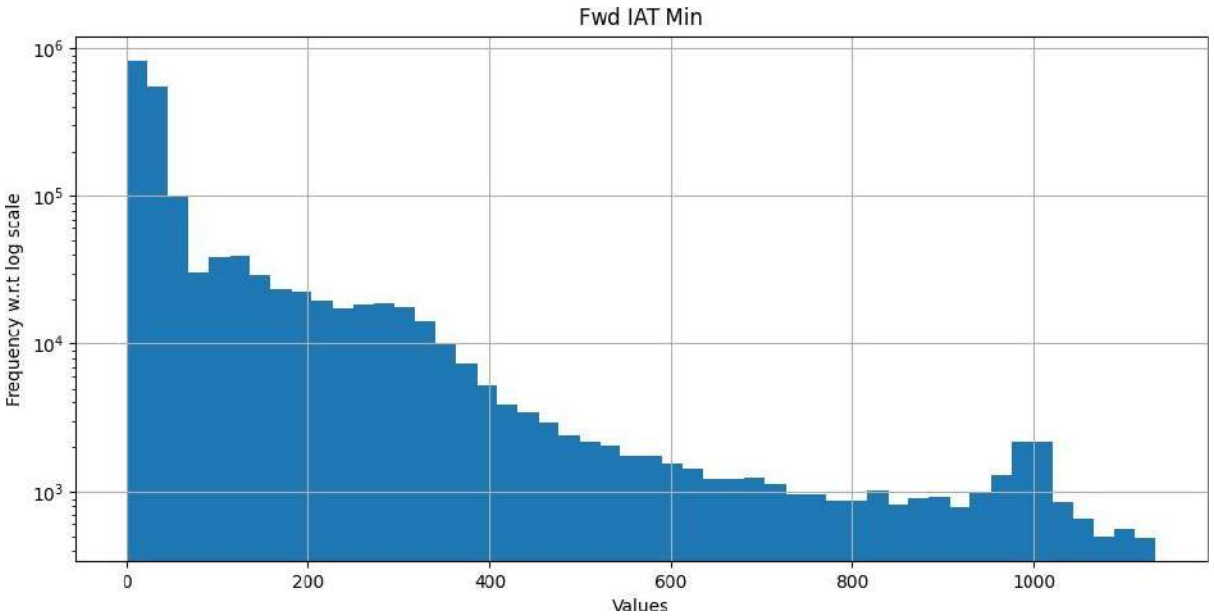


Figure 4.11.22 Histogram of Fwd IAT Min plotted on log scale after handling negative values and outliers

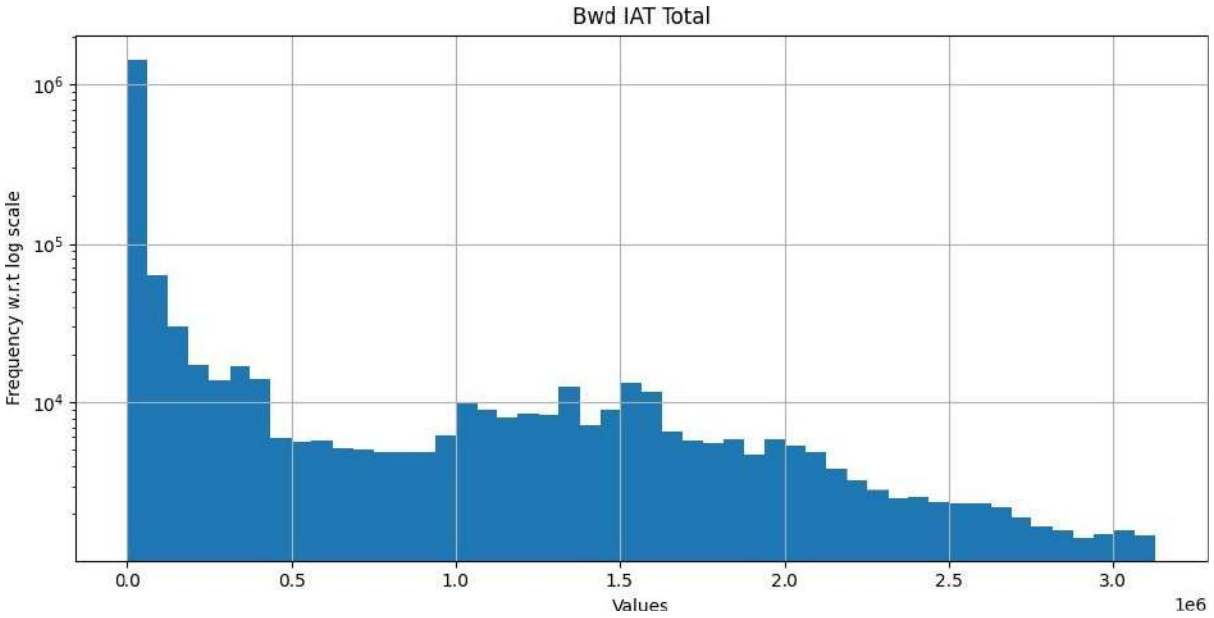


Figure 4.11.23 Histogram of Bwd IAT Total plotted on log scale after handling negative values and outliers

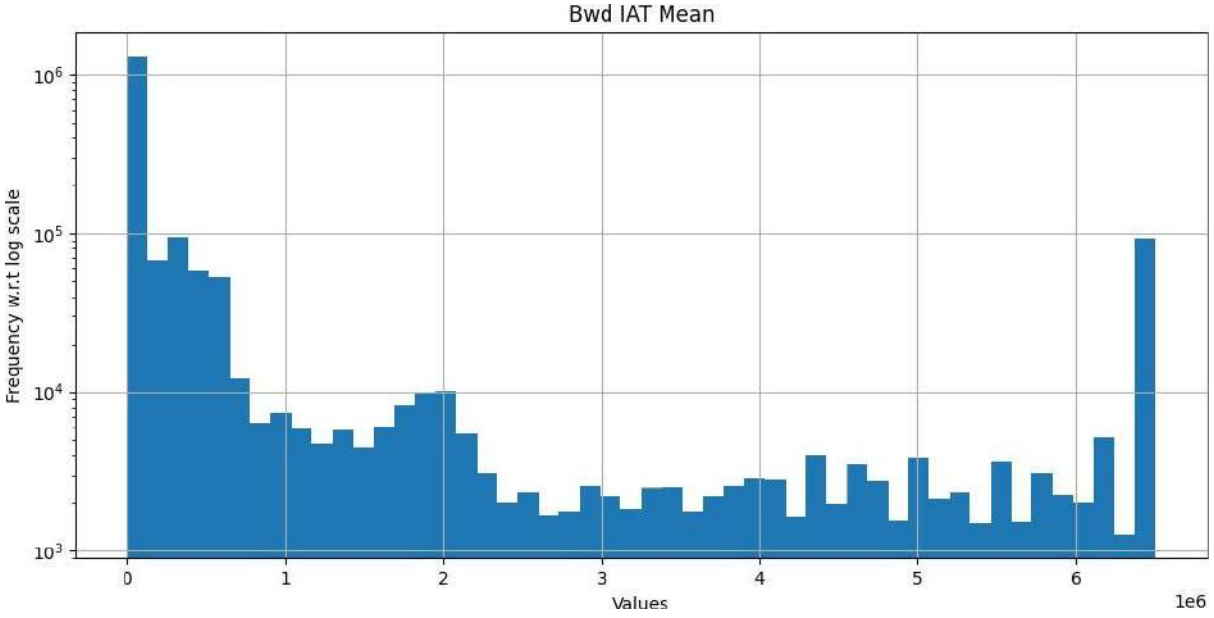


Figure 4.11.24 Histogram of Bwd IAT Mean plotted on log scale after handling negative values and outliers

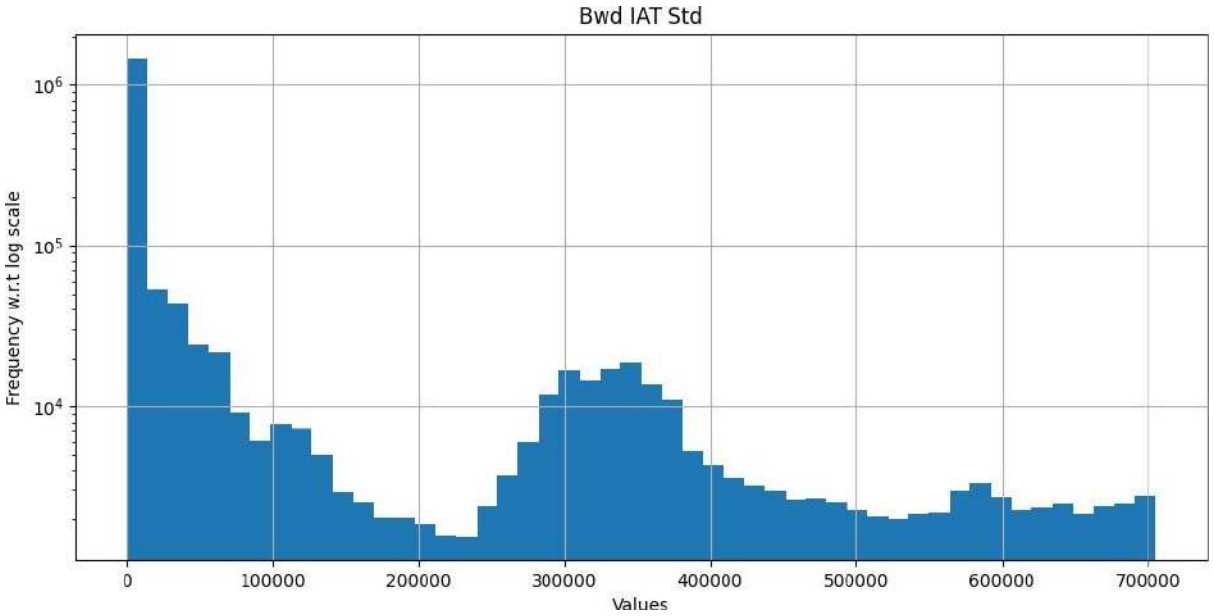


Figure 4.11.25 Histogram of Bwd IAT Std plotted on log scale after handling negative values and outliers

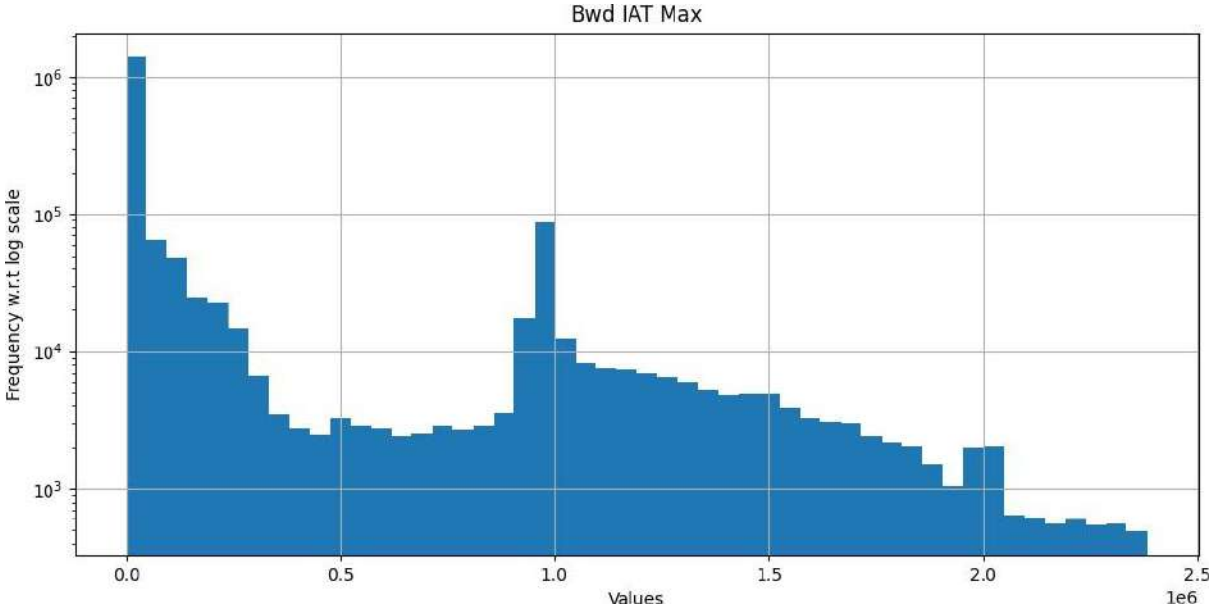


Figure 4.11.26 Histogram of Bwd IAT Max plotted on log scale after handling negative values and outliers

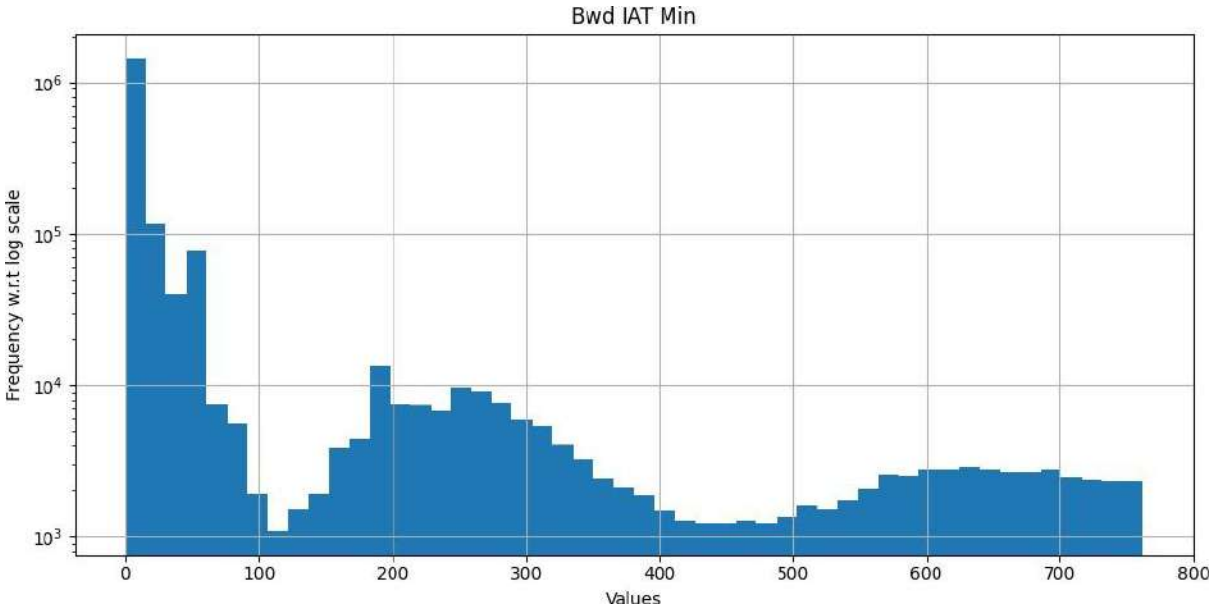


Figure 4.11.27 Histogram of Bwd IAT Min plotted on log scale after handling negative values and outliers

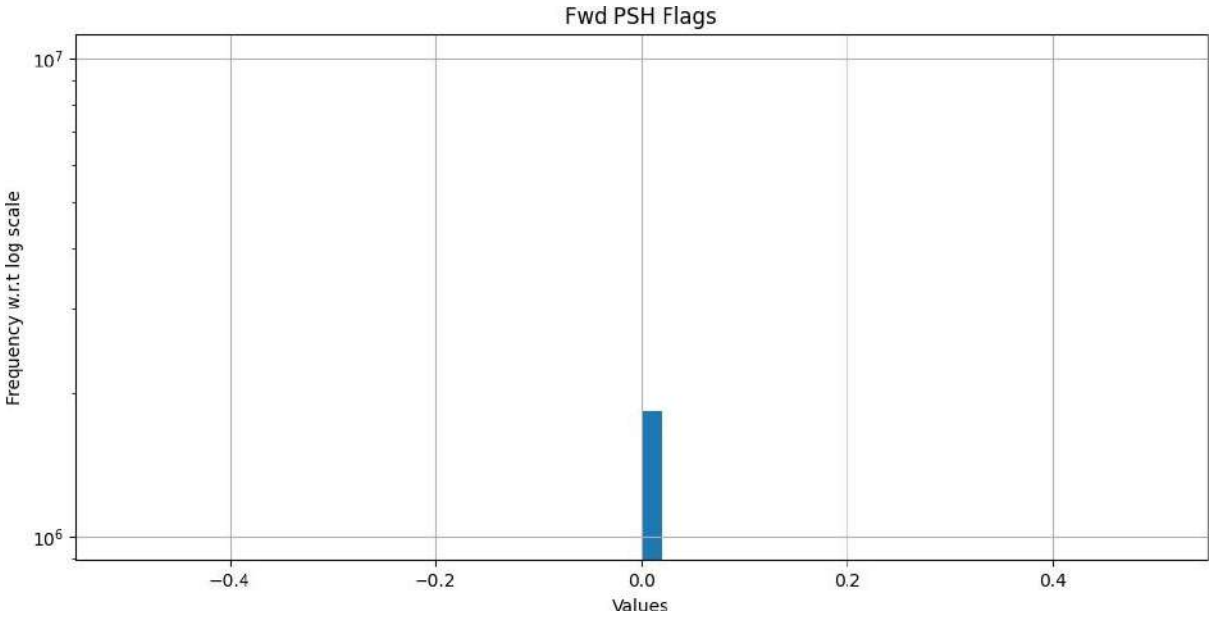


Figure 4.11.28 Histogram of Fwd PSH Flags plotted on log scale after handling negative values and outliers

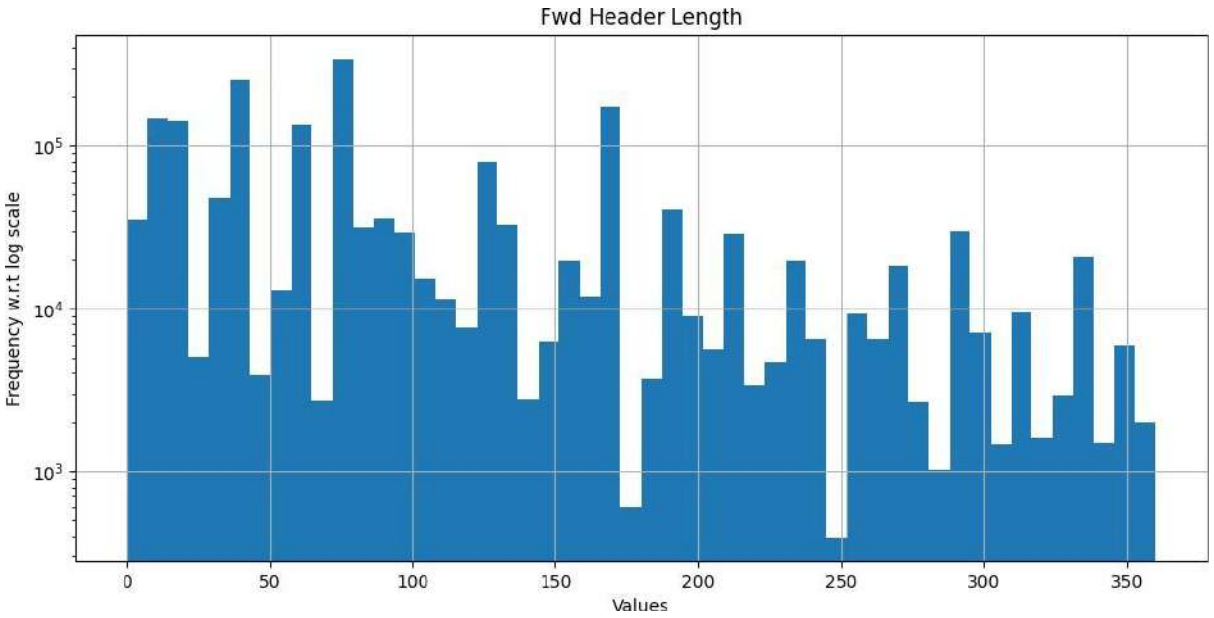


Figure 4.11.29 Histogram of Fwd Header Length plotted on log scale after handling negative values and outliers

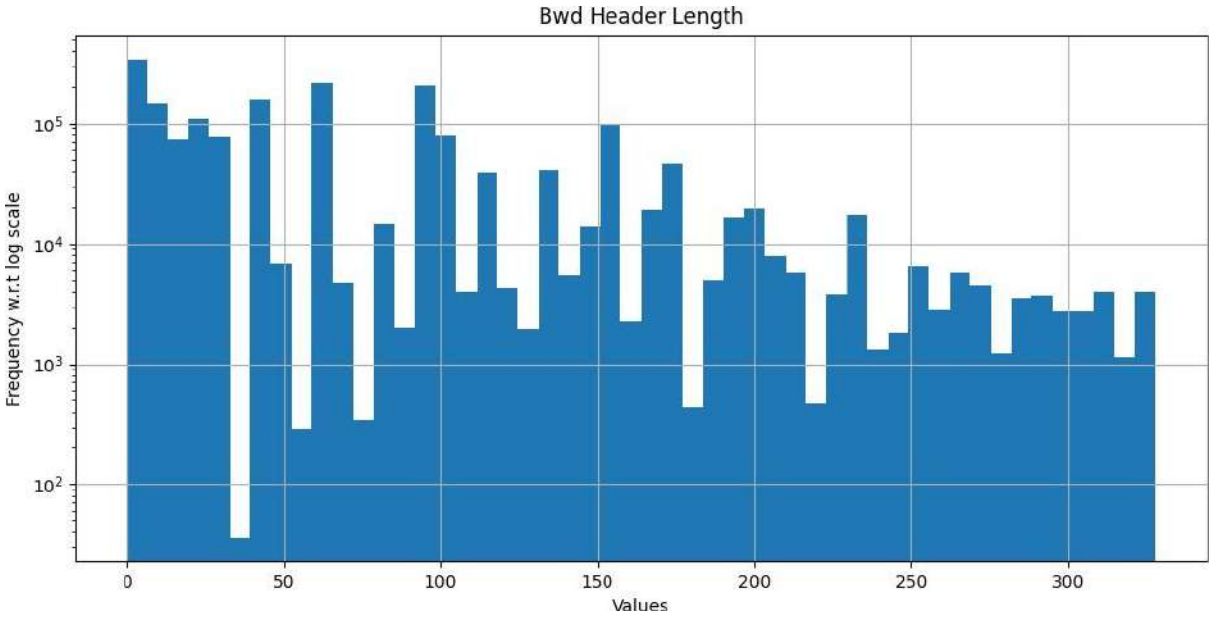


Figure 4.11.30 Histogram of Bwd Header Length plotted on log scale after handling negative values and outliers

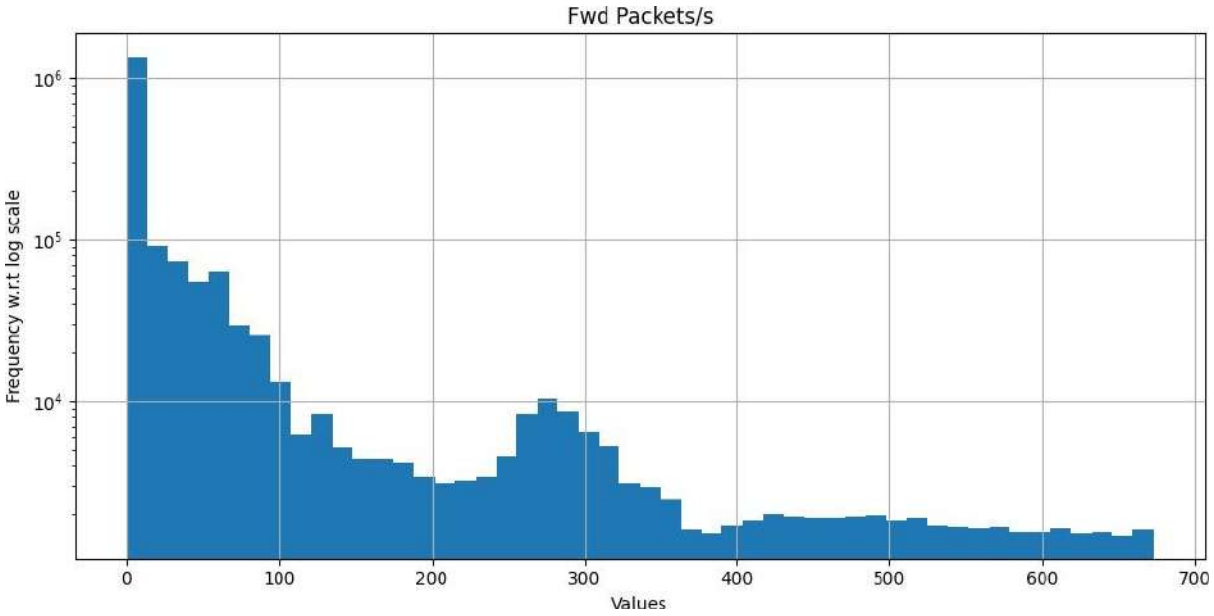


Figure 4.11.31 Histogram of Fwd Packets/s plotted on log scale after handling negative values and outliers

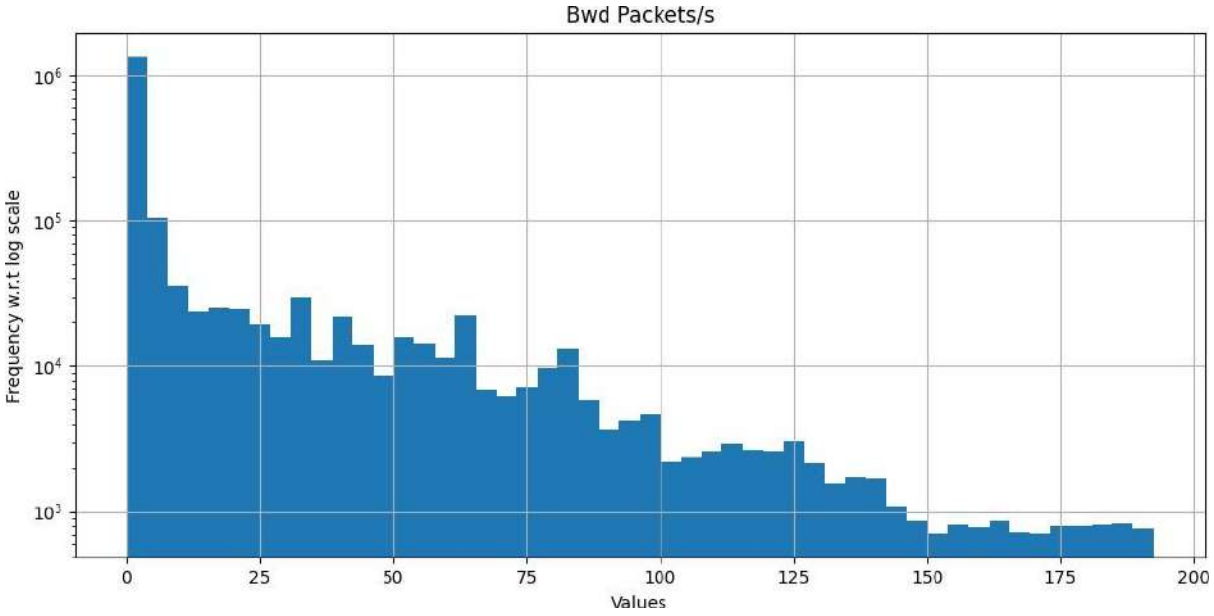


Figure 4.11.32 Histogram of Bwd Packets/s plotted on log scale after handling negative values and outliers

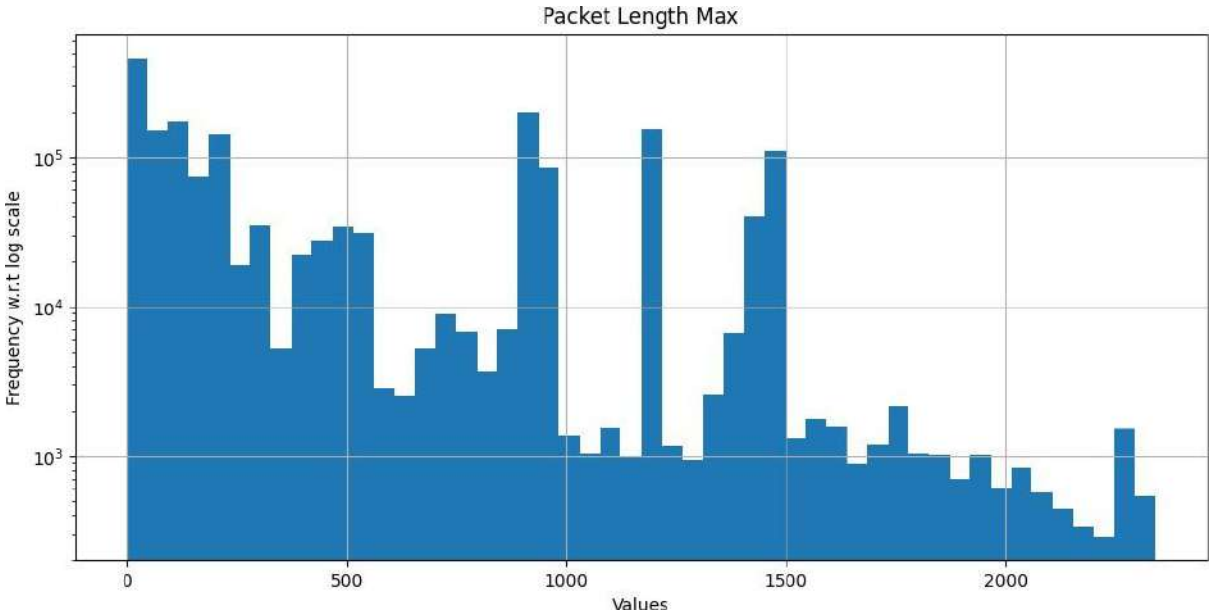


Figure 4.11.33 Histogram of Packet Length Max plotted on log scale after handling negative values and outliers

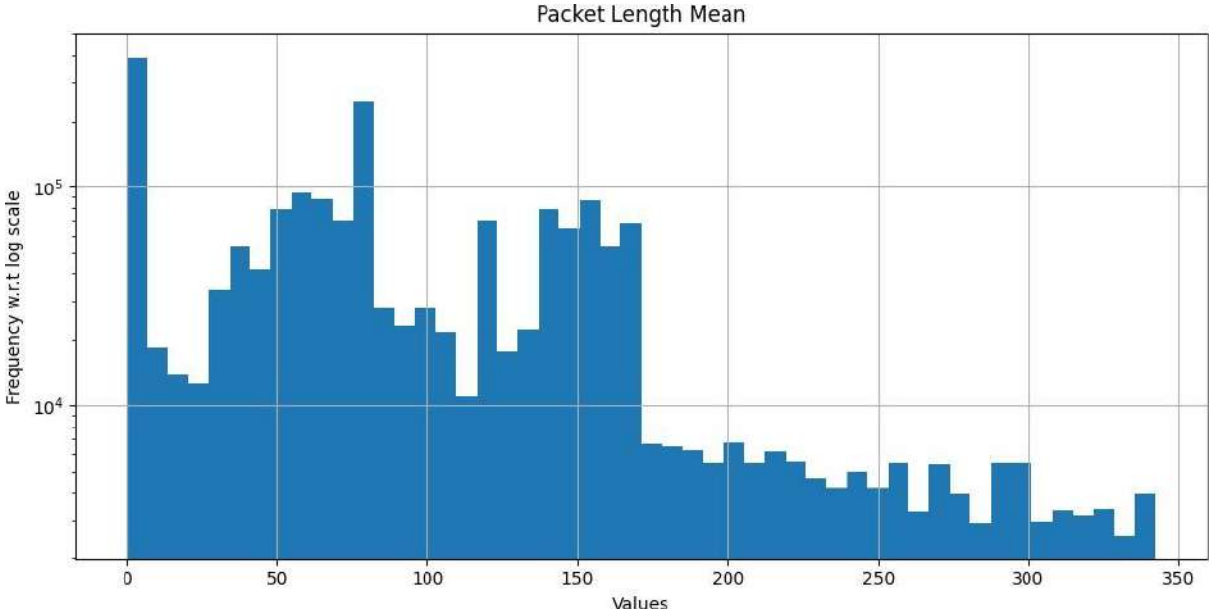


Figure 4.11.34 Histogram of Packet Length Mean plotted on log scale after handling negative values and outliers

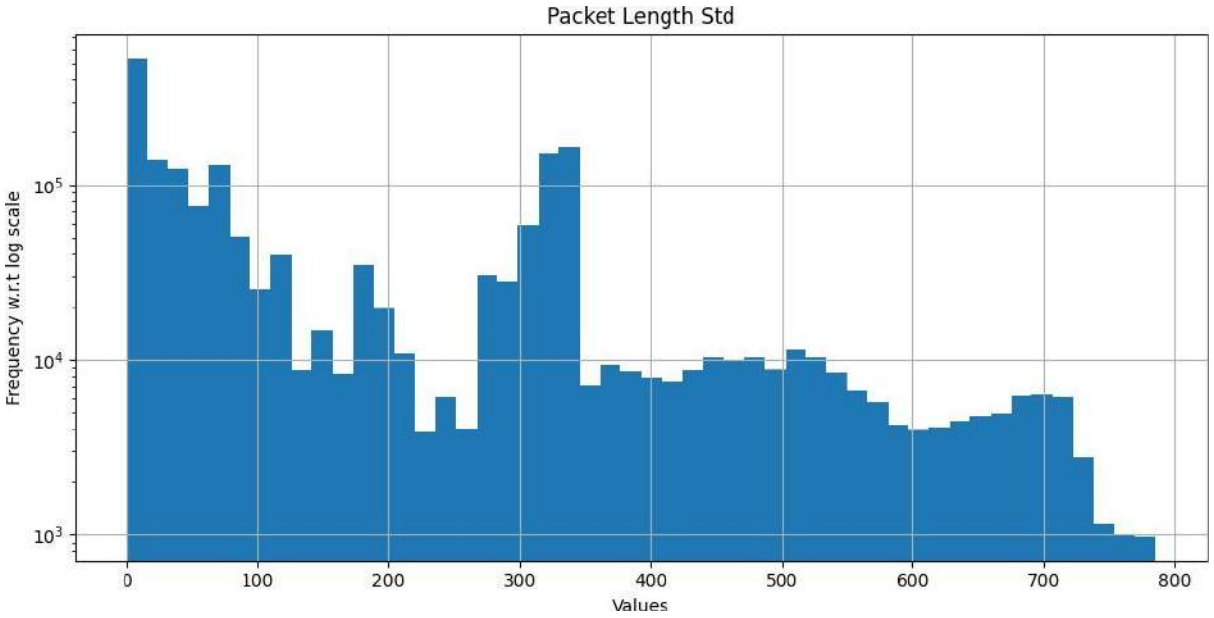


Figure 4.11.35 Histogram of Packet Length Std plotted on log scale after handling negative values and outliers

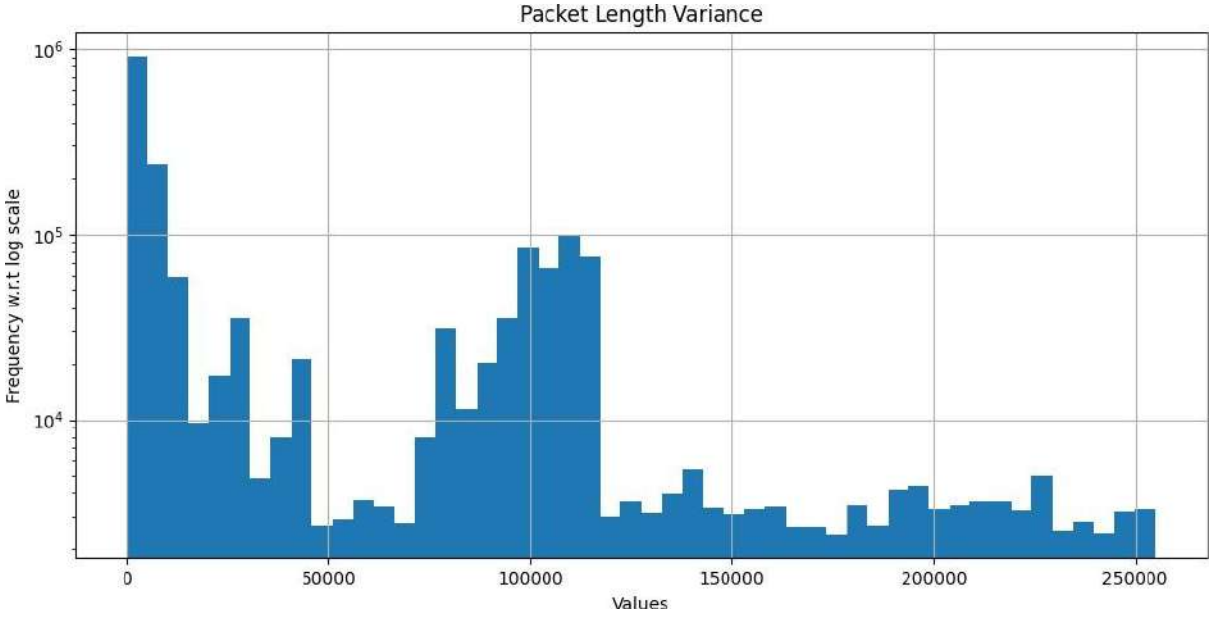


Figure 4.11.36 Histogram of Packet Length Variance plotted on log scale after handling negative values and outliers

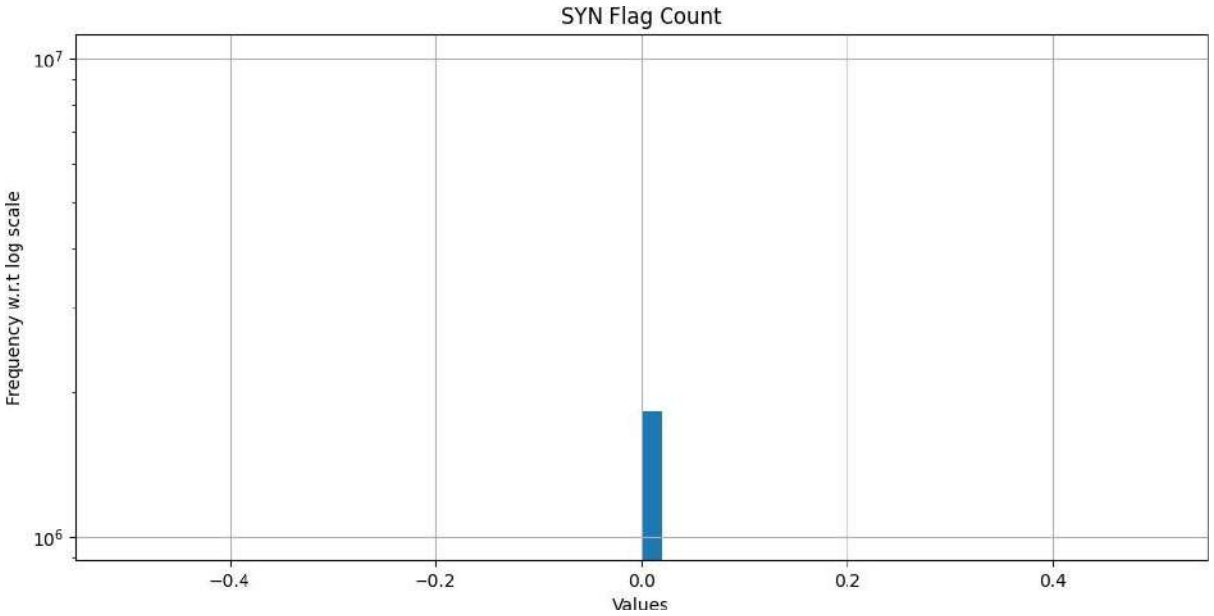


Figure 4.11.37 Histogram of SYN Flag Count plotted on log scale after handling negative values and outliers

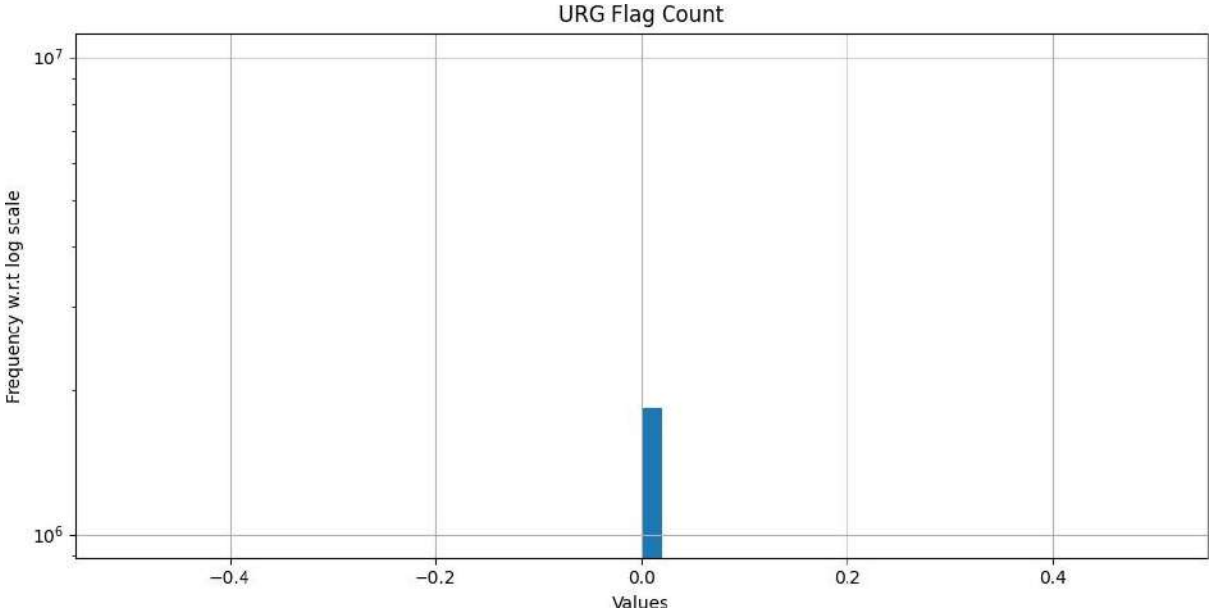


Figure 4.11.38 Histogram of URG Flag Count plotted on log scale after handling negative values and outliers

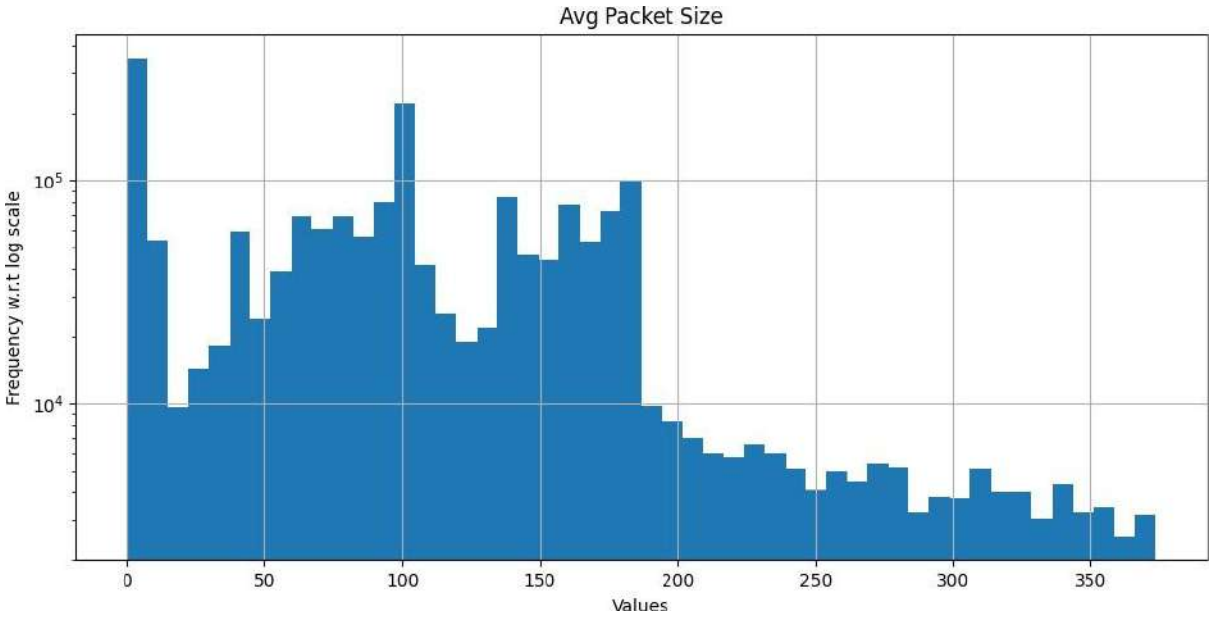


Figure 4.11.39 Histogram of Avg Packet Size plotted on log scale after handling negative values and outliers

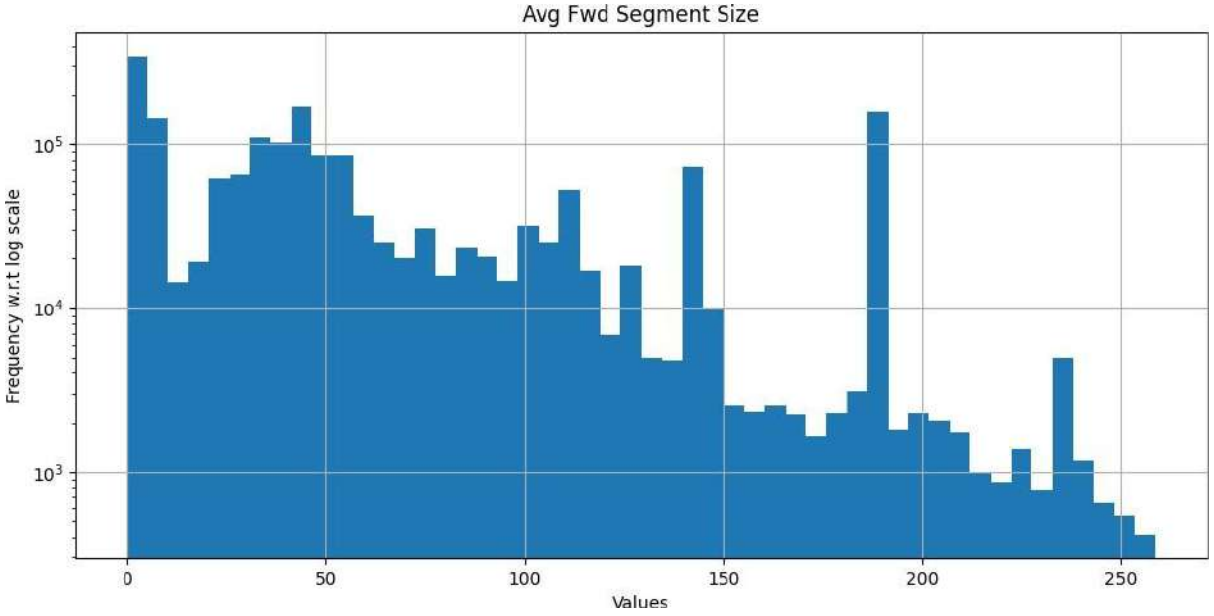


Figure 4.11.40 Histogram of Avg Fwd Segment Size plotted on log scale after handling negative values and outliers

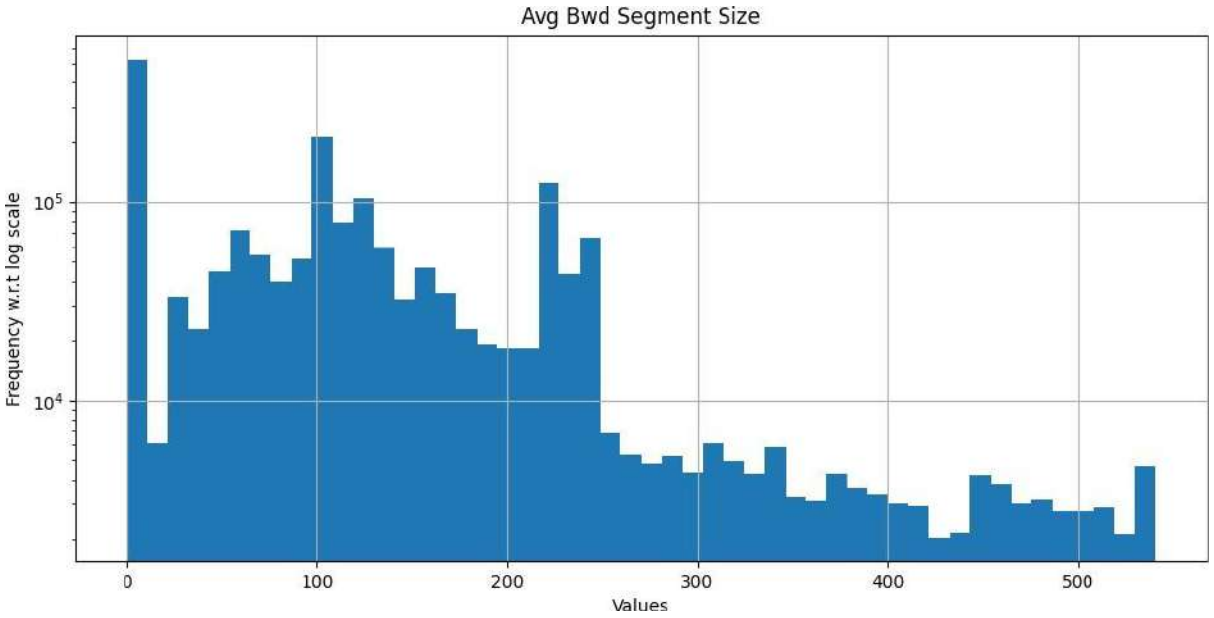


Figure 4.11.41 Histogram of Avg Bwd Segment Size plotted on log scale after handling negative values and outliers

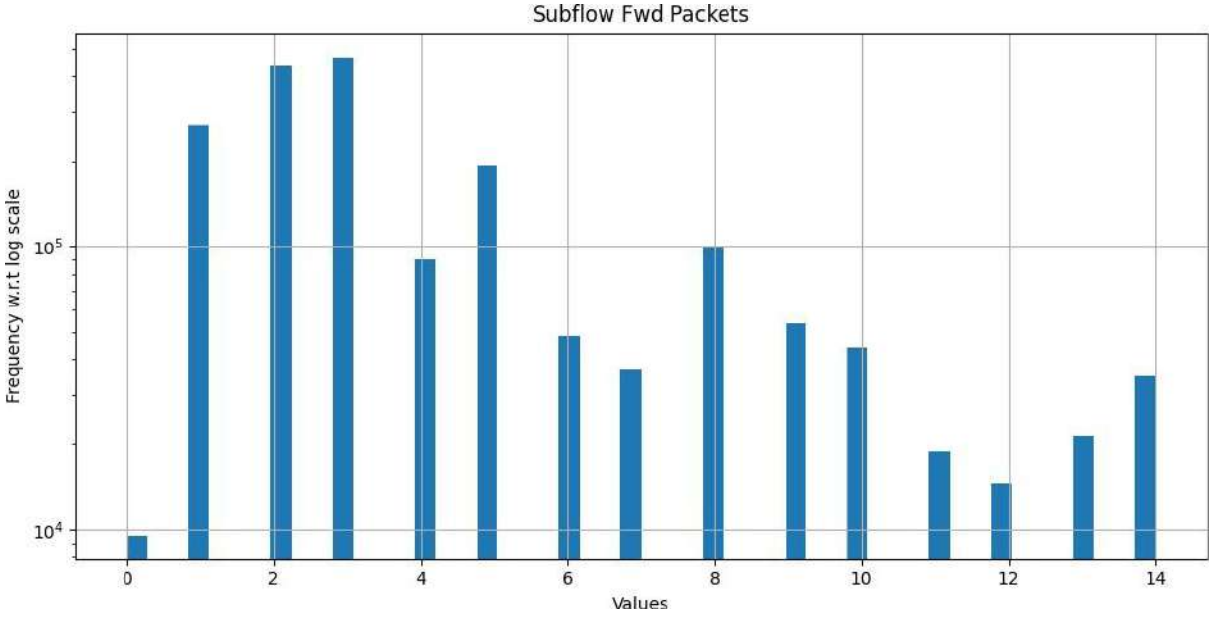


Figure 4.11.42 Histogram of Subflow Fwd Packets plotted on log scale after handling negative values and outliers

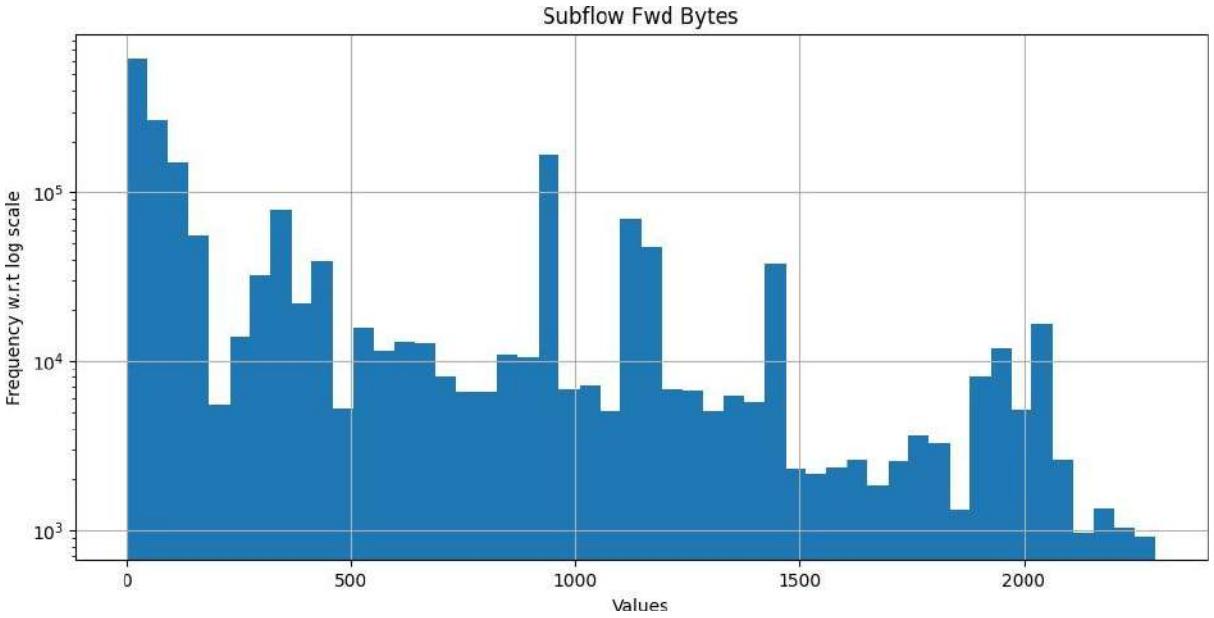


Figure 4.11.43 Histogram of Subflow Fwd Bytes plotted on log scale after handling negative values and outliers

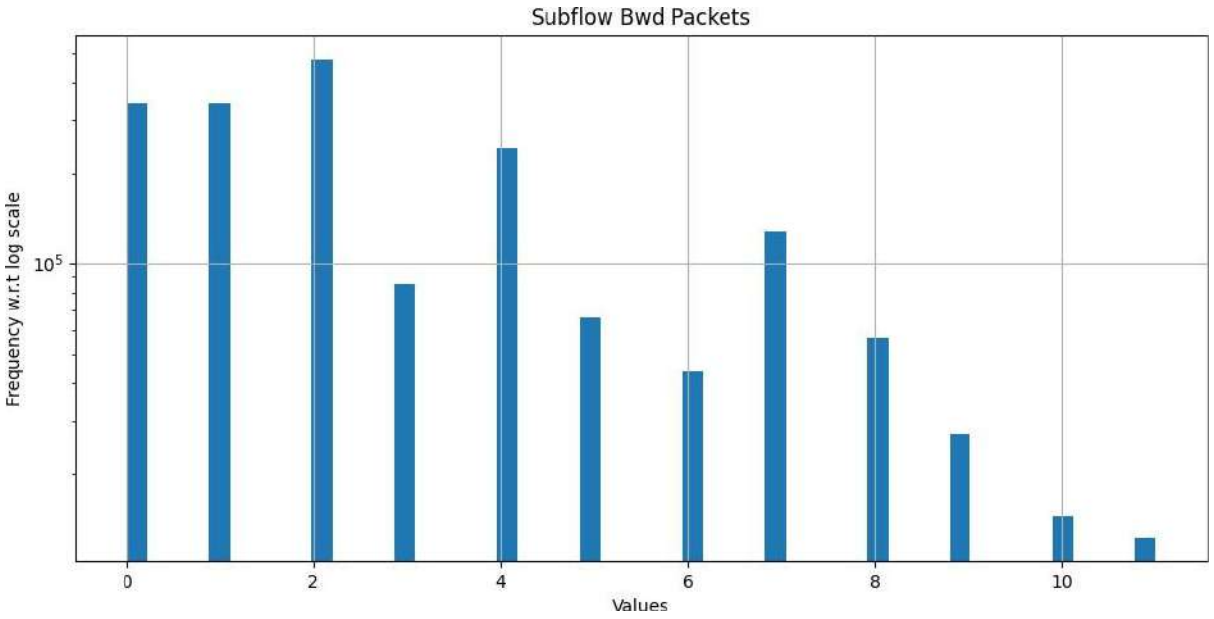


Figure 4.11.44 Histogram of Subflow Bwd Packets plotted on log scale after handling negative values and outliers

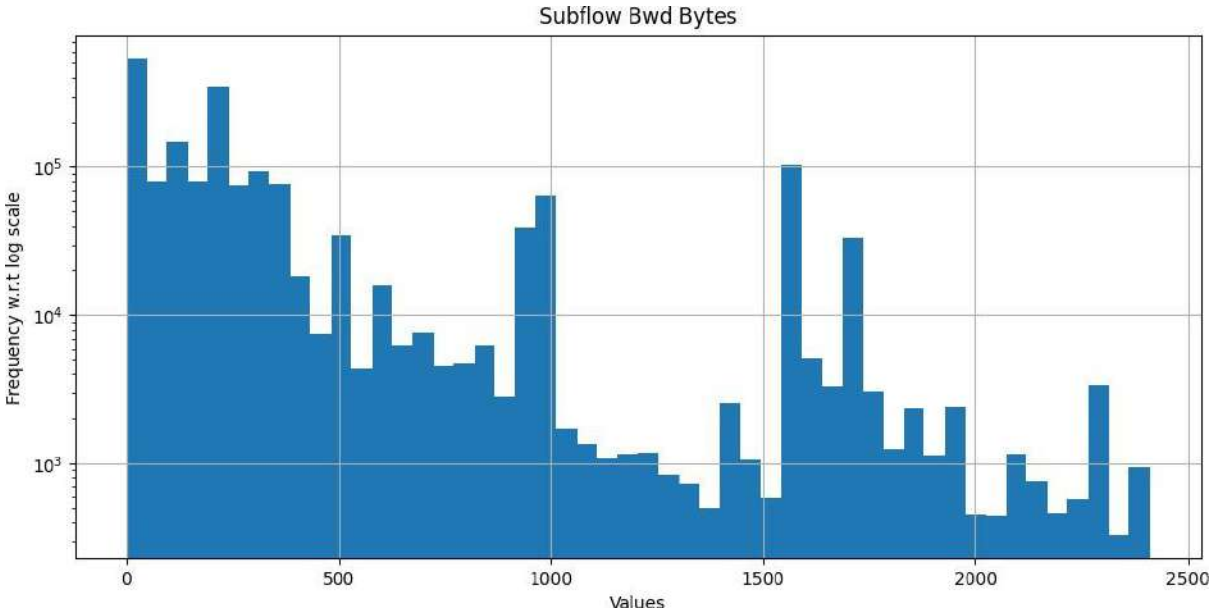


Figure 4.11.45 Histogram of Subflow Bwd Bytes plotted on log scale after handling negative values and outliers

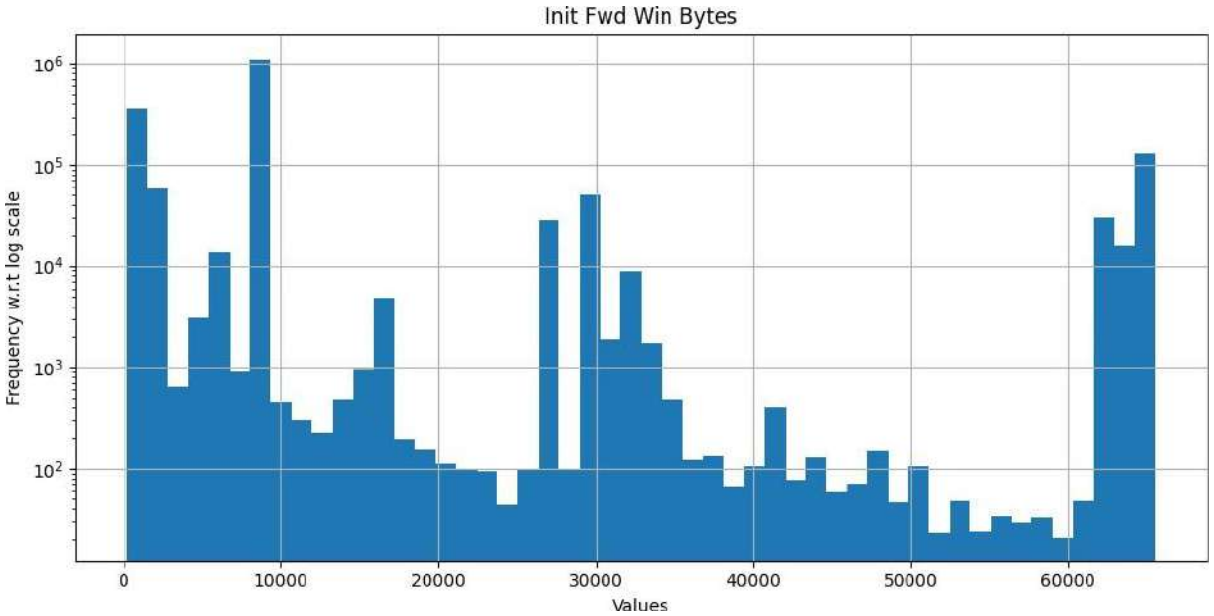


Figure 4.11.46 Histogram of Init Fwd Win Bytes plotted on log scale after handling negative values and outliers

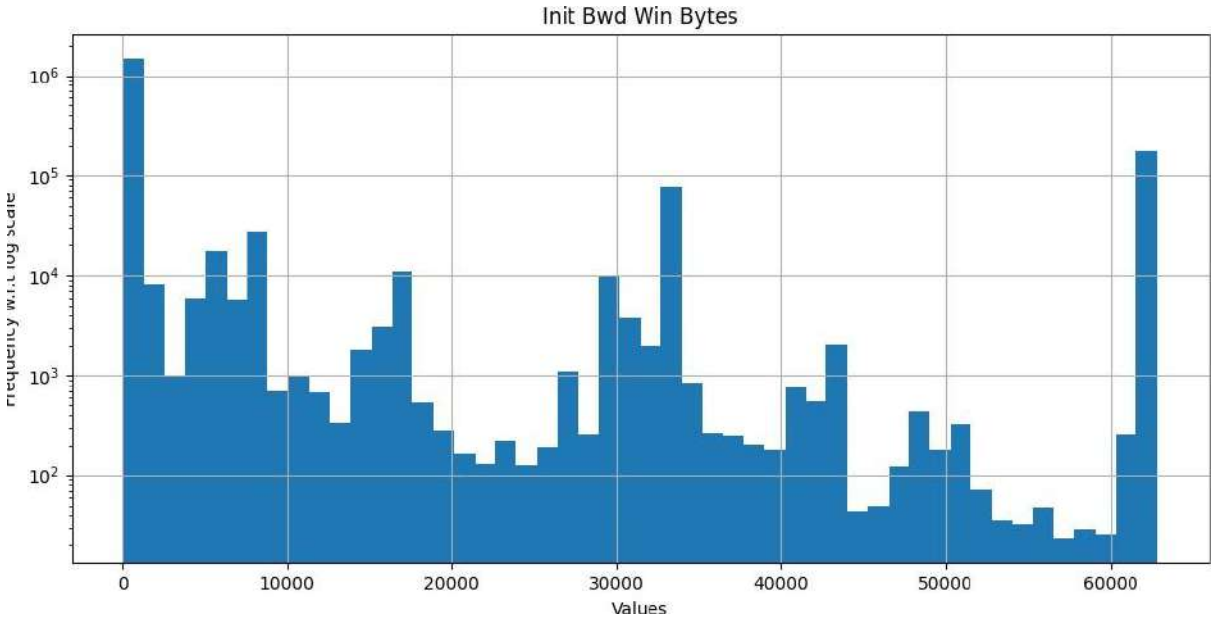


Figure 4.11.47 Histogram of Init Bwd Win Bytes plotted on log scale after handling negative values and outliers

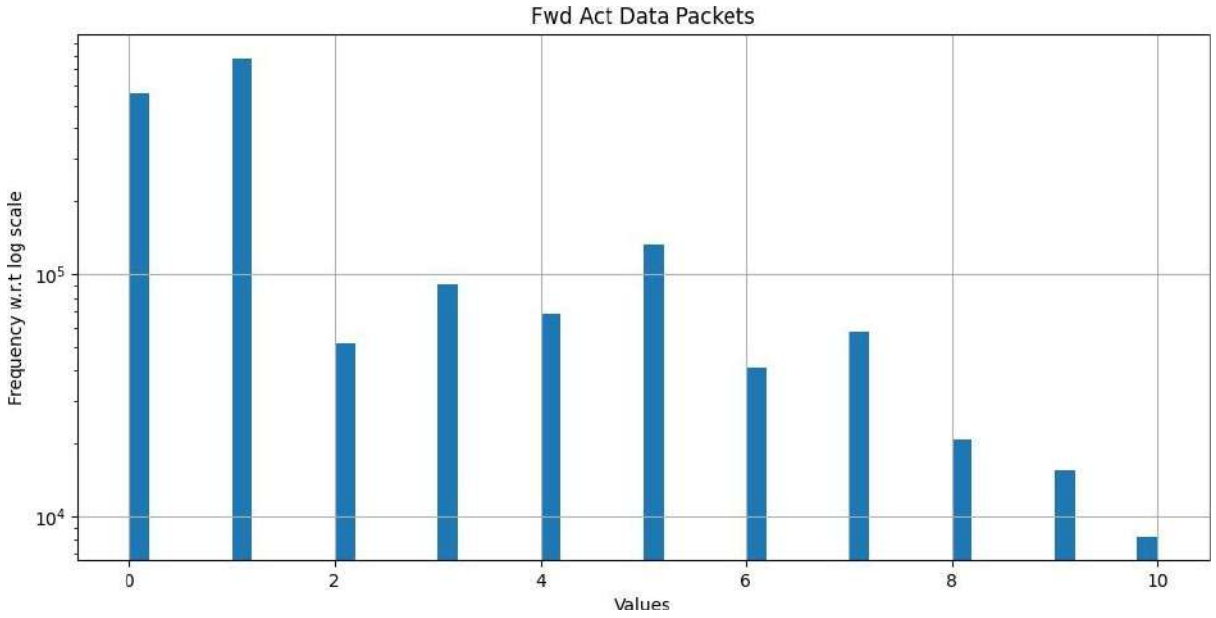


Figure 4.11.48 Histogram of Fwd Act Data Packets plotted on log scale after handling negative values and outliers

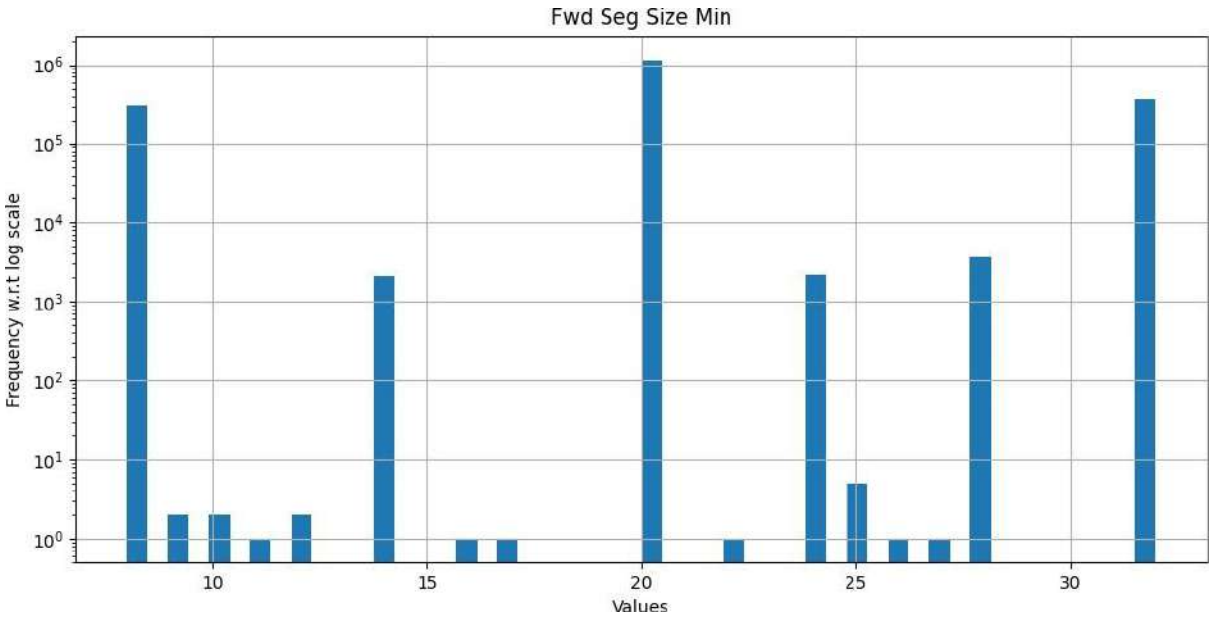


Figure 4.11.49 Histogram of Fwd Seg Size Min plotted on log scale after handling negative values and outliers

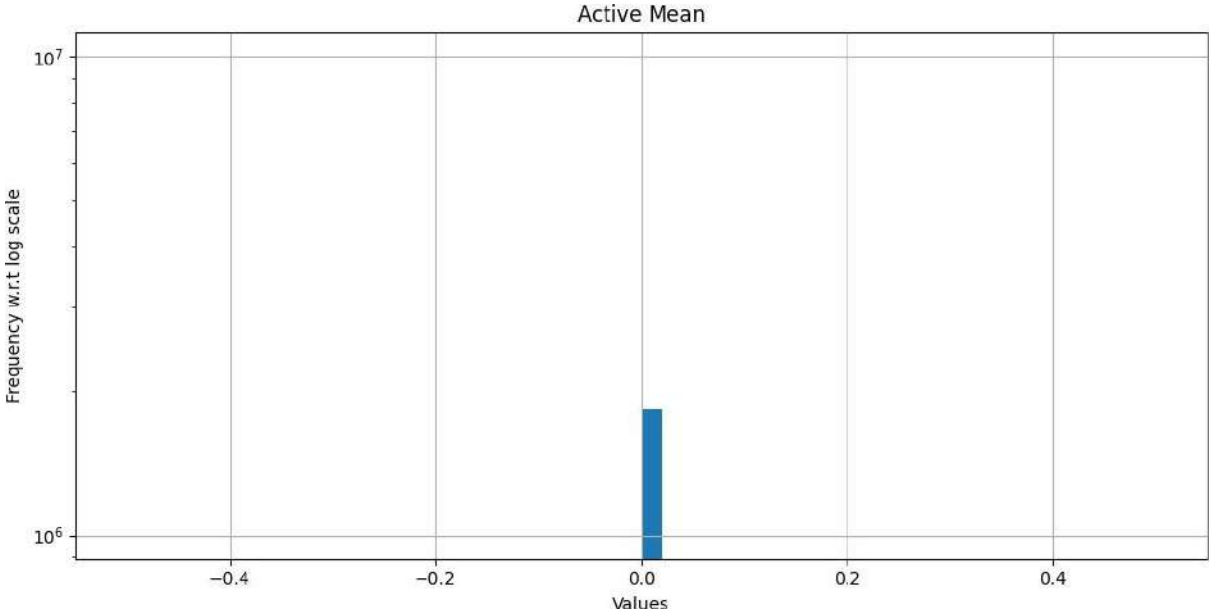


Figure 4.11.50 Histogram of Active Mean plotted on log scale after handling negative values and outliers

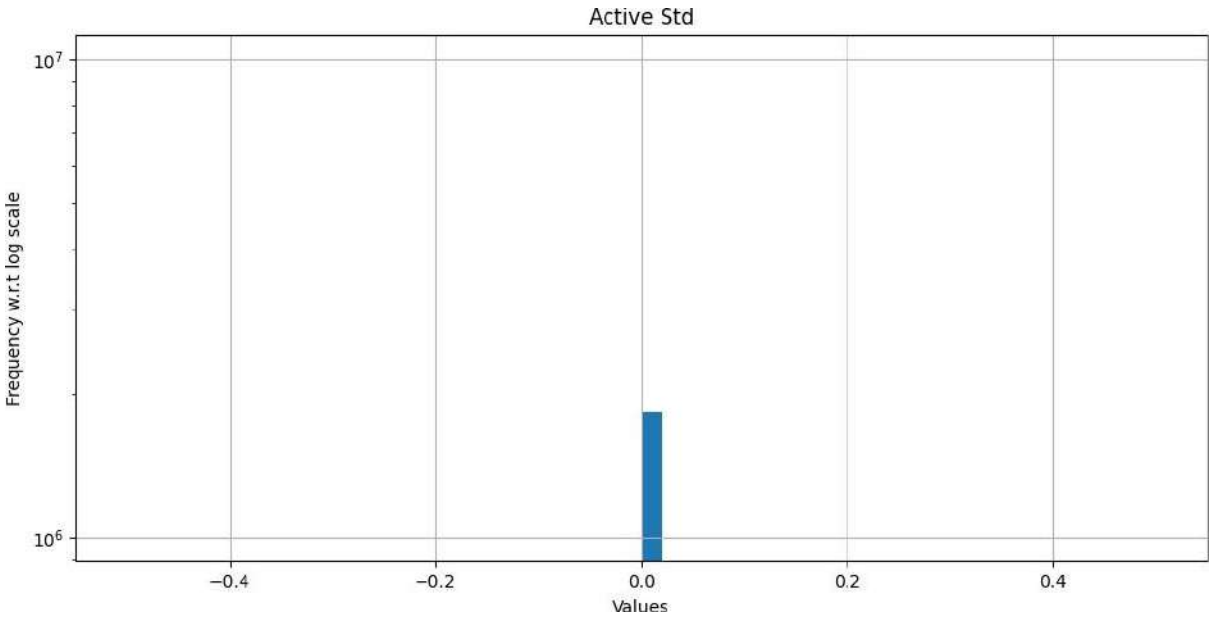


Figure 4.11.51 Histogram of Active Std plotted on log scale after handling negative values and outliers

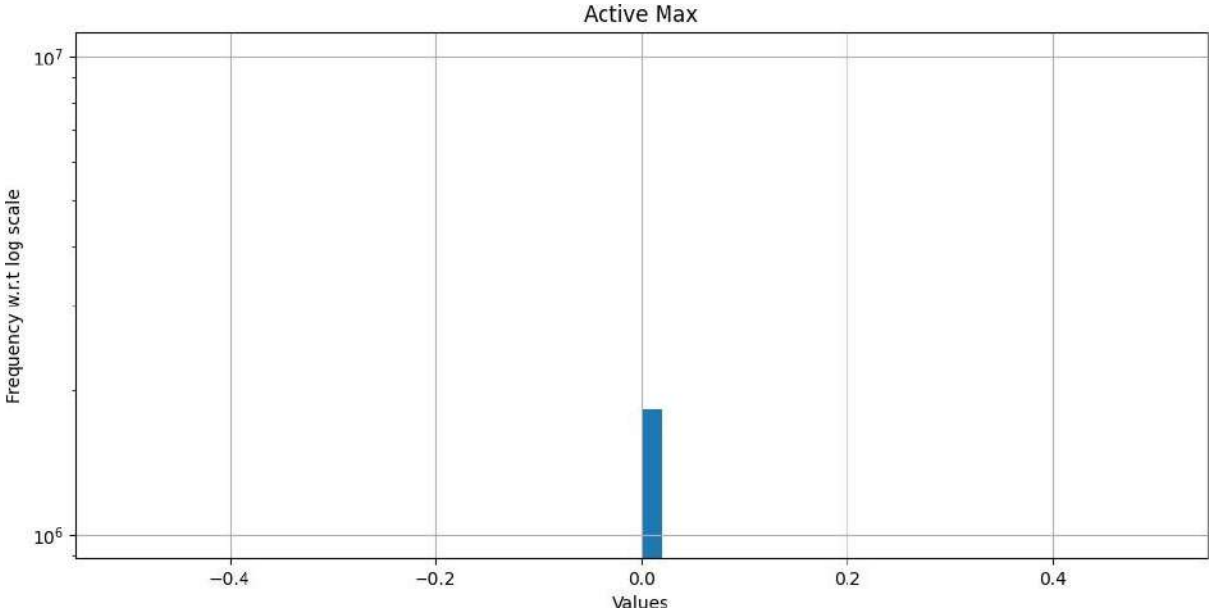


Figure 4.11.52 Histogram of Active Max plotted on log scale after handling negative values and outliers

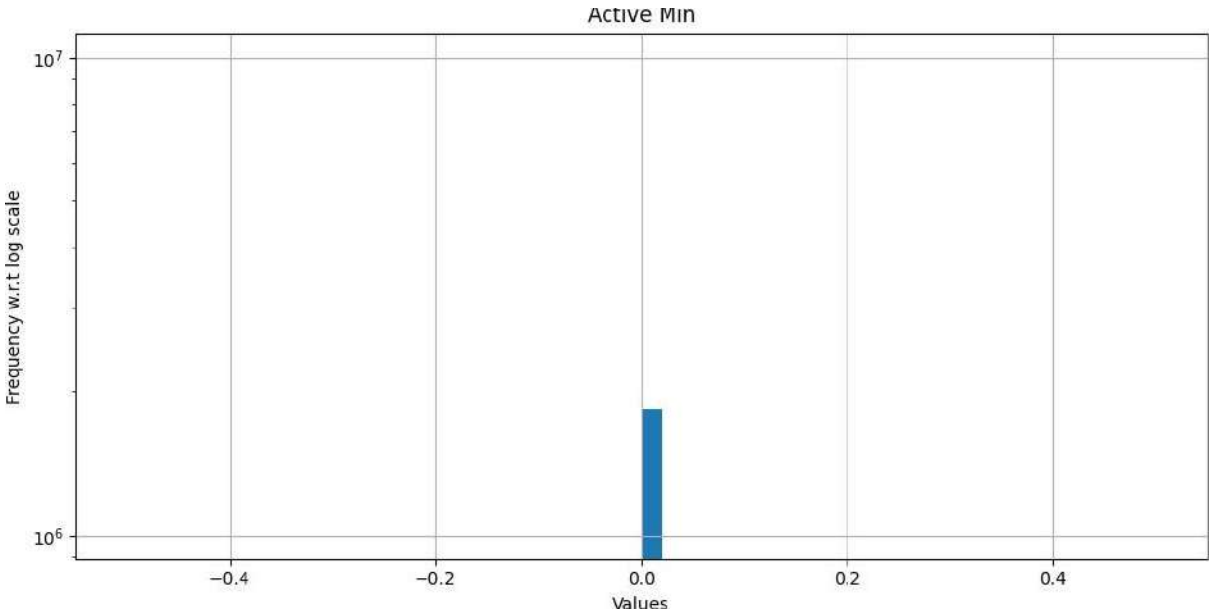


Figure 4.11.53 Histogram of Active Min plotted on log scale after handling negative values and outliers

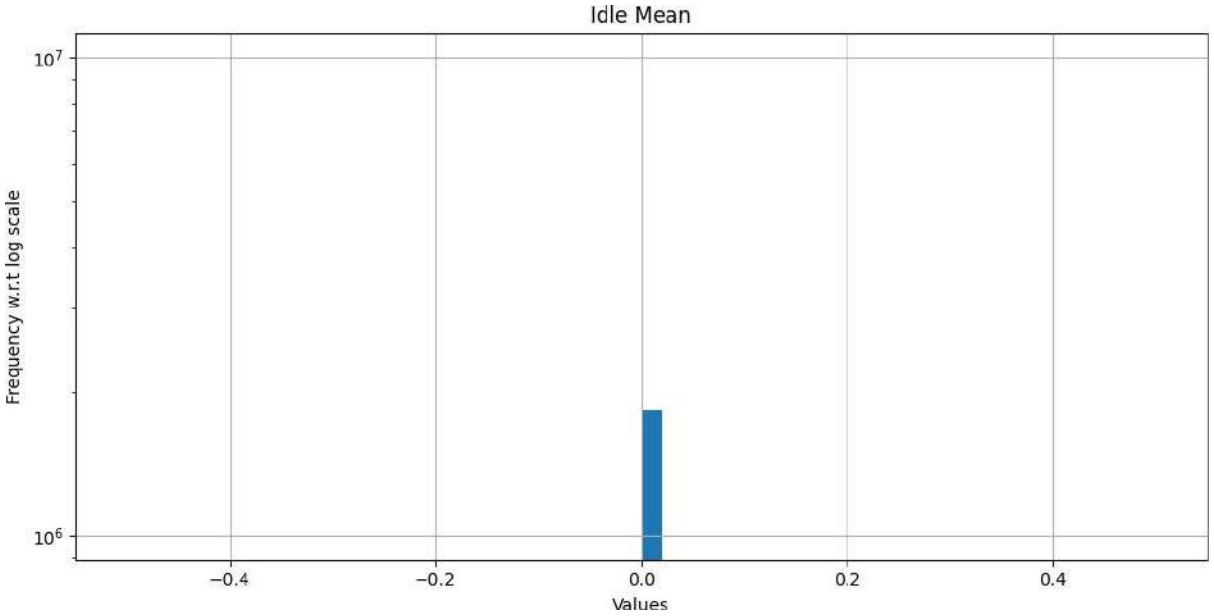


Figure 4.11.54 Histogram of Idle Mean plotted on log scale after handling negative values and outliers

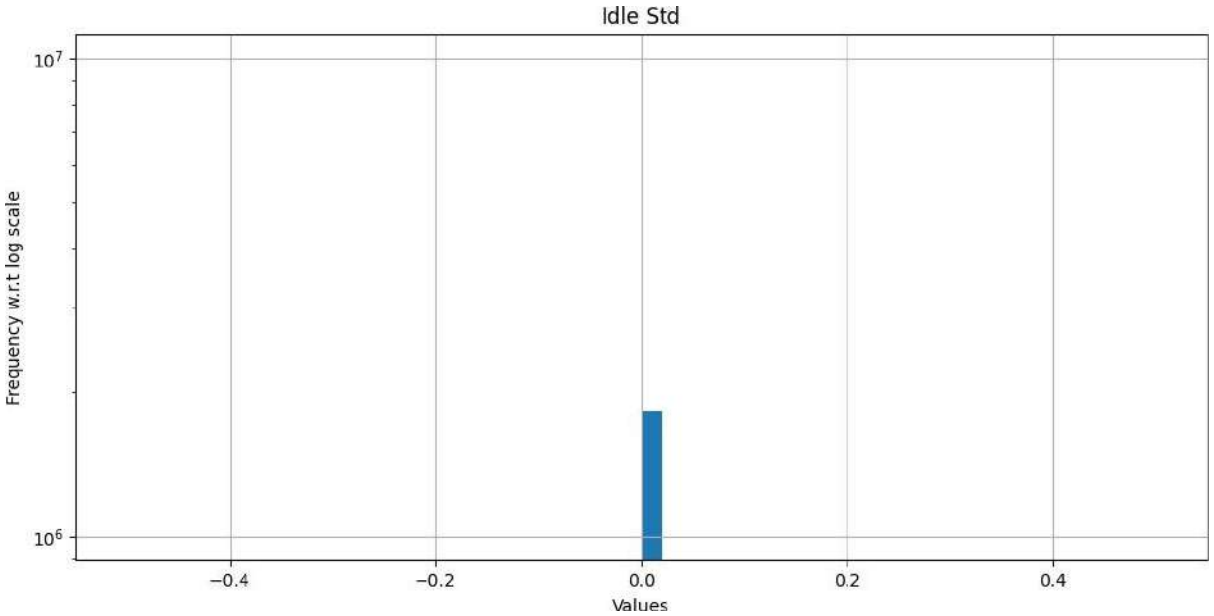


Figure 4.11.55 Histogram of Idle Std plotted on log scale after handling negative values and outliers

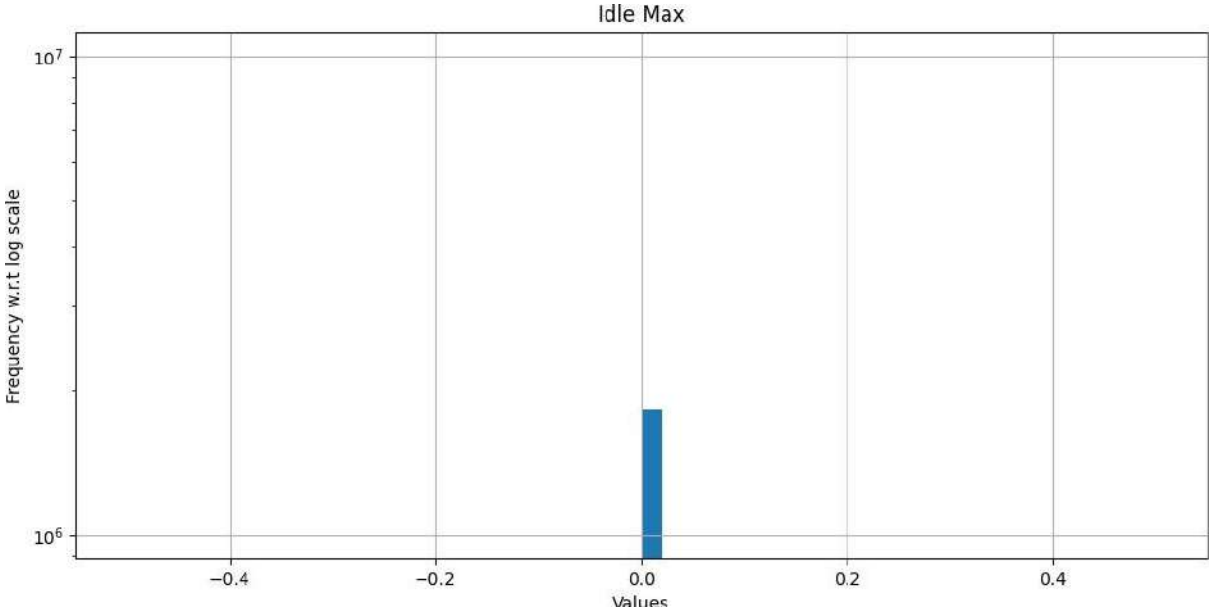


Figure 4.11.56 Histogram of Idle Max plotted on log scale after handling negative values and outliers

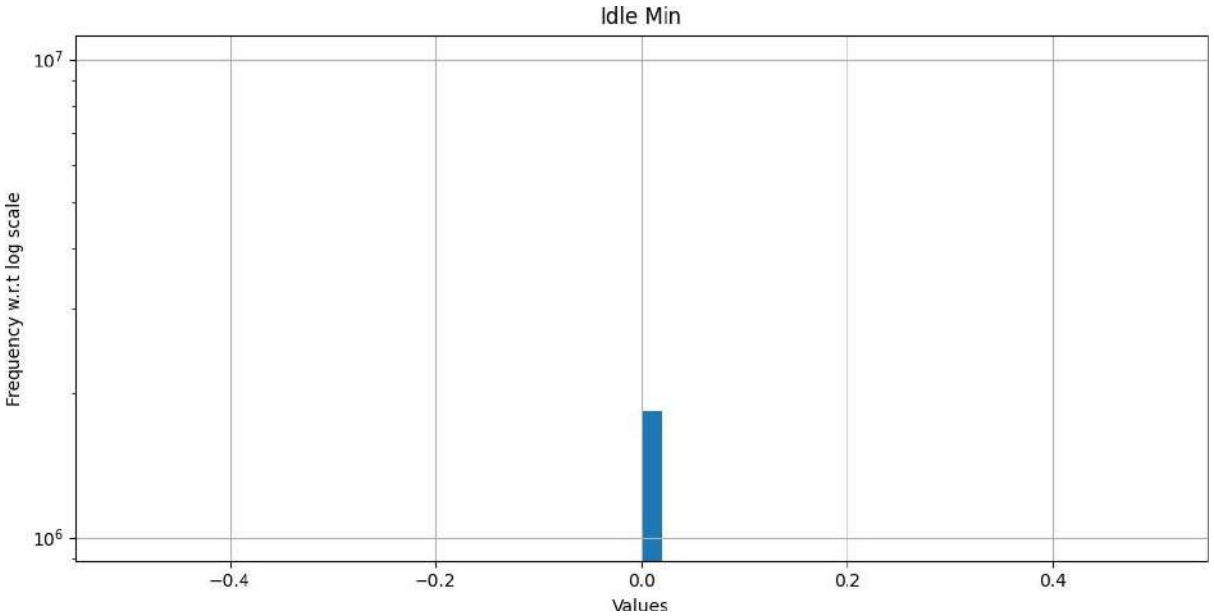


Figure 4.11.57 Histogram of Idle Min plotted on log scale after handling negative values and outliers

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).

4.12 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.

Thus, it was essential to determine the optimal number of bins to plot these charts for each independent feature.

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, Fredman-Diaconis rule was used.

Freedman-Diaconis rule was selected because: -

1. It helps to compute bin width based on each feature's IQR. Thus, it helps to reduce the impact of skewness in data.
2. It does not assume the feature to be normally distributed.
3. Since uses IQR, the rule also helps to handle values at extreme end and compute optimal the number of bins.

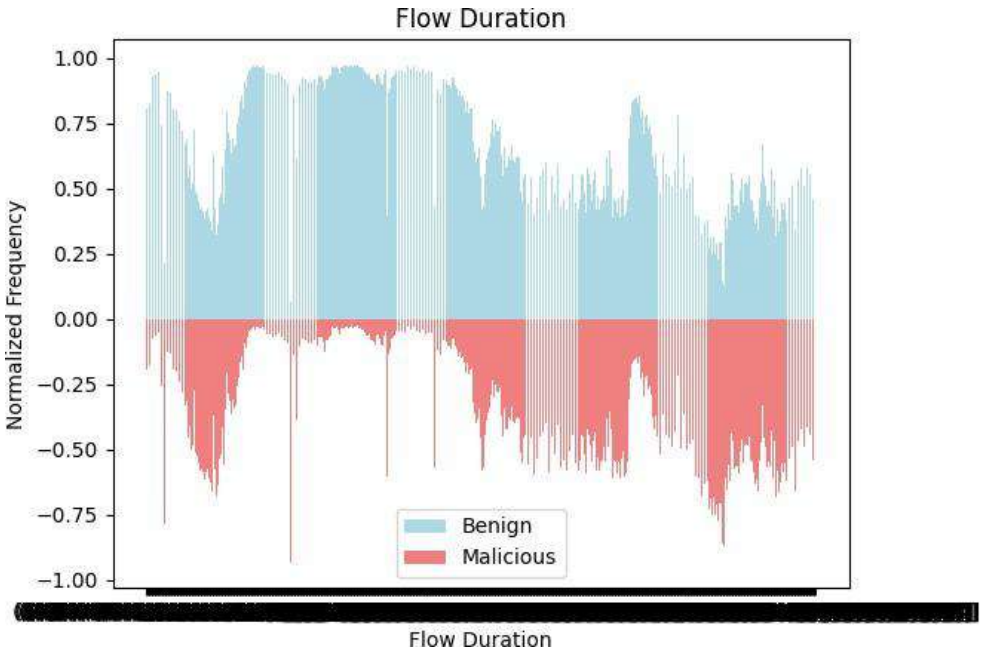


Figure 4.12.1 Pyramid chart of Flow Duration w.r.t isMalicious

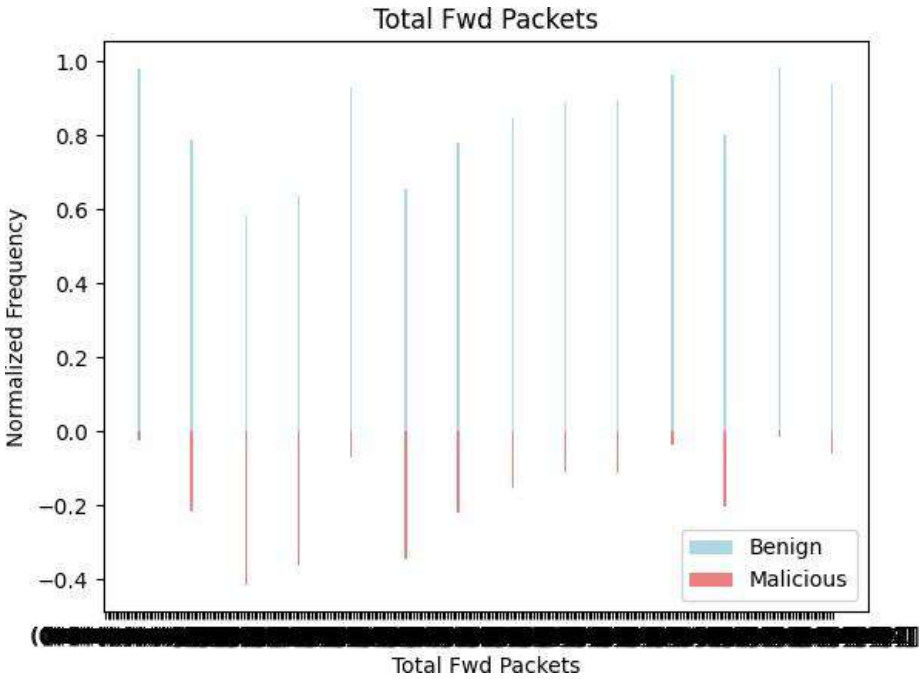


Figure 4.12.2 Pyramid chart of Total Fwd Packets w.r.t isMalicious

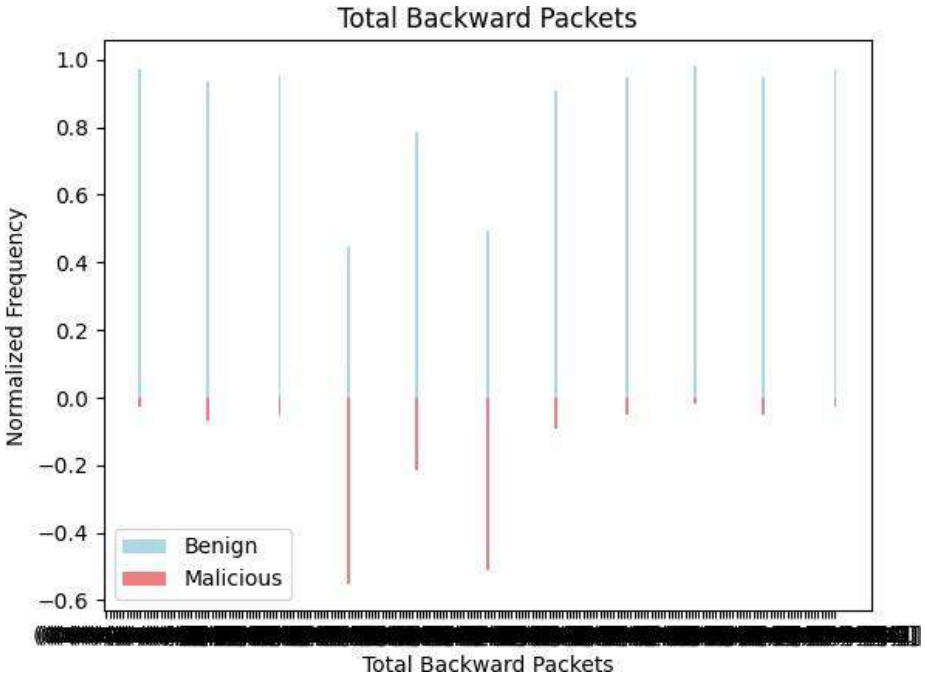


Figure 4.12.3 Pyramid chart of Total Backward Packets w.r.t isMalicious

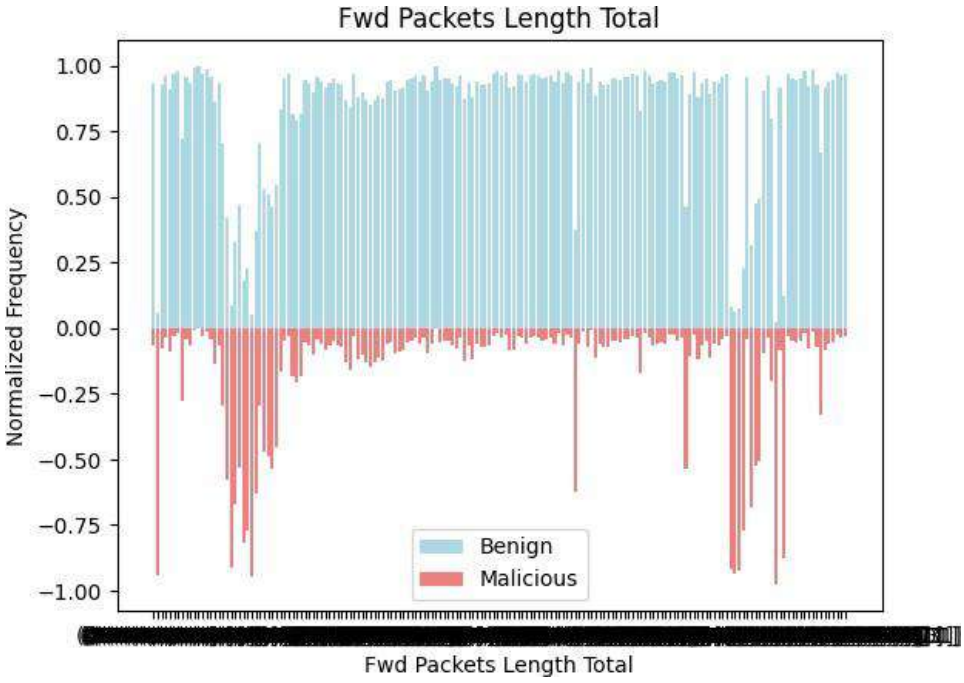


Figure 4.12.4 Pyramid chart of Fwd Packets Length Total w.r.t isMalicious

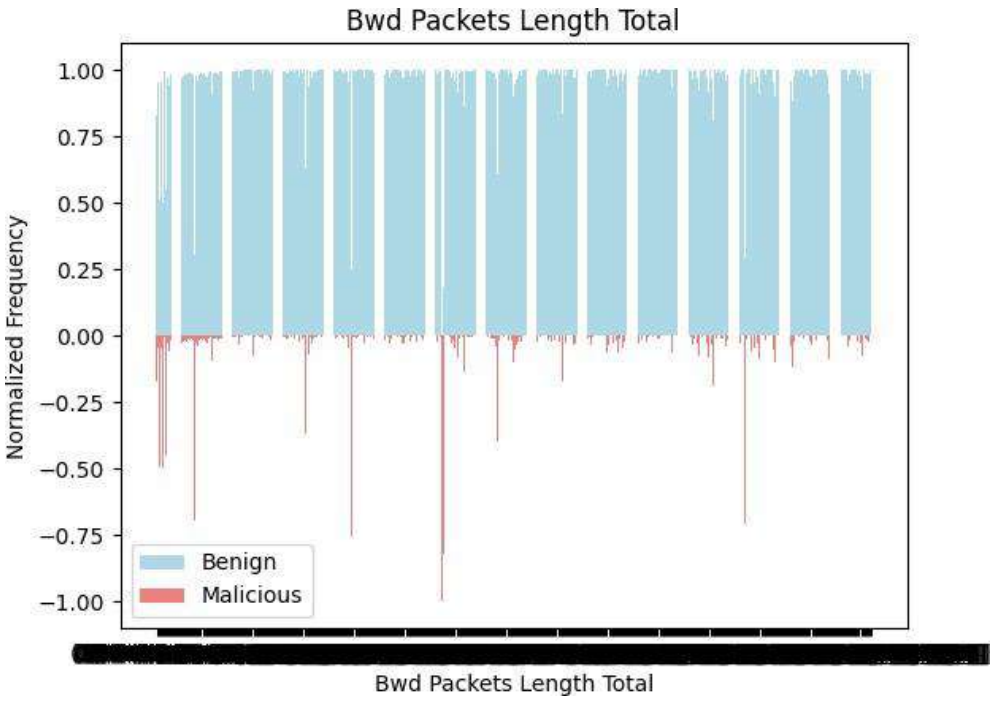


Figure 4.12.5 Pyramid chart of Bwd Packets Length Total w.r.t isMalicious

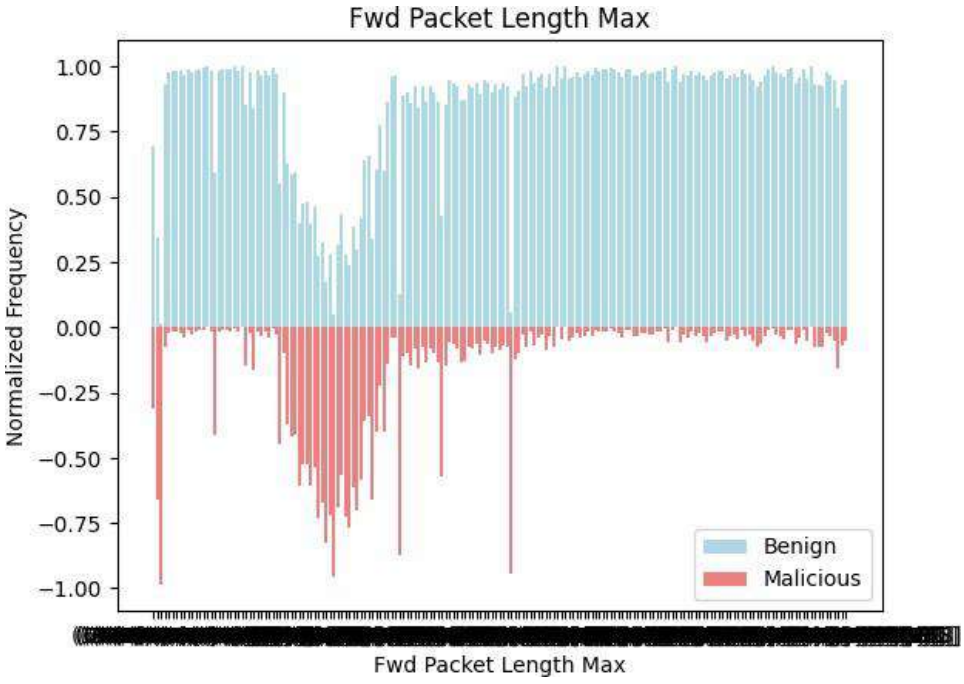


Figure 4.12.6 Pyramid chart of Fwd Packet Length Max w.r.t isMalicious

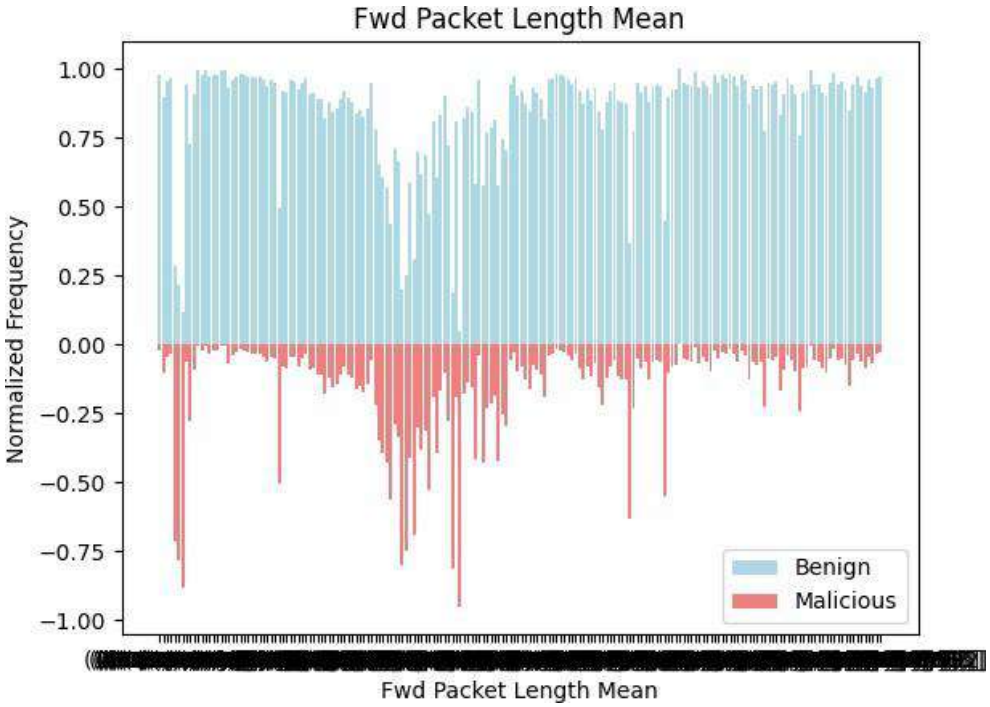


Figure 4.12.7 Pyramid chart of Fwd Packet Length Mean w.r.t isMalicious

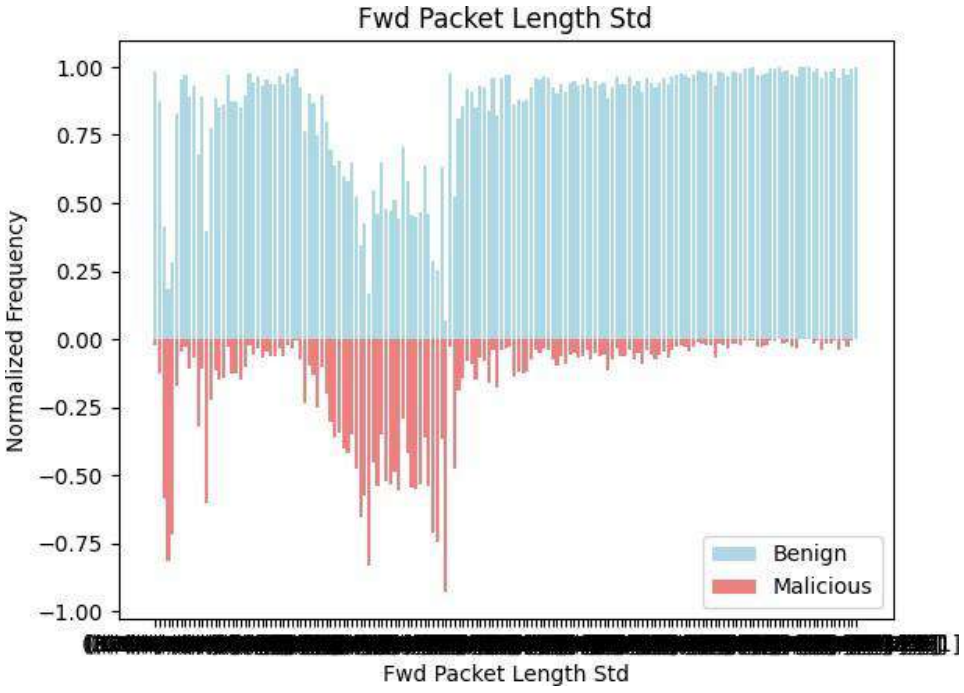


Figure 4.12.8 Pyramid chart of Fwd Packet Length Std w.r.t isMalicious

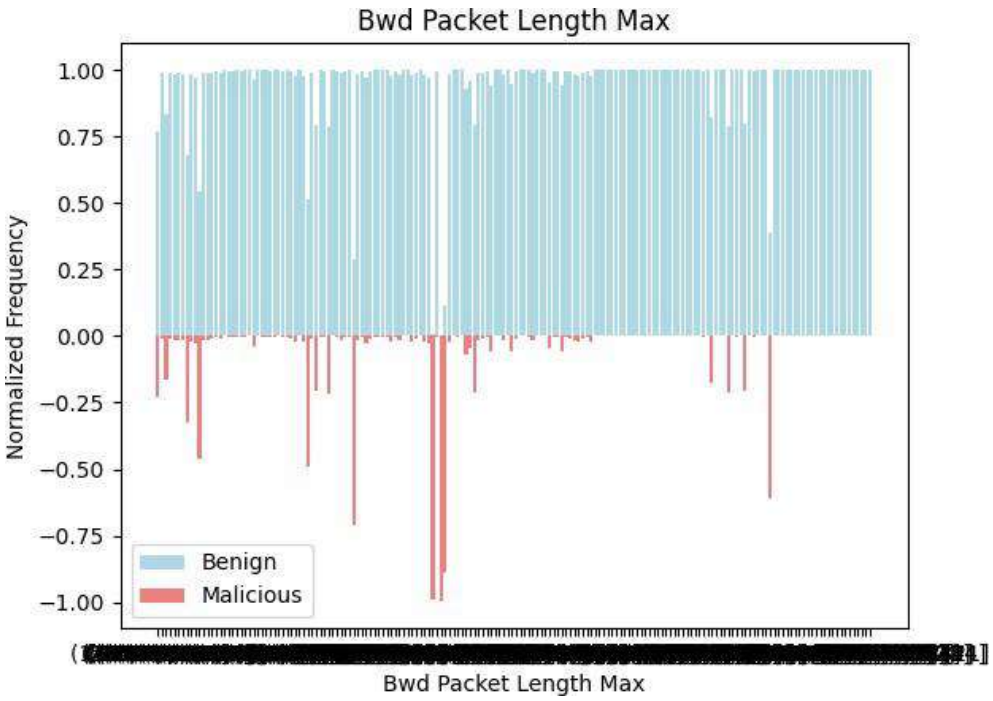


Figure 4.12.9 Pyramid chart of Bwd Packet Length Max w.r.t isMalicious

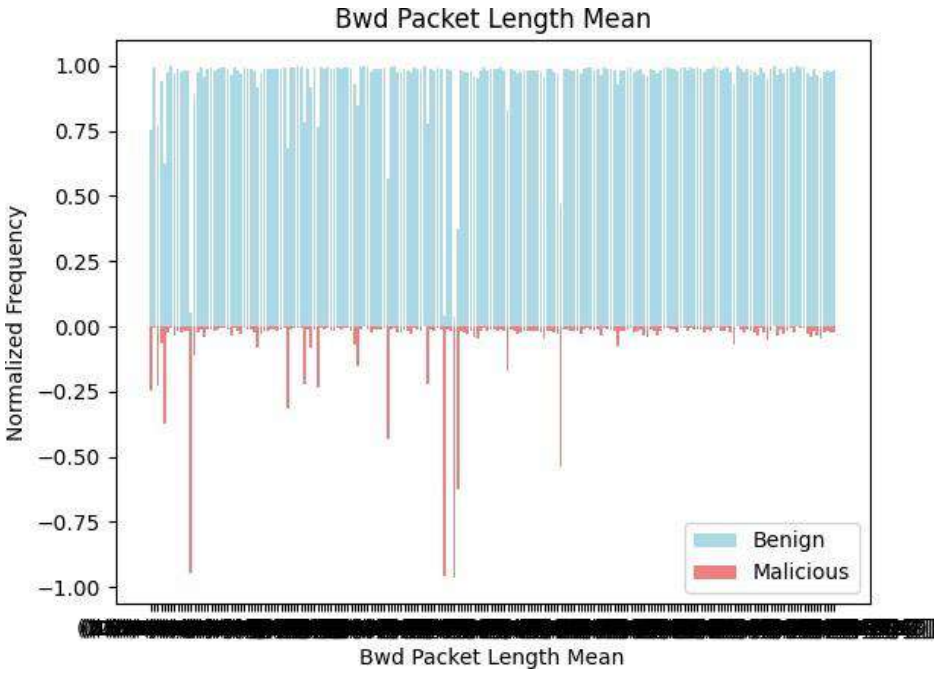


Figure 4.12.10 Pyramid chart of Bwd Packet Length Mean w.r.t isMalicious

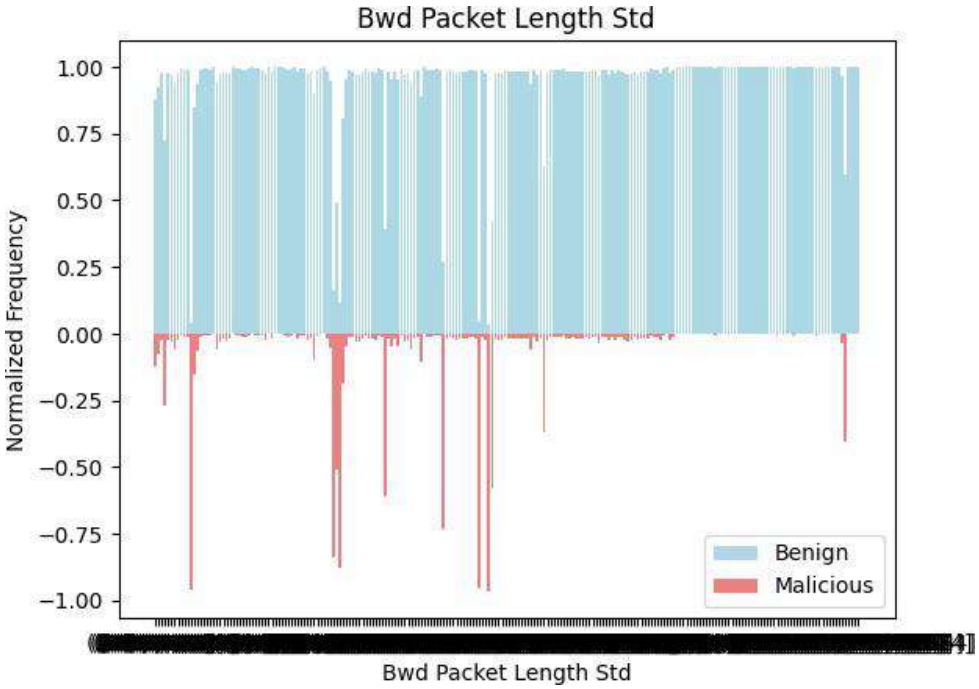


Figure 4.12.11 Pyramid chart of Bwd Packet Length Std w.r.t isMalicious

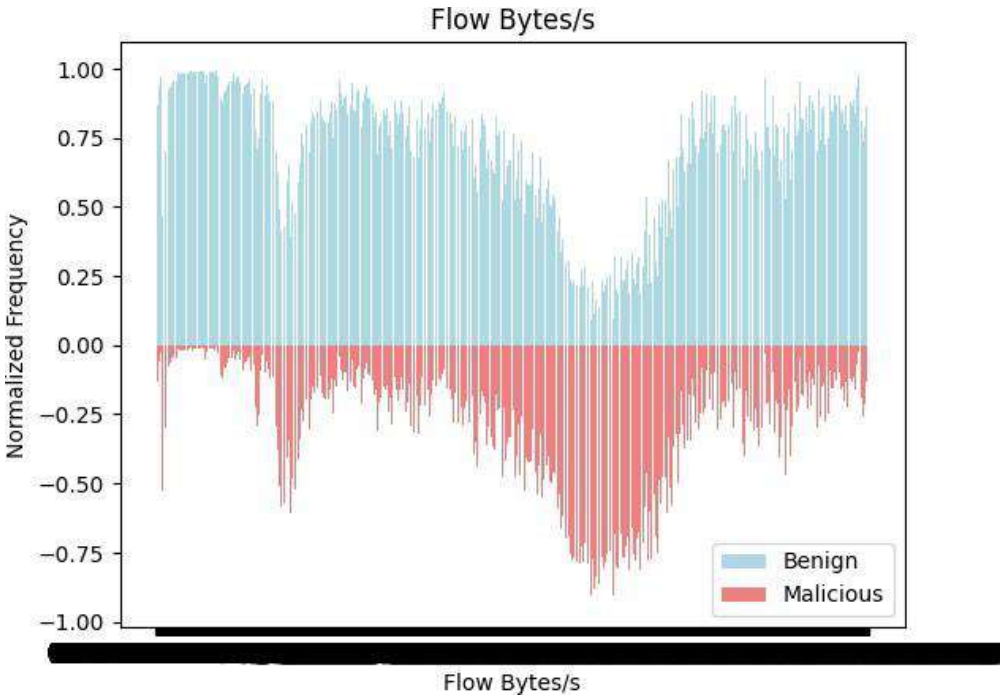


Figure 4.12.12 Pyramid chart of Flow Bytes/s w.r.t isMalicious

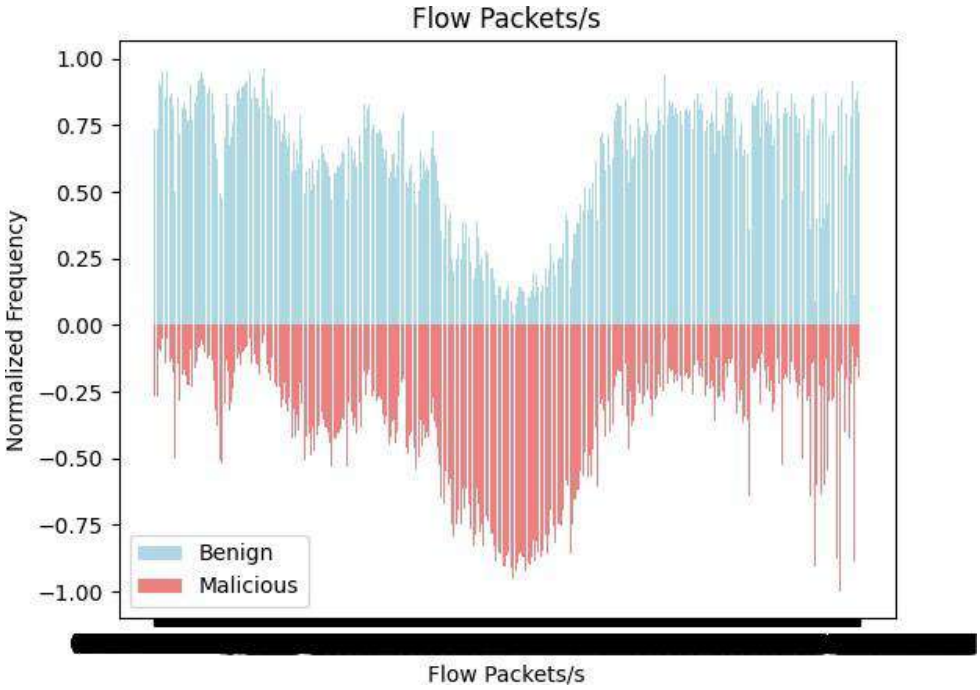


Figure 4.12.13 Pyramid chart of Flow Packets/s w.r.t isMalicious

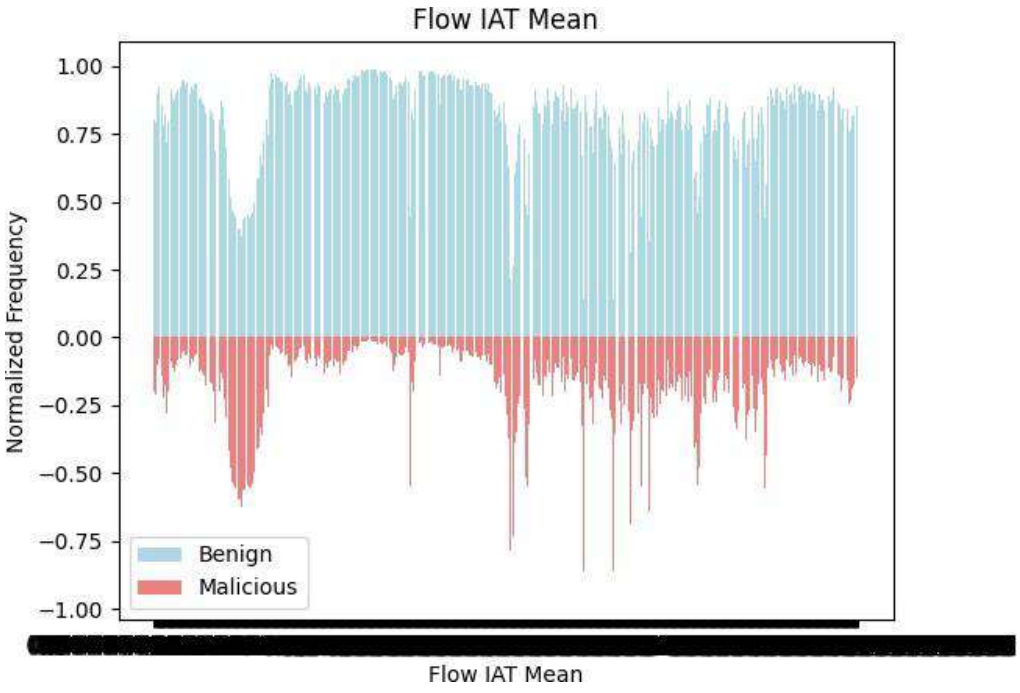


Figure 4.12.14 Pyramid chart of Flow IAT Mean w.r.t isMalicious

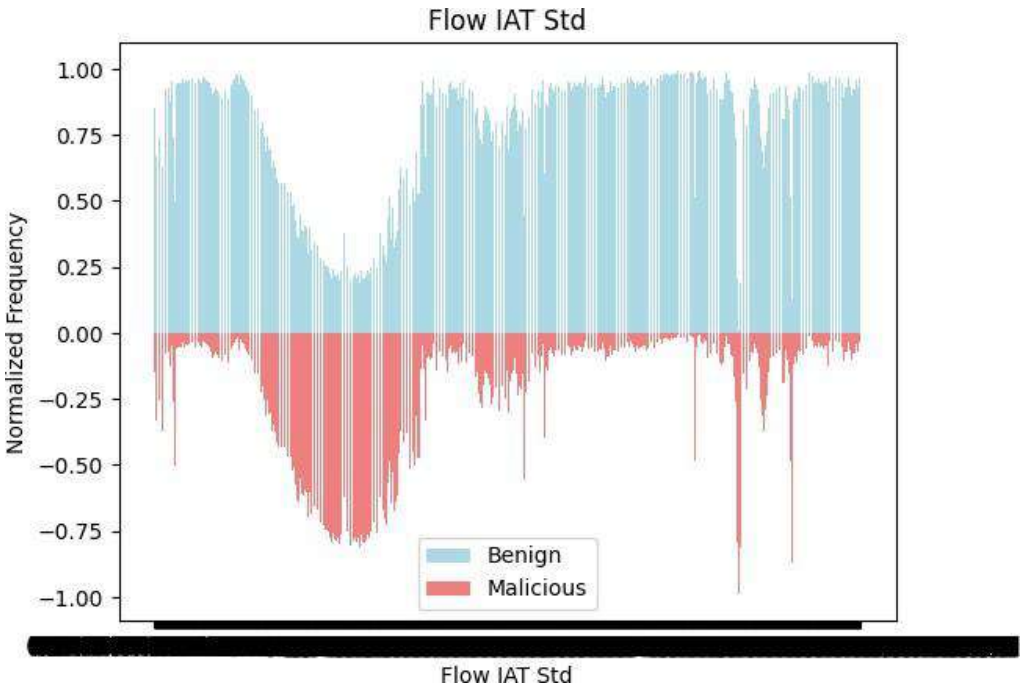


Figure 4.12.15 Pyramid chart of Flow IAT Std w.r.t isMalicious

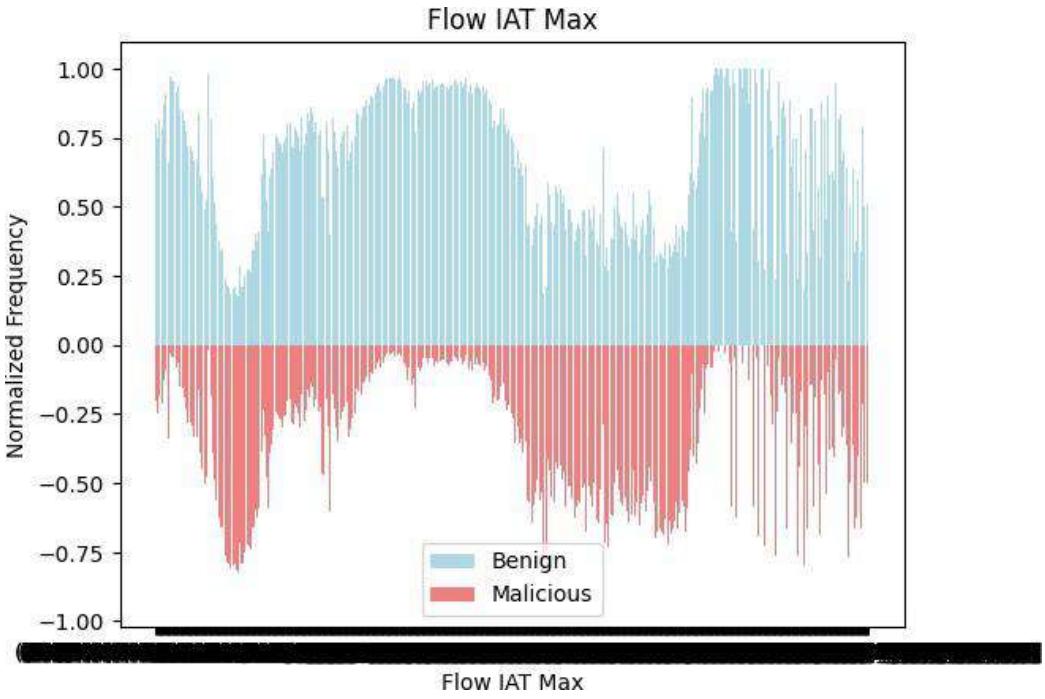


Figure 4.12.16 Pyramid chart of Flow IAT Max w.r.t isMalicious

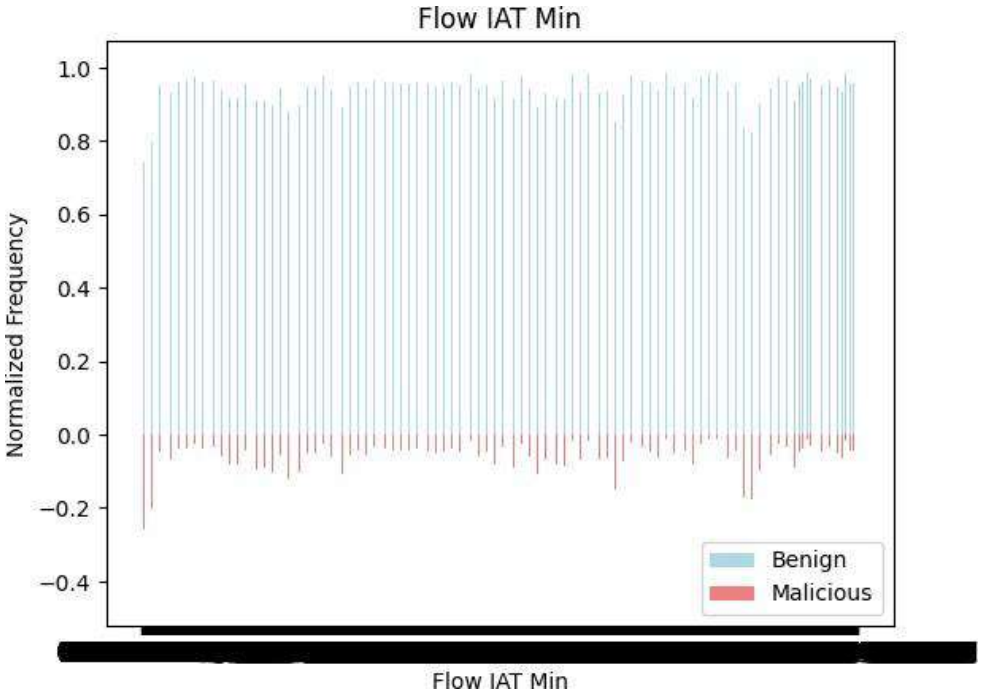


Figure 4.12.17 Pyramid chart of Flow IAT Min w.r.t isMalicious

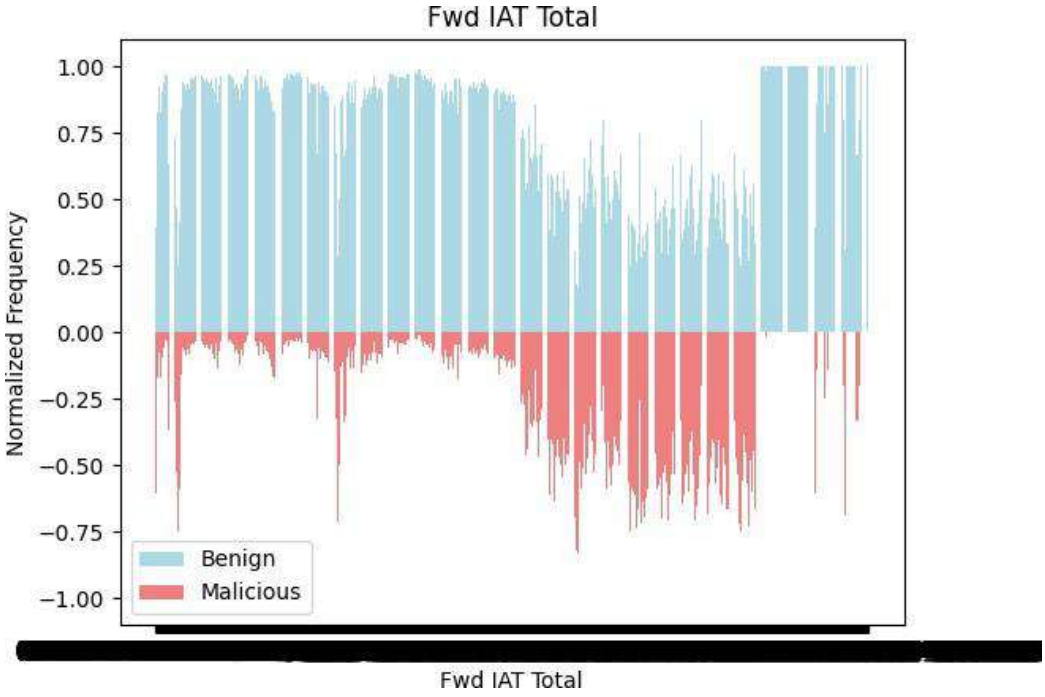


Figure 4.12.18 Pyramid chart of Fwd IAT Total w.r.t isMalicious

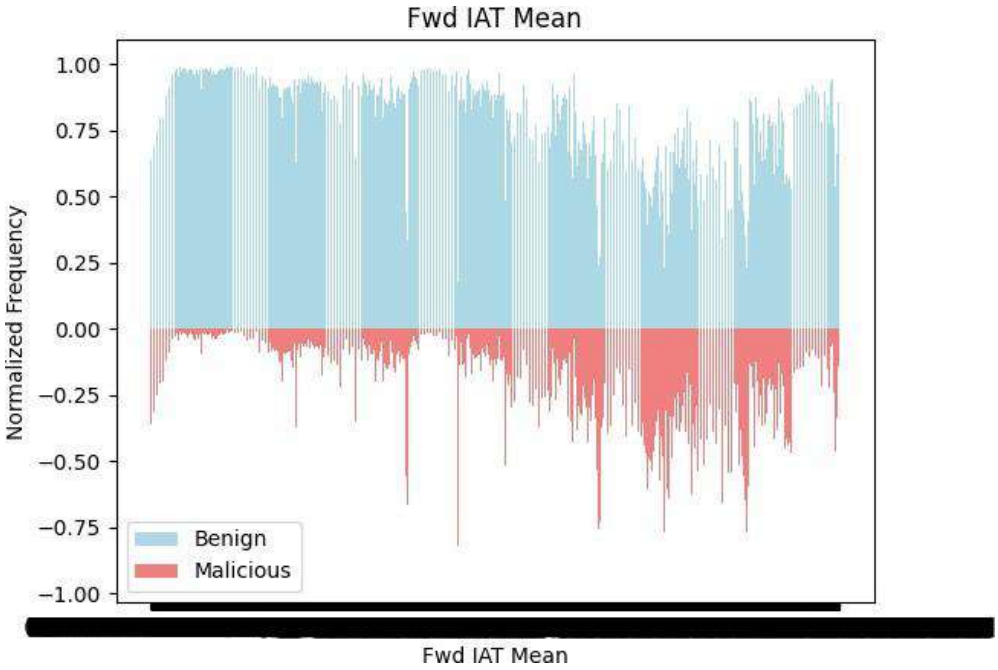


Figure 4.12.19 Pyramid chart of Fwd IAT Mean w.r.t isMalicious

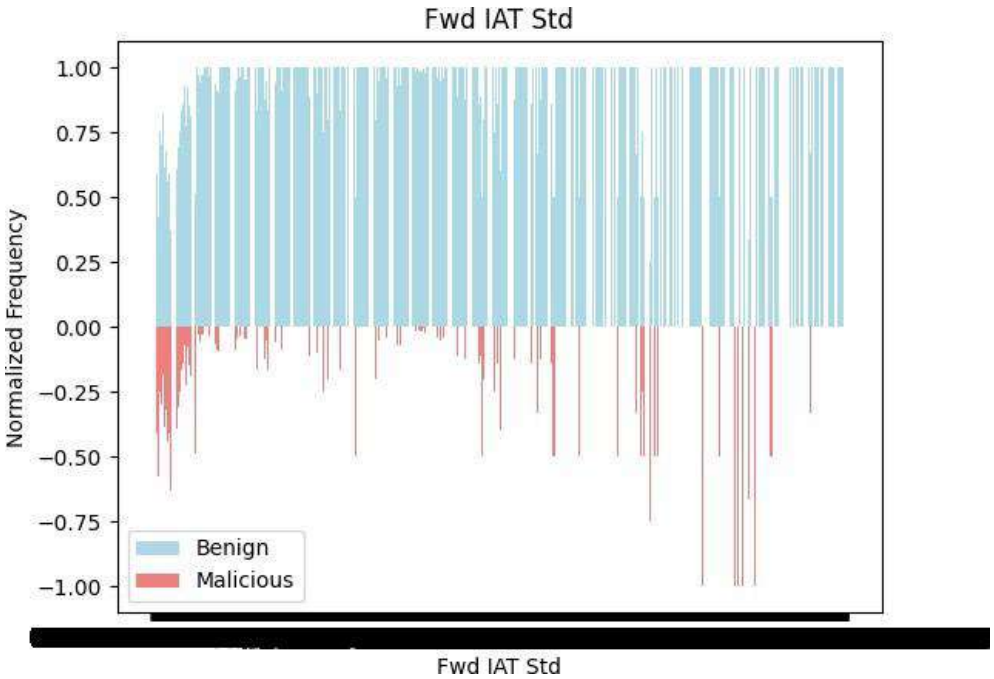


Figure 4.12.20 Pyramid chart of Fwd IAT Std w.r.t isMalicious

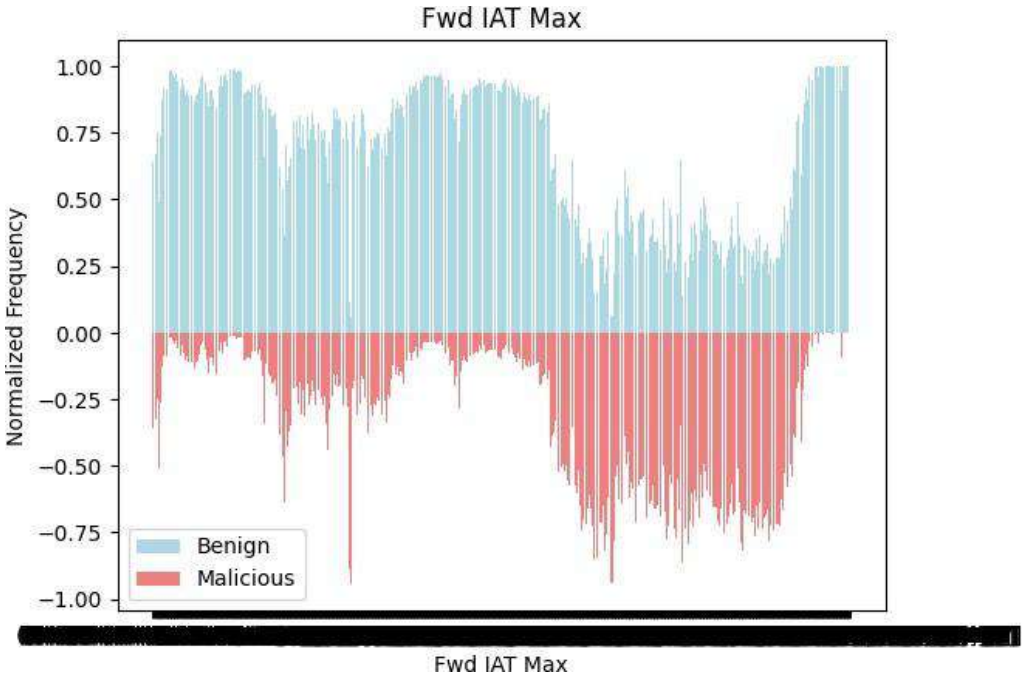


Figure 4.12.21 Pyramid chart of Fwd IAT Max w.r.t isMalicious

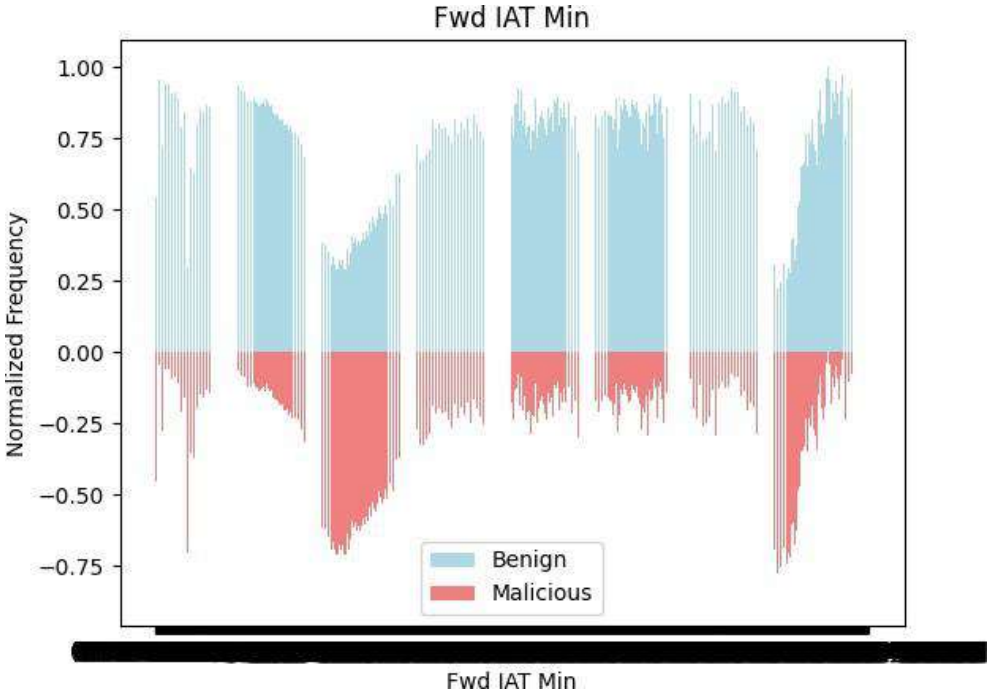


Figure 4.12.22 Pyramid chart of Fwd IAT Min w.r.t isMalicious

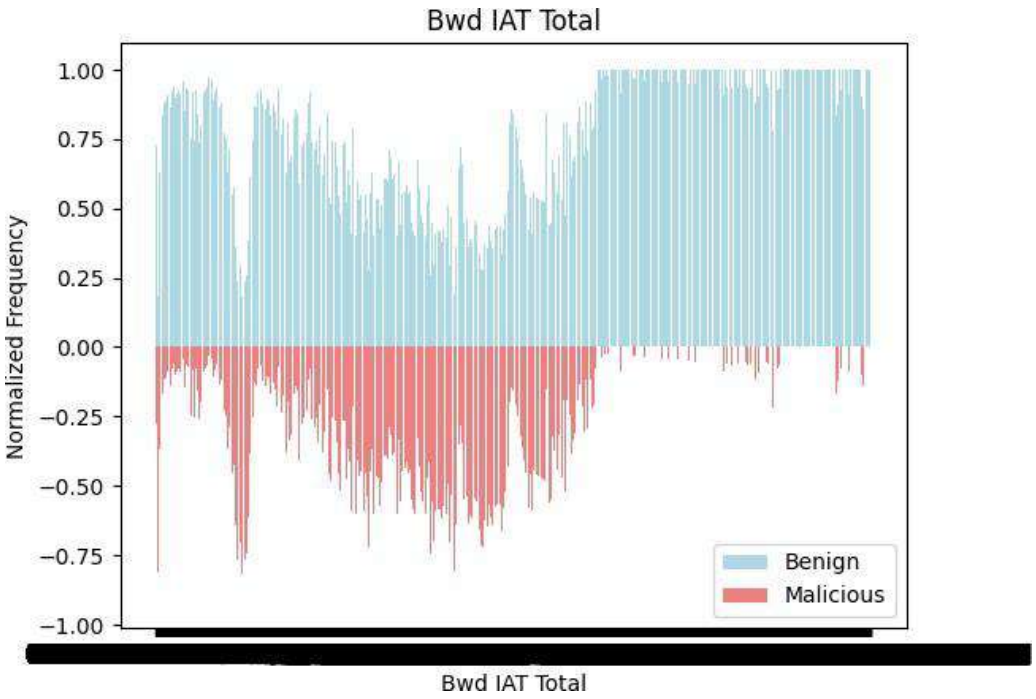


Figure 4.12.23 Pyramid chart of Bwd IAT Total w.r.t isMalicious

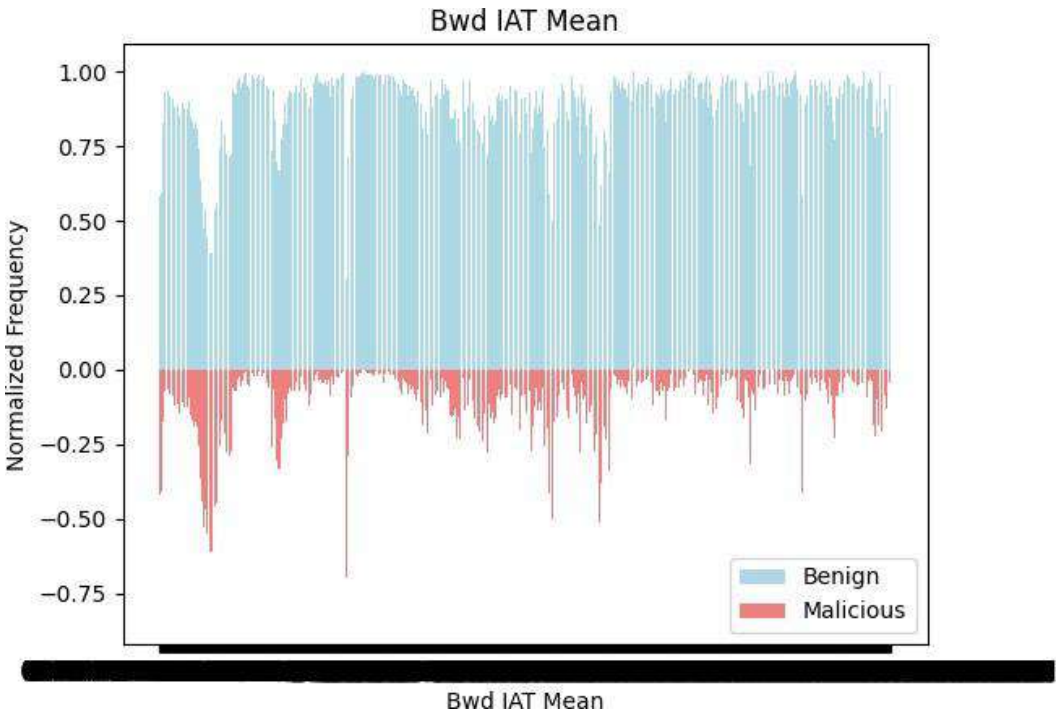


Figure 4.12.24 Pyramid chart of Bwd IAT Mean w.r.t isMalicious

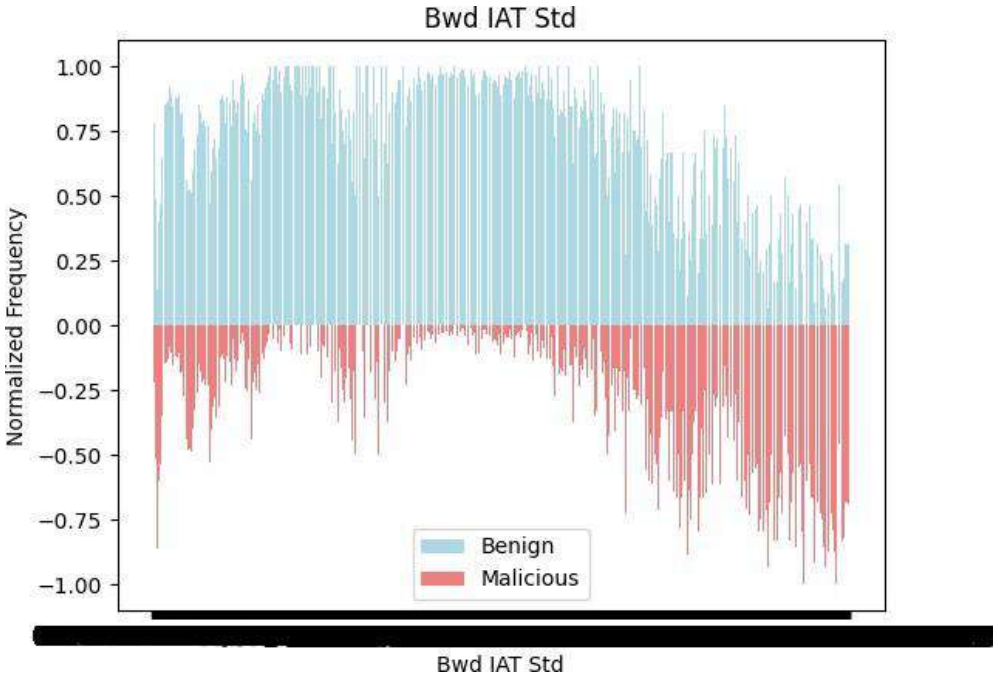


Figure 4.12.25 Pyramid chart of Bwd IAT Std w.r.t isMalicious

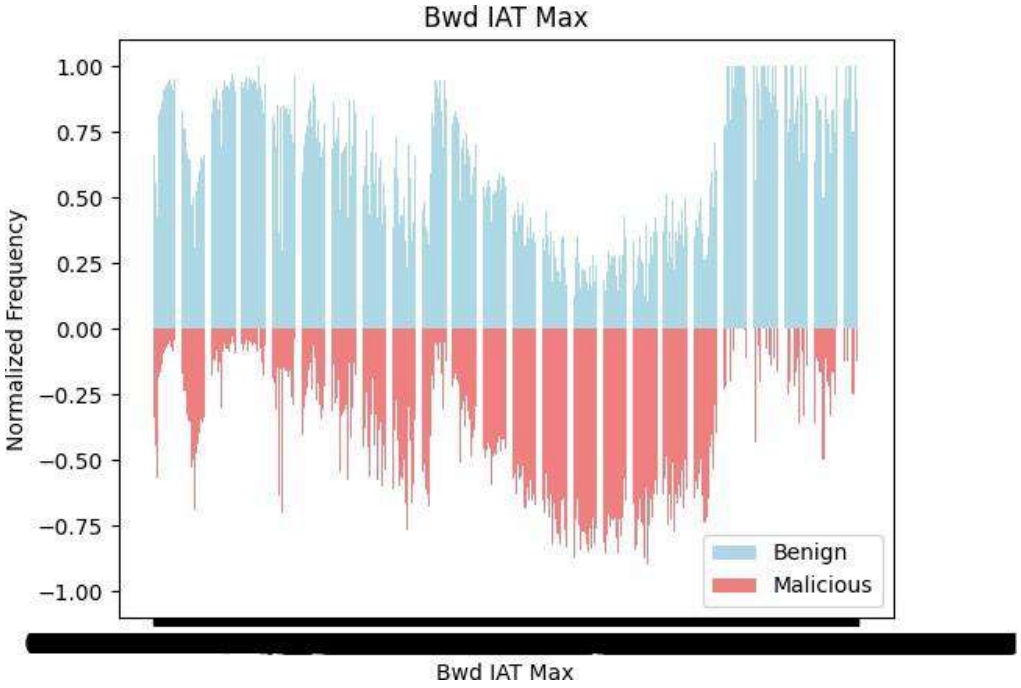


Figure 4.12.26 Pyramid chart of Bwd IAT Max w.r.t isMalicious

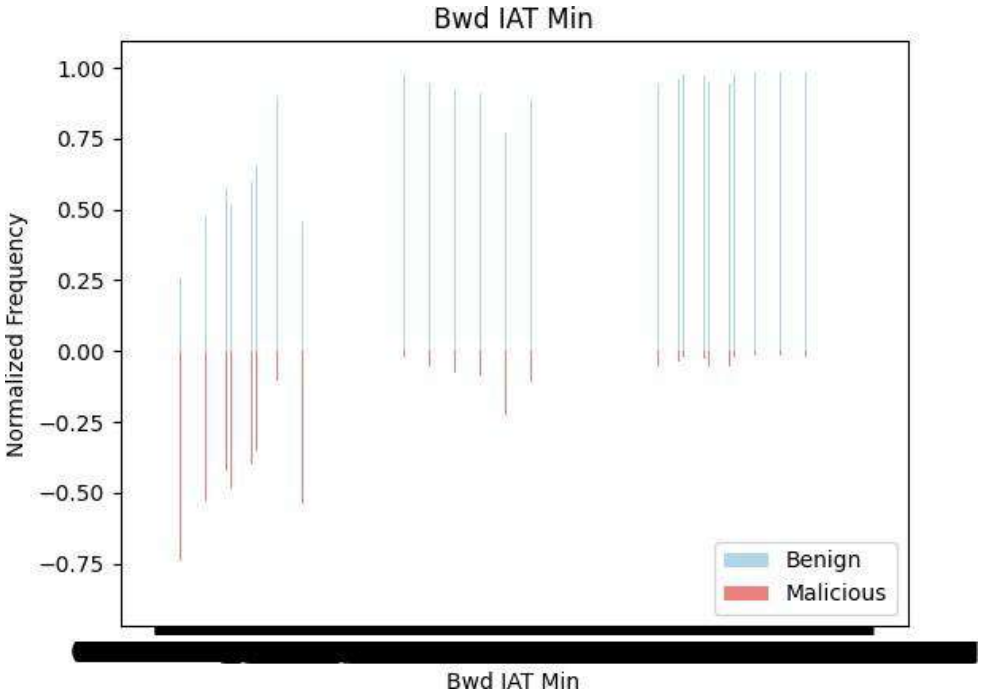


Figure 4.12.27 Pyramid chart of Bwd IAT Min w.r.t isMalicious

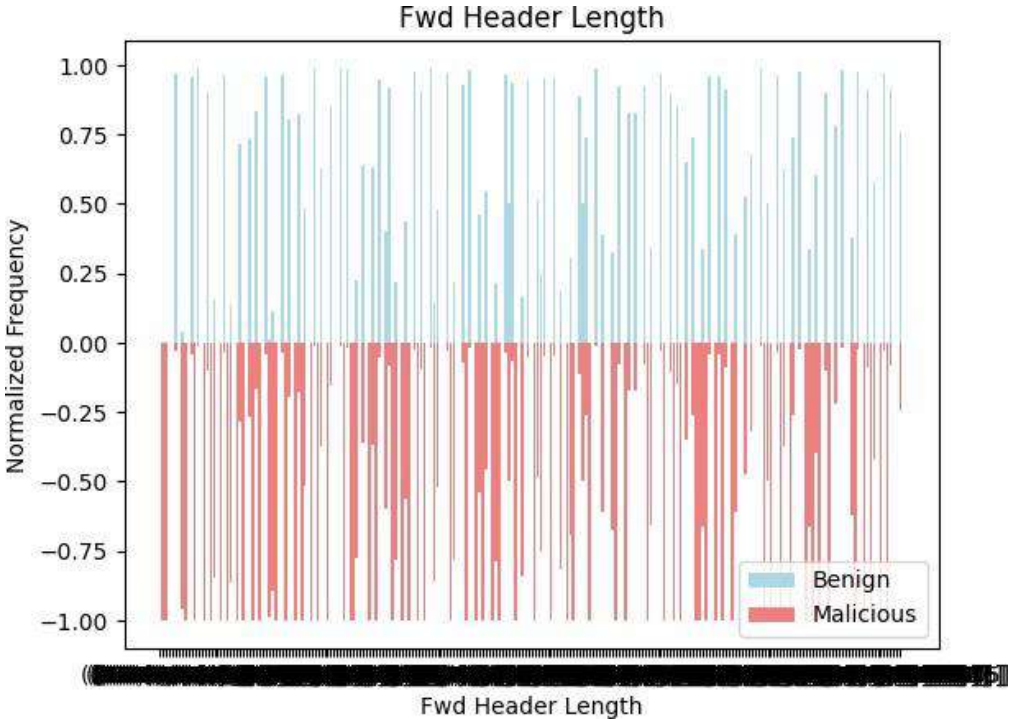


Figure 4.12.28 Pyramid chart of Fwd Header Lenngth w.r.t isMalicious

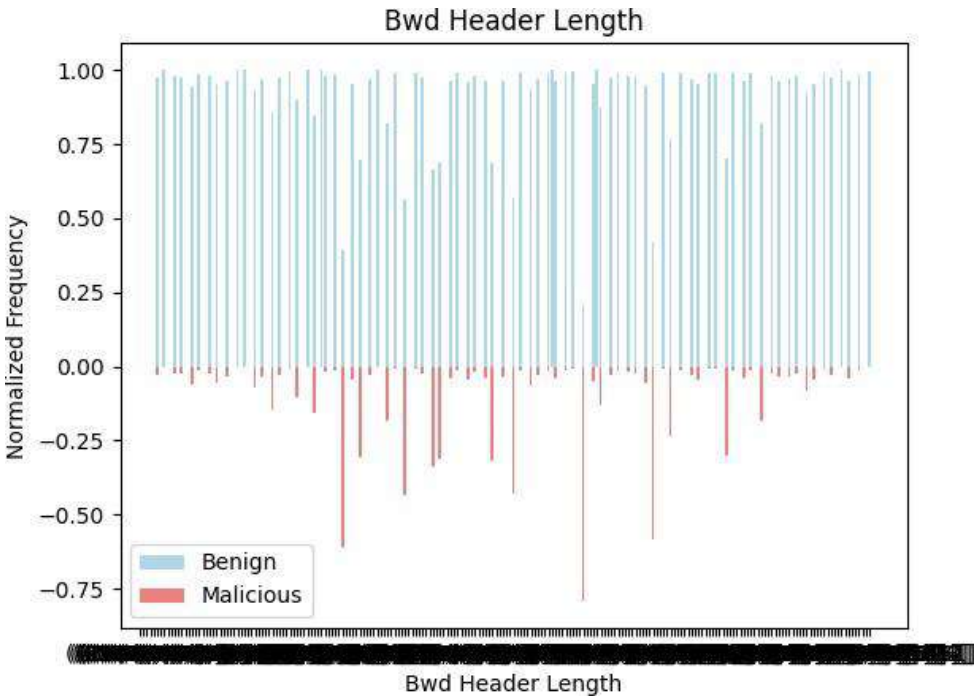


Figure 4.12.29 Pyramid chart of Bwd Header Lenngth w.r.t isMalicious

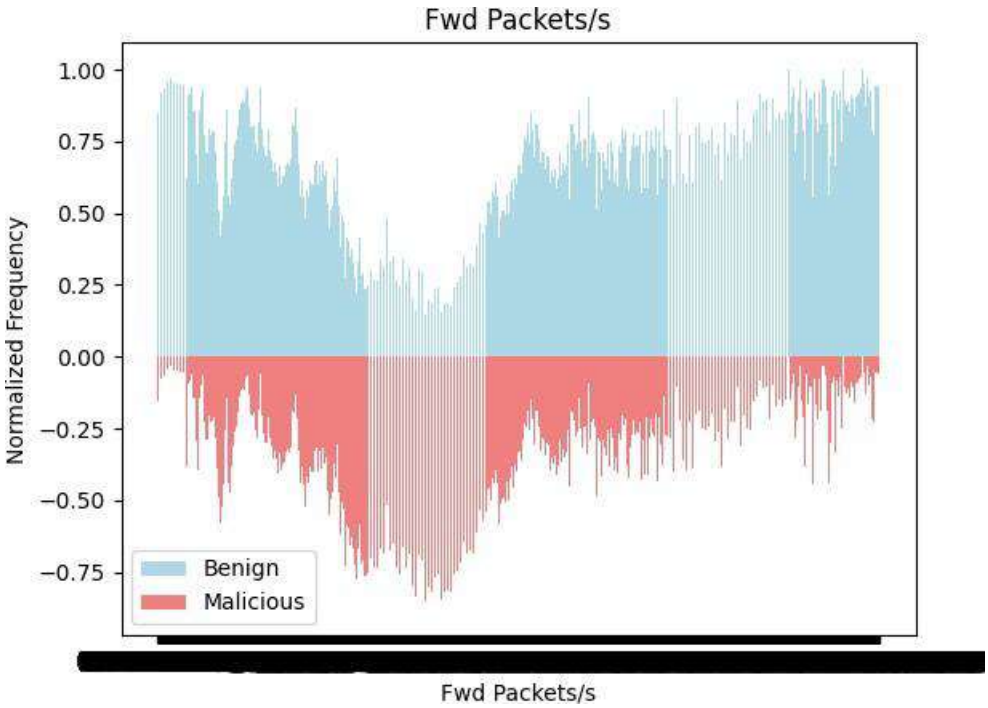


Figure 4.12.30 Pyramid chart of Fwd Packets/s w.r.t isMalicious

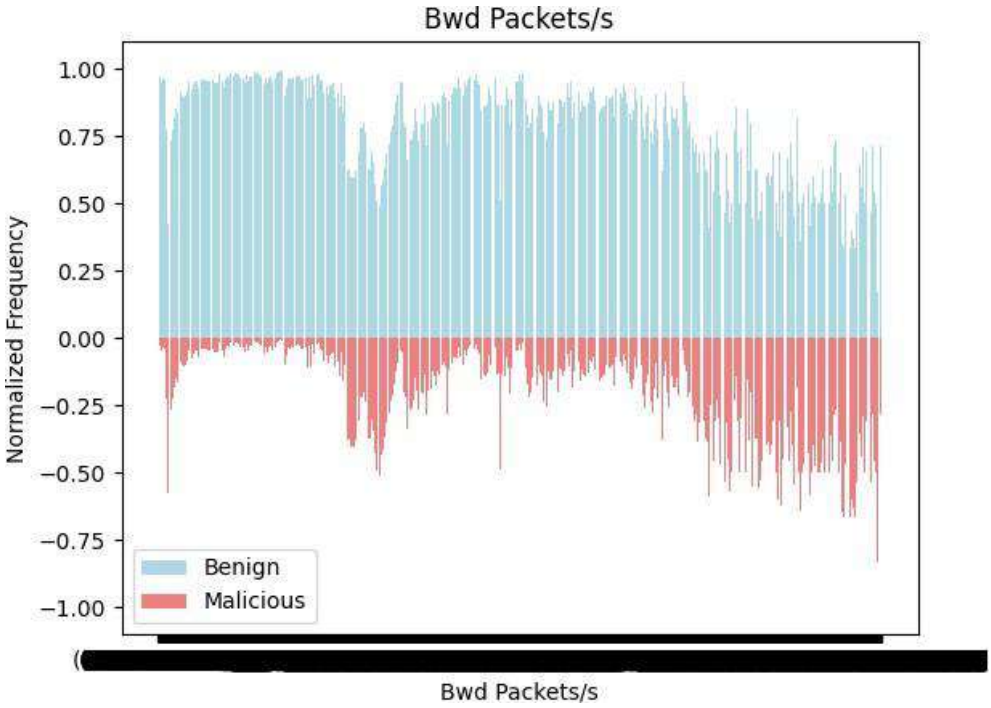


Figure 4.12.31 Pyramid chart of Bwd Packets/s w.r.t isMalicious

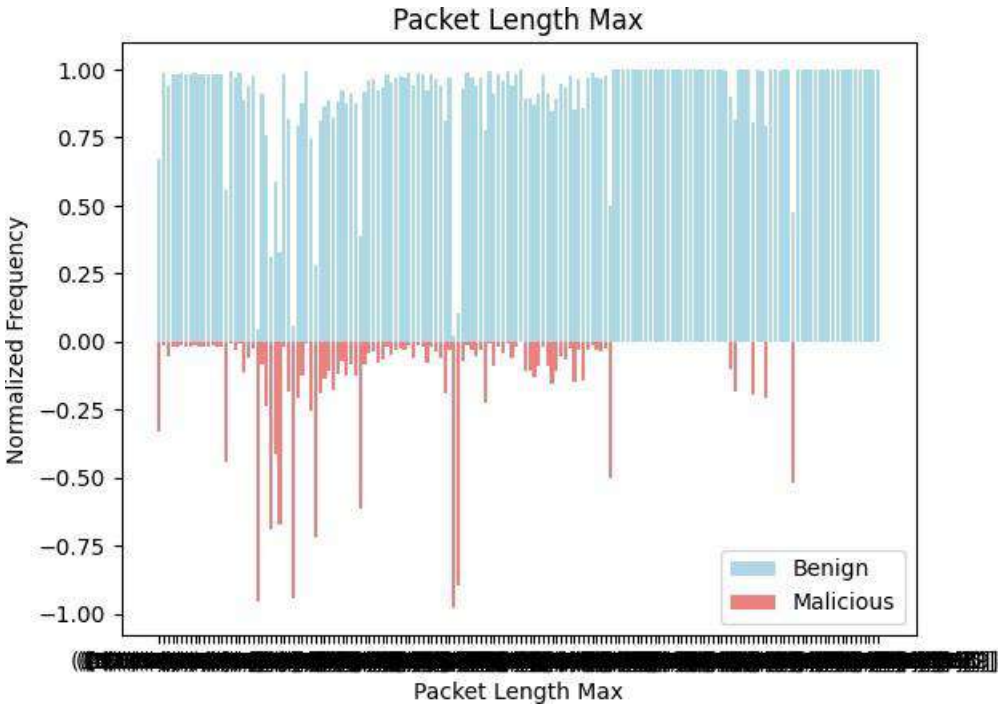
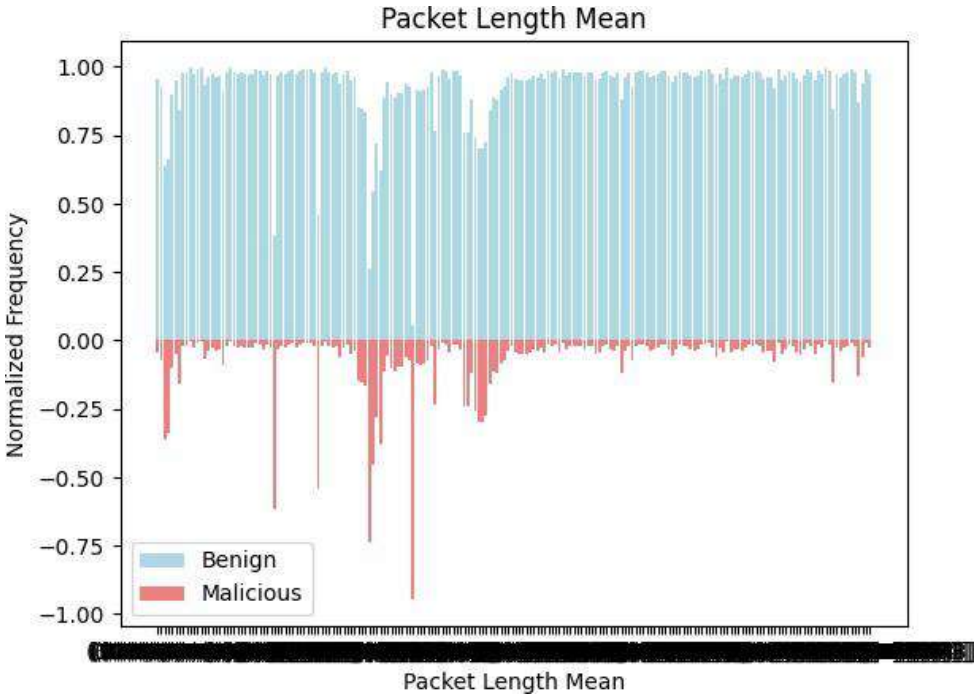


Figure 4.12.32 Pyramid chart of Packet Length Max w.r.t isMalicious



4.12.33 Pyramid chart of Packet Length Mean w.r.t isMalicious

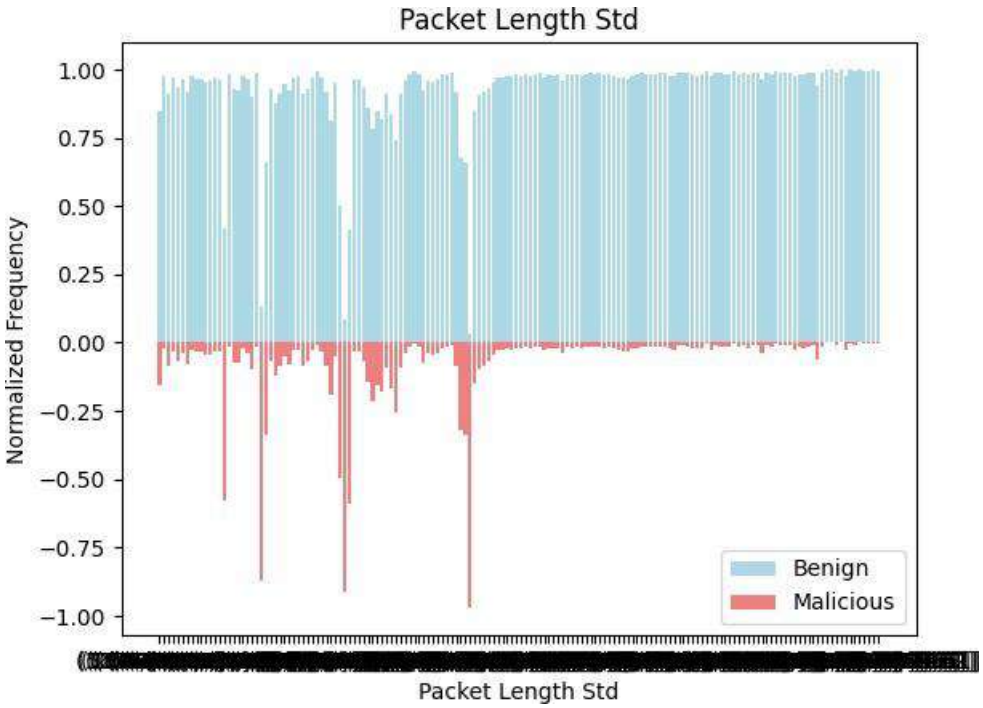


Figure 4.12.34 Pyramid chart of Packet Length Std w.r.t isMalicious

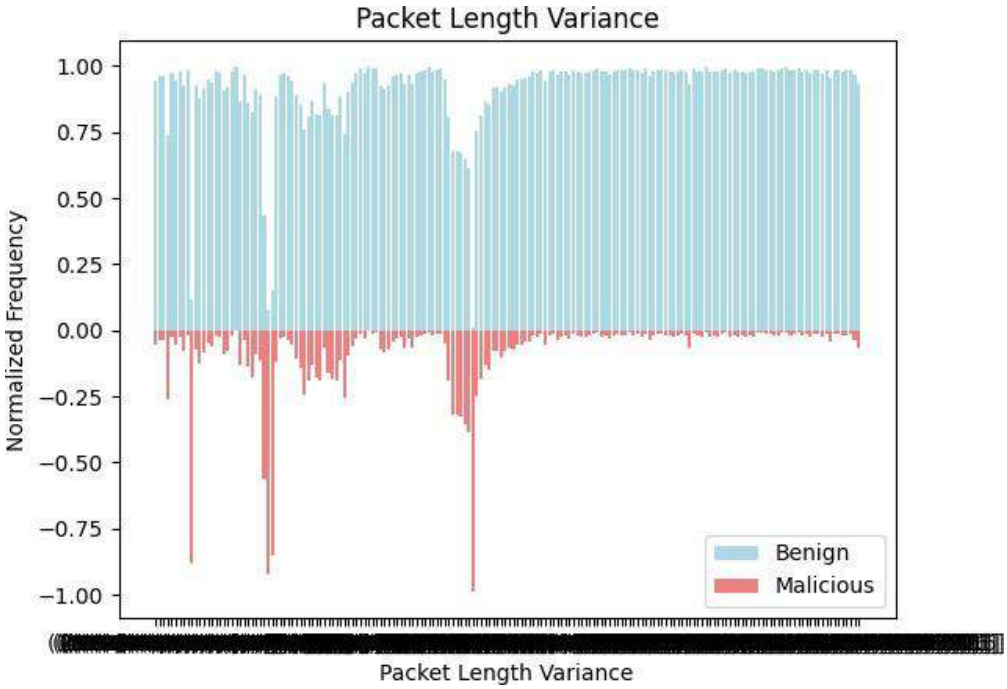


Figure 4.12.35 Pyramid chart of Packet Length Variance w.r.t isMalicious

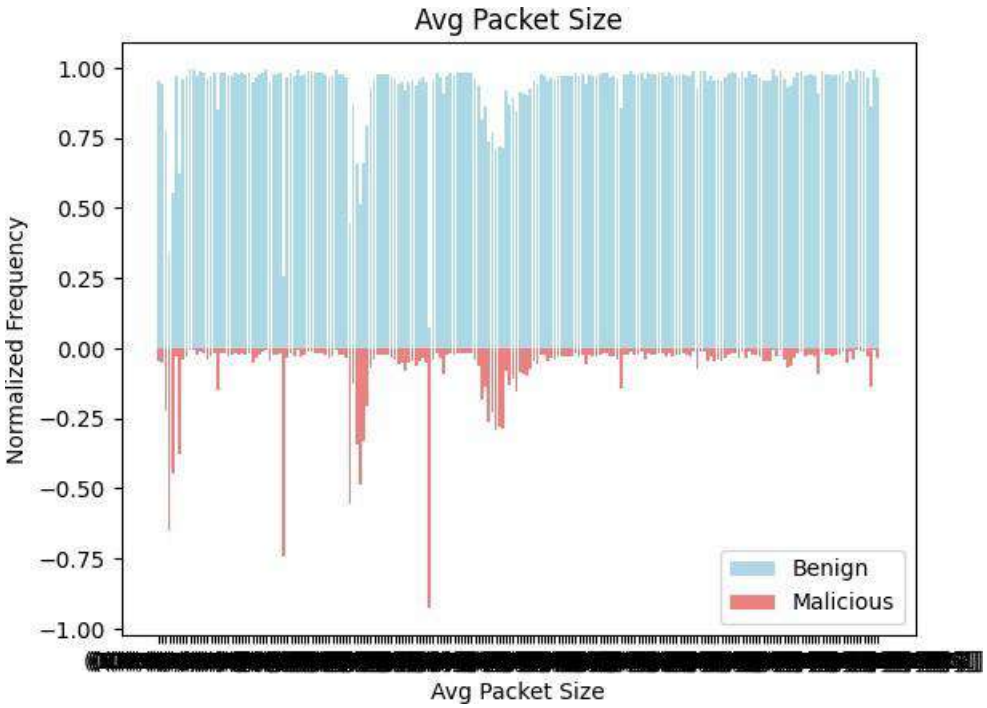


Figure 4.12.36 Pyramid chart of Avg Packet Size w.r.t isMalicious

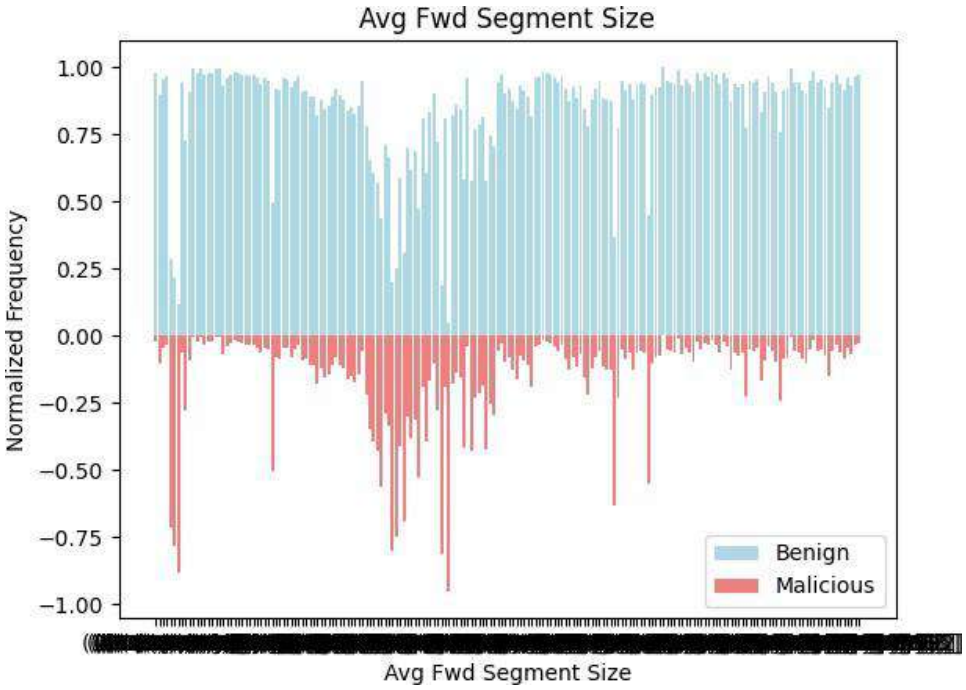


Figure 4.12.37 Pyramid chart of Avg Fwd Segment Size w.r.t isMalicious

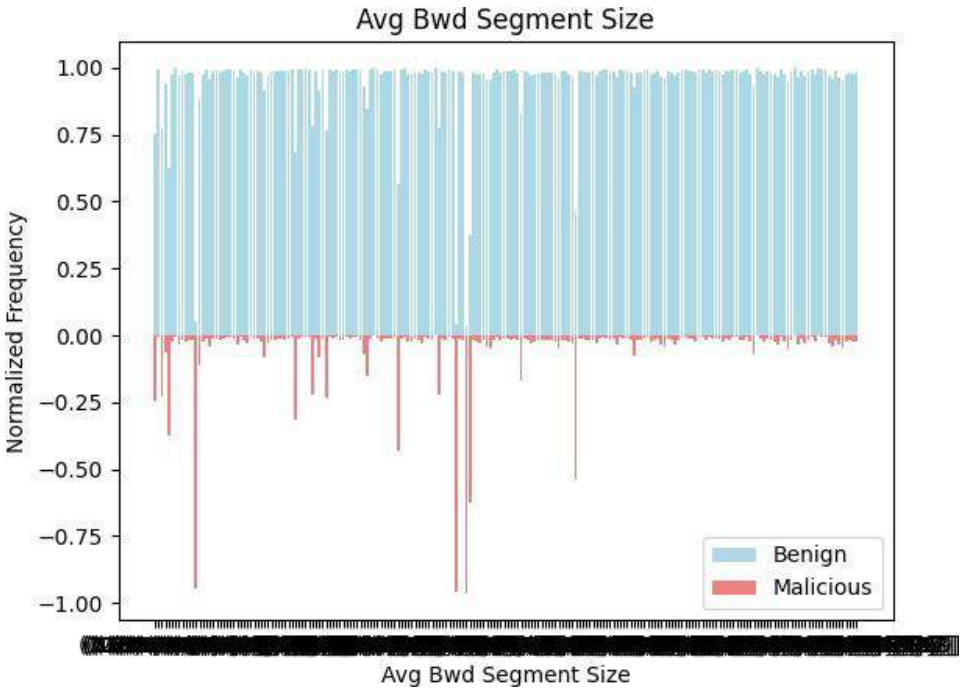


Figure 4.12.38 Pyramid chart of Avg Bwd Segment Size w.r.t isMalicious

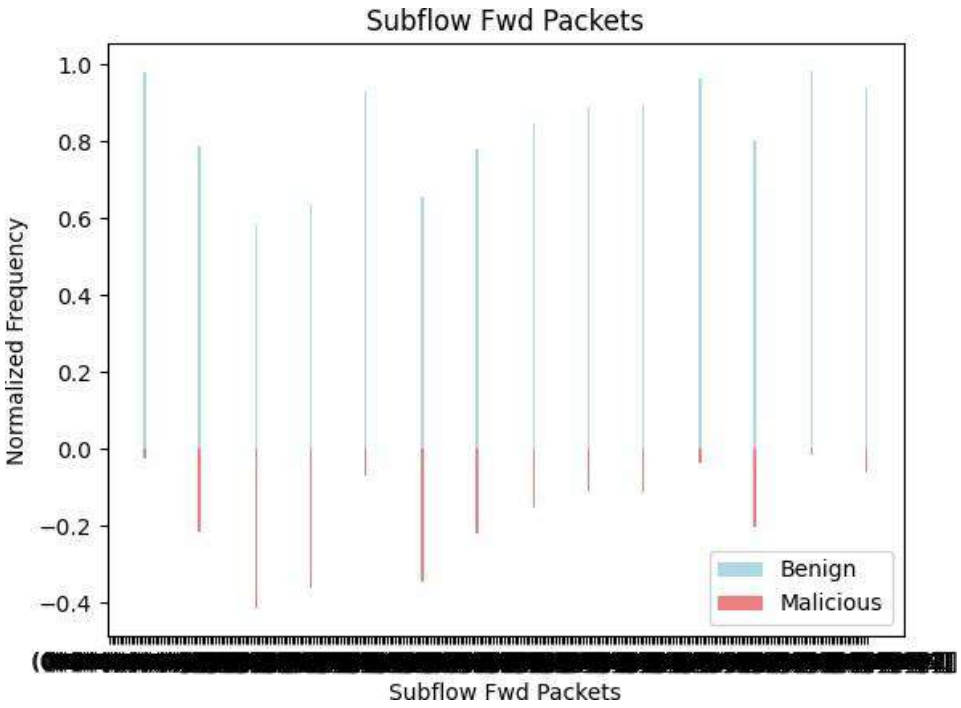


Figure 4.12.39 Pyramid chart of Subflow Fwd Packets w.r.t isMalicious

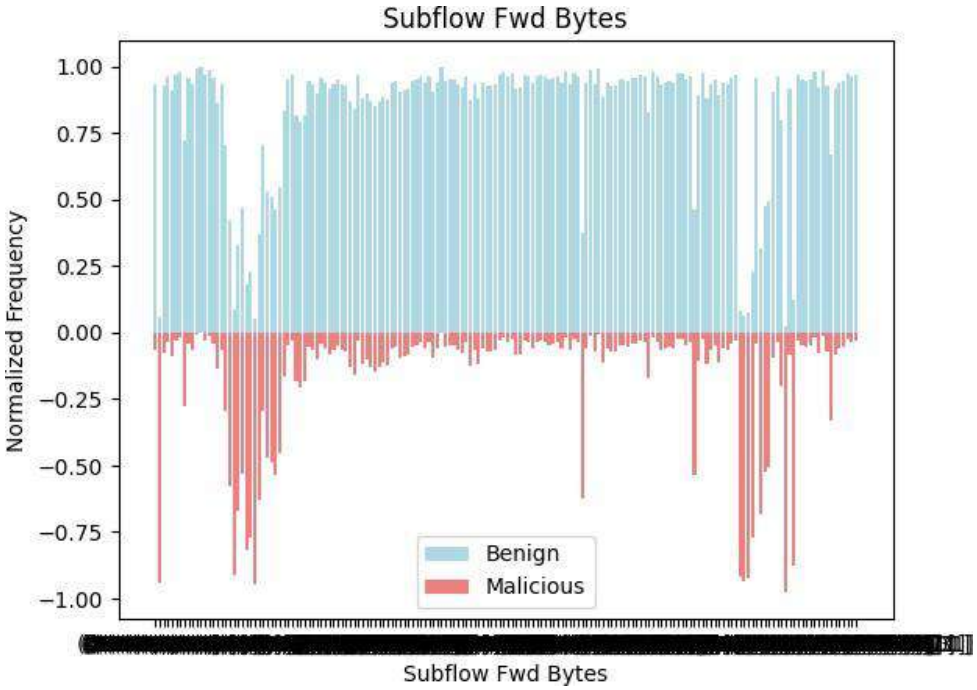


Figure 4.12.40 Pyramid chart of Subflow Fwd Bytes w.r.t isMalicious

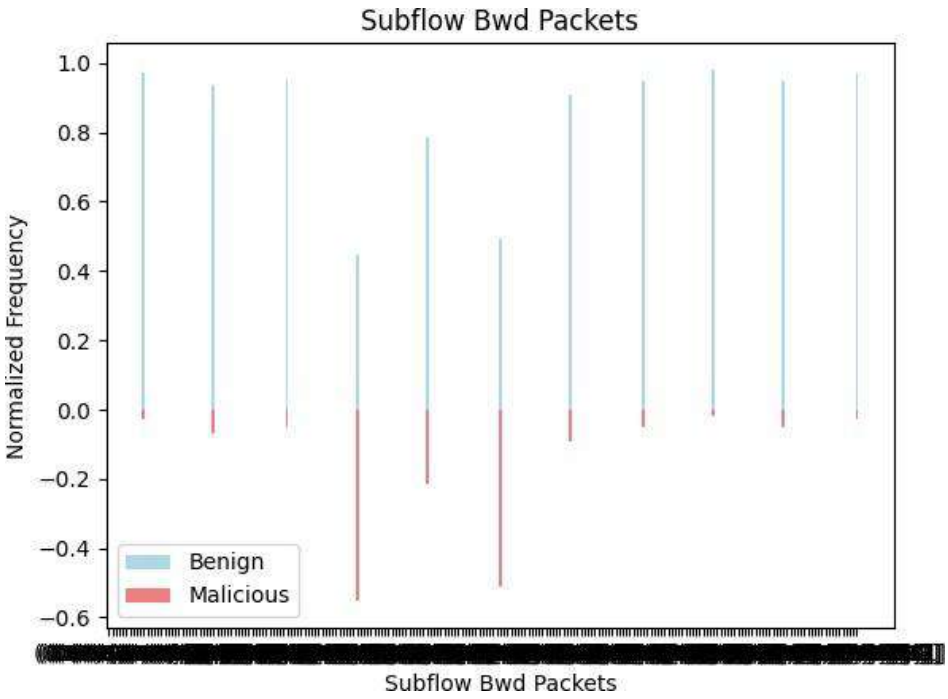


Figure 4.12.41 Pyramid chart of Subflow Bwd Packets w.r.t isMalicious

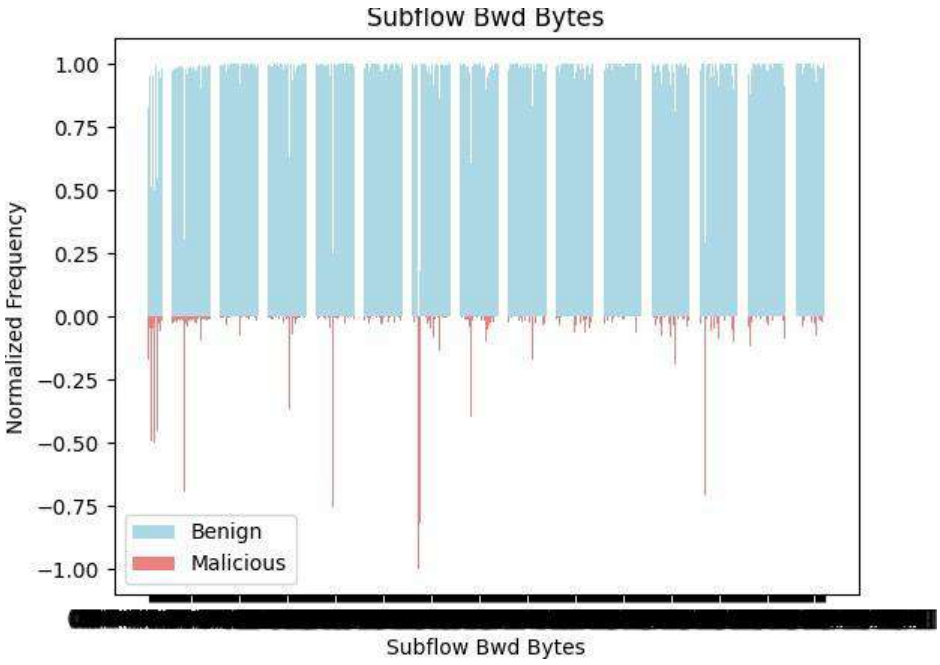


Figure 4.12.42 Pyramid chart of Subflow Bwd Bytes w.r.t isMalicious

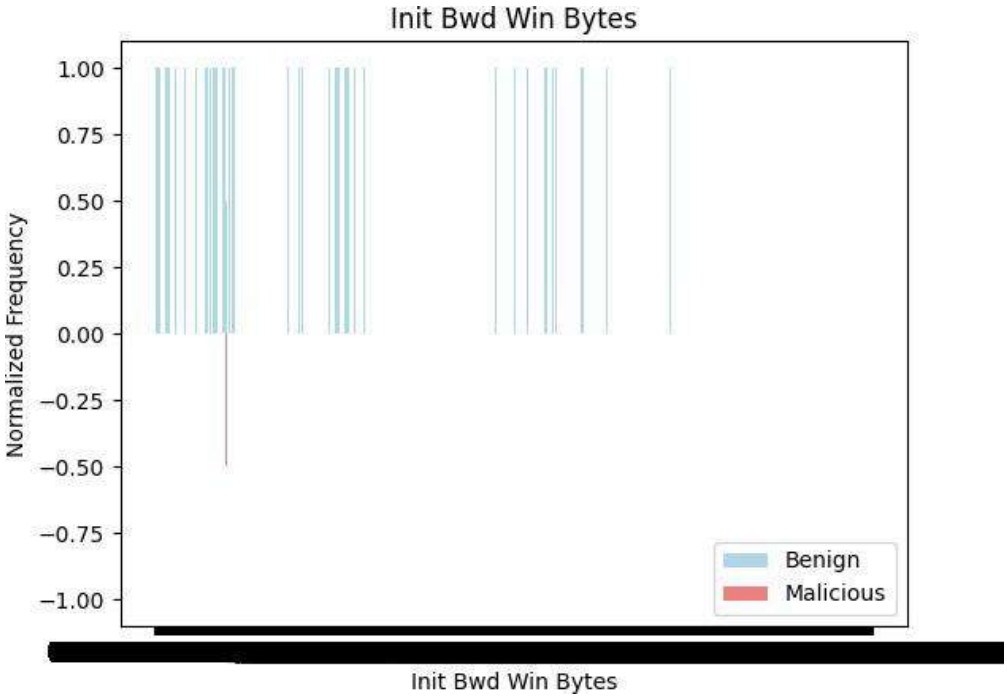


Figure 4.12.43 Pyramid chart of Init Bwd Win Bytes w.r.t isMalicious

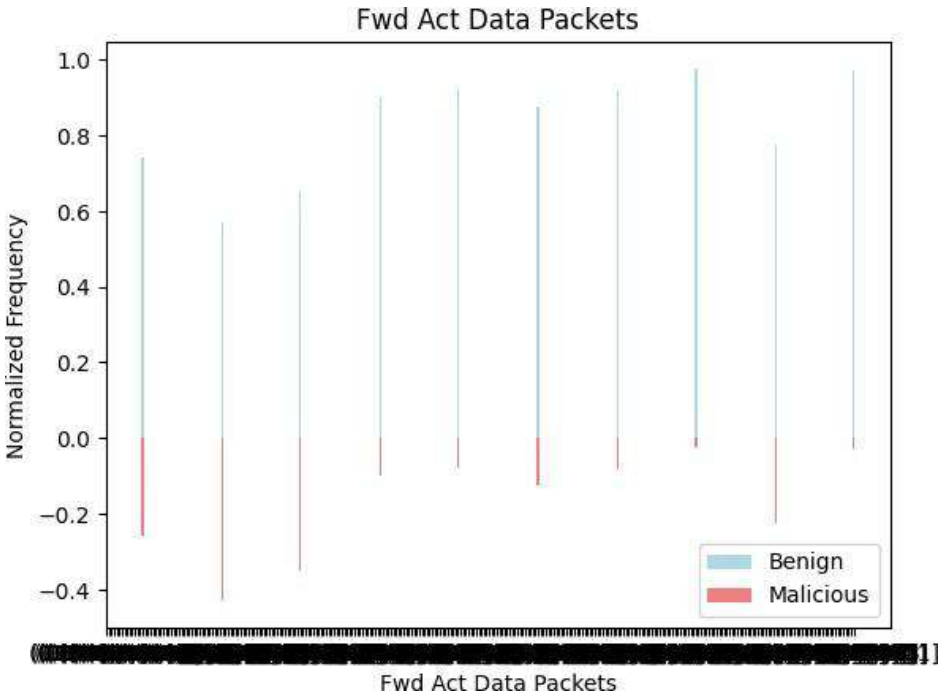


Figure 4.12.44 Pyramid chart of Fwd Act Data Packets w.r.t isMalicious

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.

4.13 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 4.13.1: Encoded values of ClassLabel

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

4.14 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 (4% of the original 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).

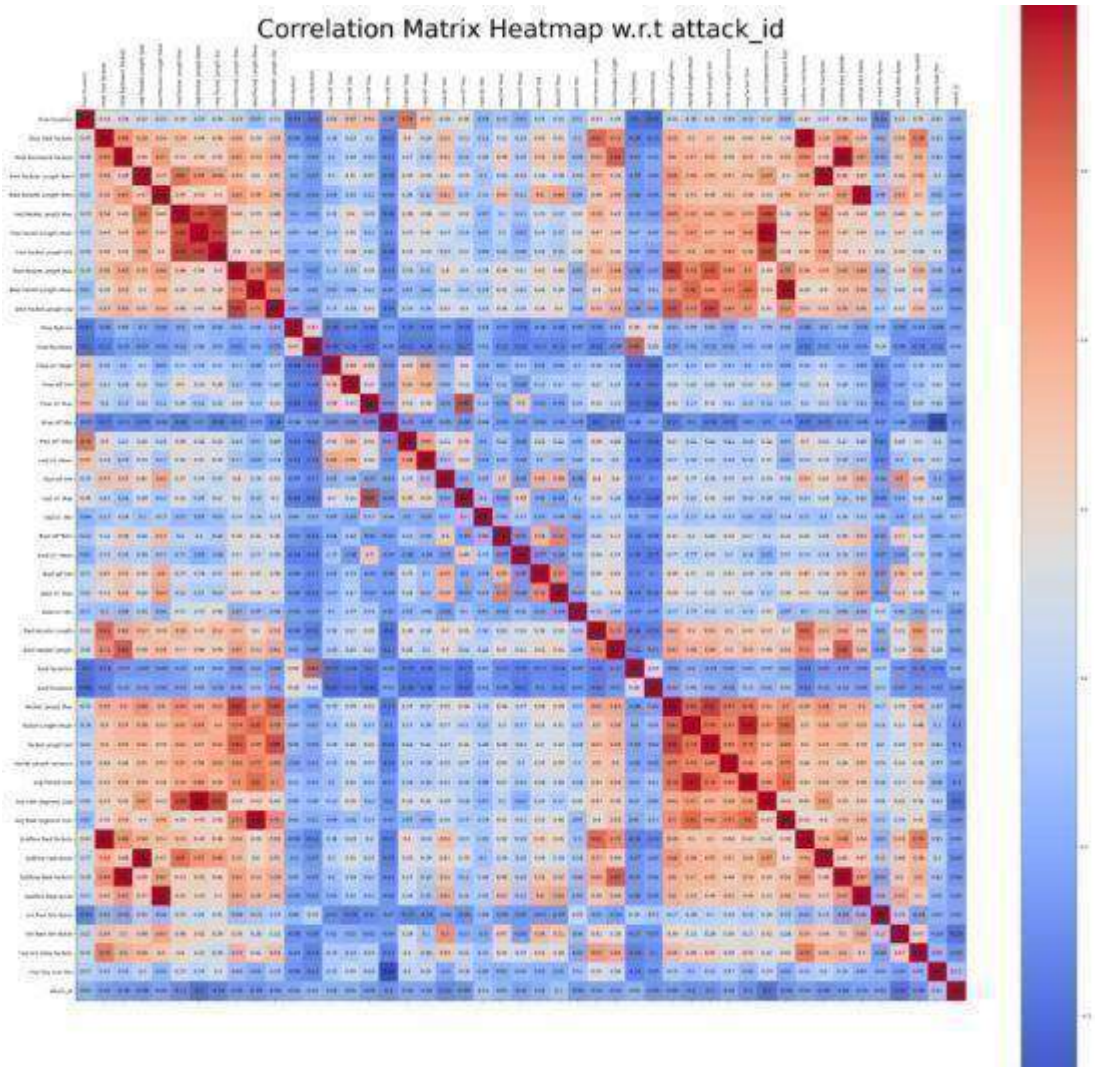


Figure 4.14.1 Correlation matrix based on 4% of the original dataset.

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: -
    a. Fwd Packet Length Max – Fwd Packet Length Mean = 0.88
    b. Fwd Packet Length Max – Fwd Packet Length Std = 0.91
    c. Bwd Packet Length Std – Packet Length Max = 0.91
    d. 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.

4.15 Renaming of feature names: -

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

4.16 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.

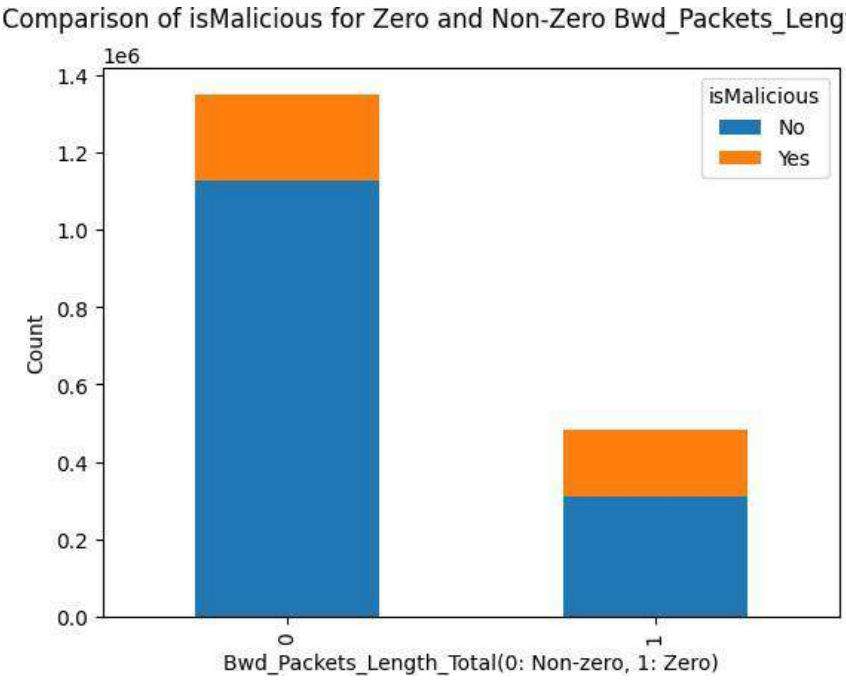


Figure 4.16.1 Stacked bar chart for Bwd Packets Length Total plotted for values which are zero and non-zero w.r.t isMalicious

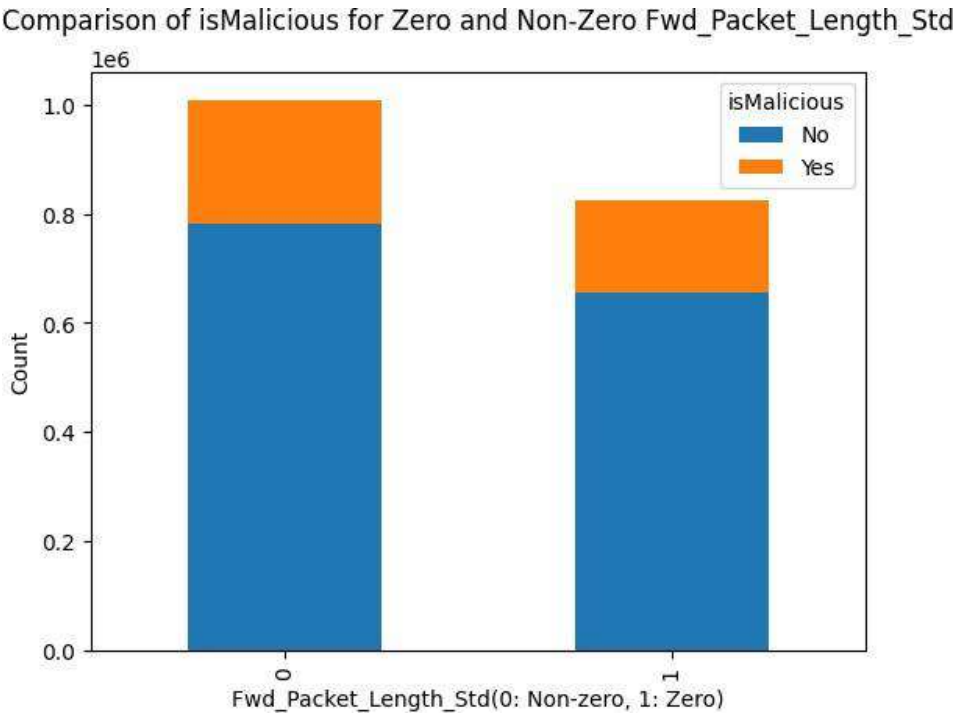


Figure 4.16.2 Stacked bar chart for Fwd Packet Length Std plotted for values which are zero and non-zero w.r.t isMalicious

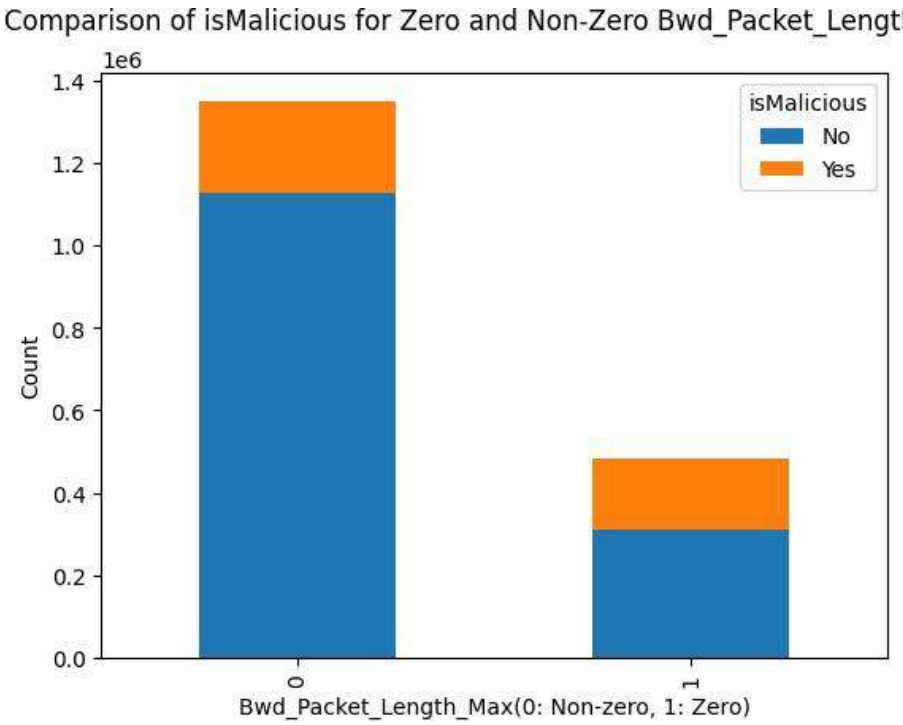


Figure 4.16.3 Stacked bar chart for Bwd Packet Length Max plotted for values which are zero and non-zero w.r.t isMalicious

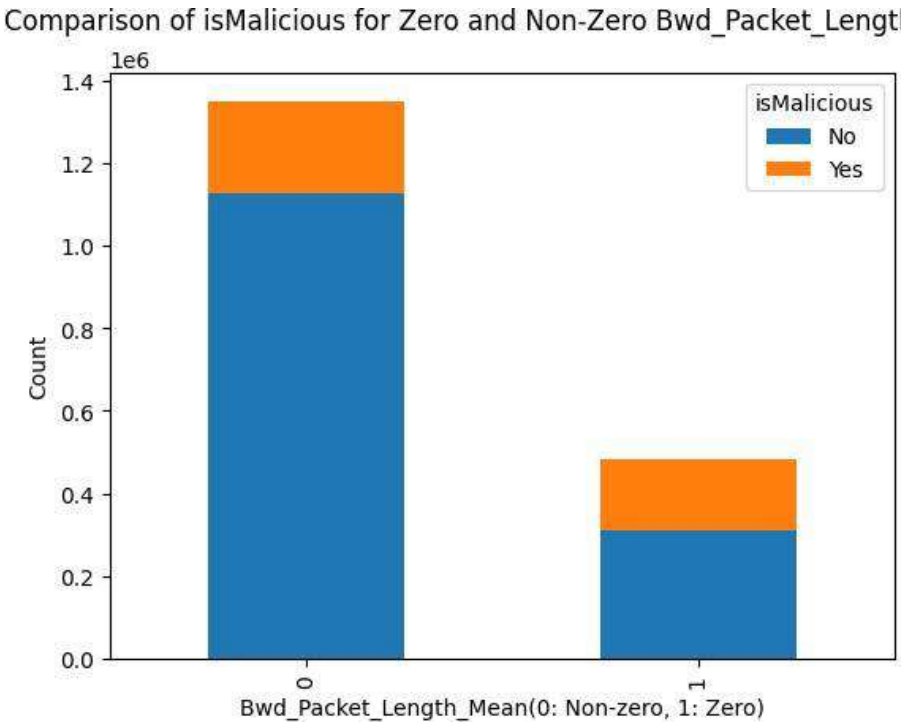


Figure 4.16.4 Stacked bar chart for Bwd Packet Length Mean plotted for values which are zero and non-zero w.r.t isMalicious

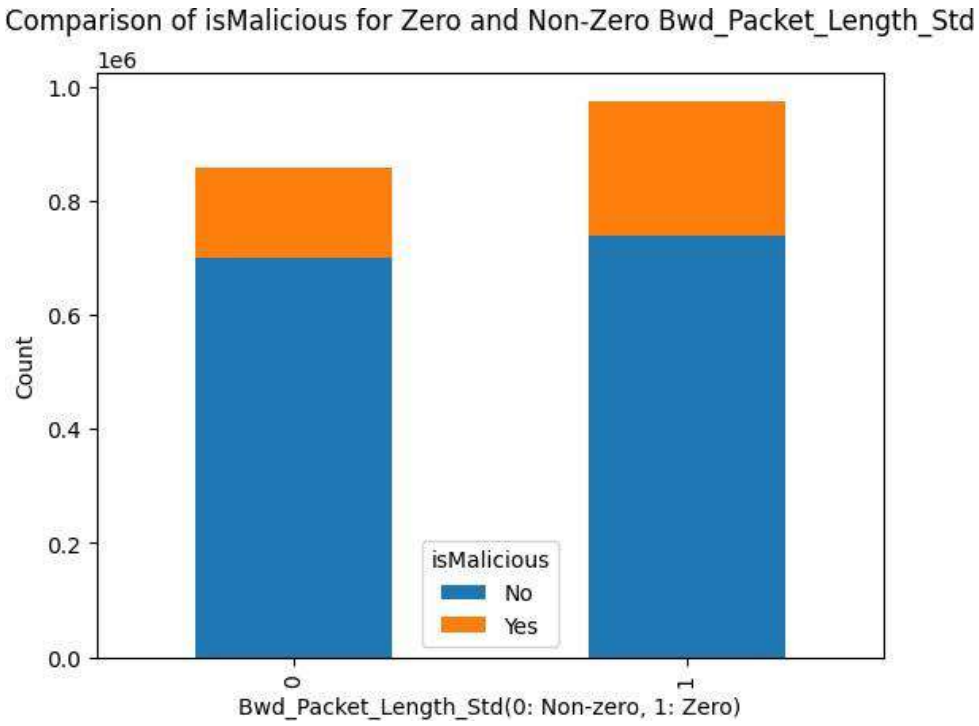


Figure 4.16.5 Stacked bar chart for Bwd Packet Length Std plotted for values which are zero and non-zero w.r.t isMalicious

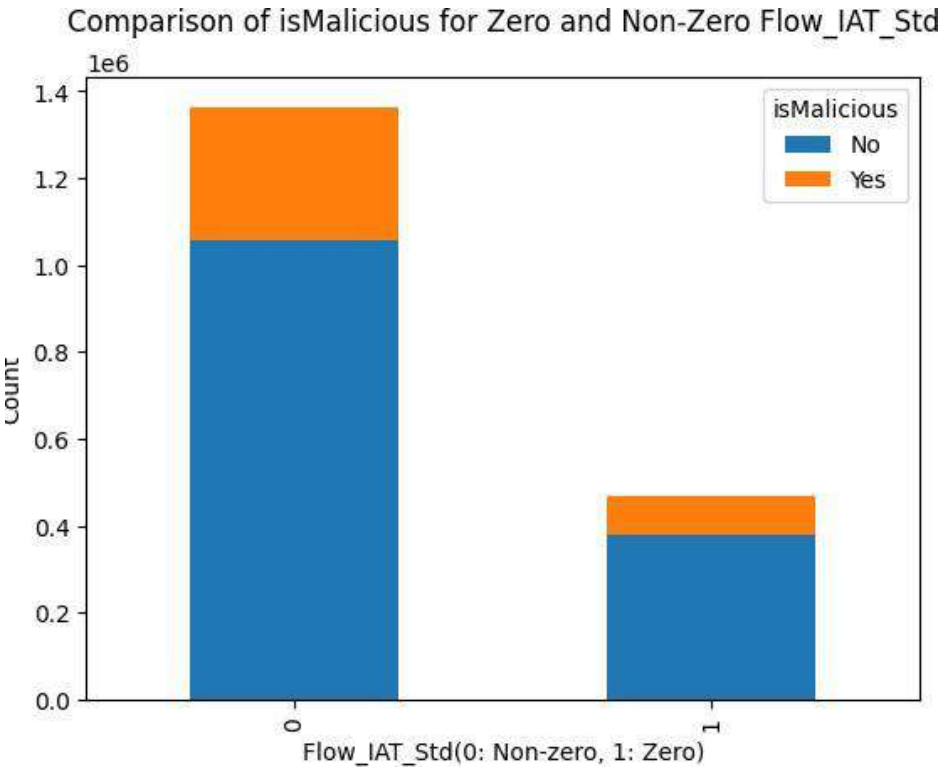


Figure 4.16.6 Stacked bar chart for Flow IAT Std plotted for values which are zero and non-zero w.r.t isMalicious

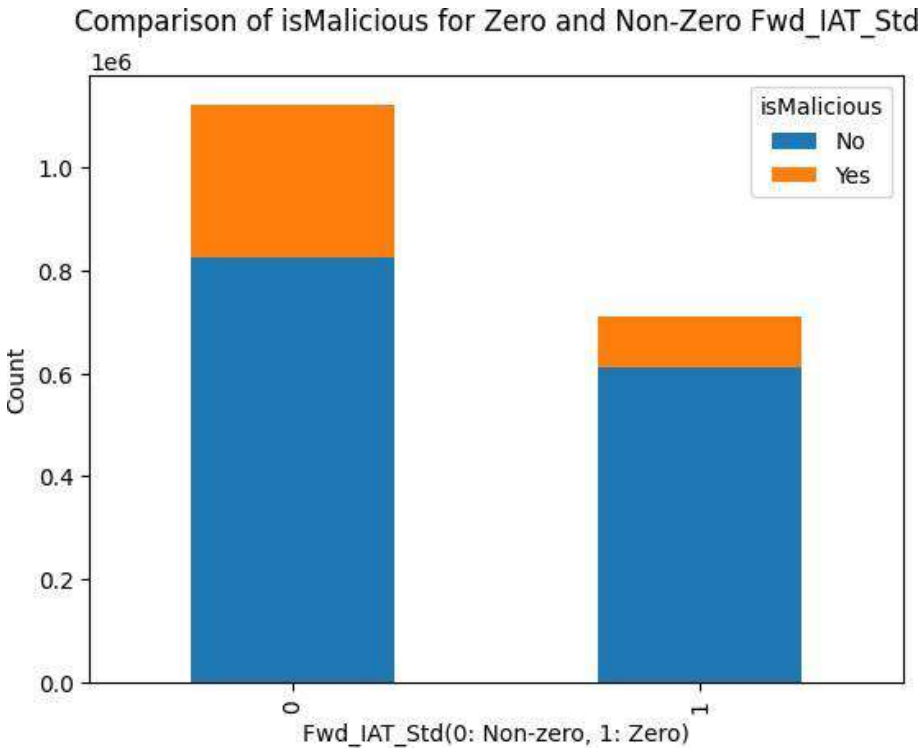


Figure 4.16.7 Stacked bar chart for Fwd IAT Std plotted for values which are zero and non-zero w.r.t isMalicious

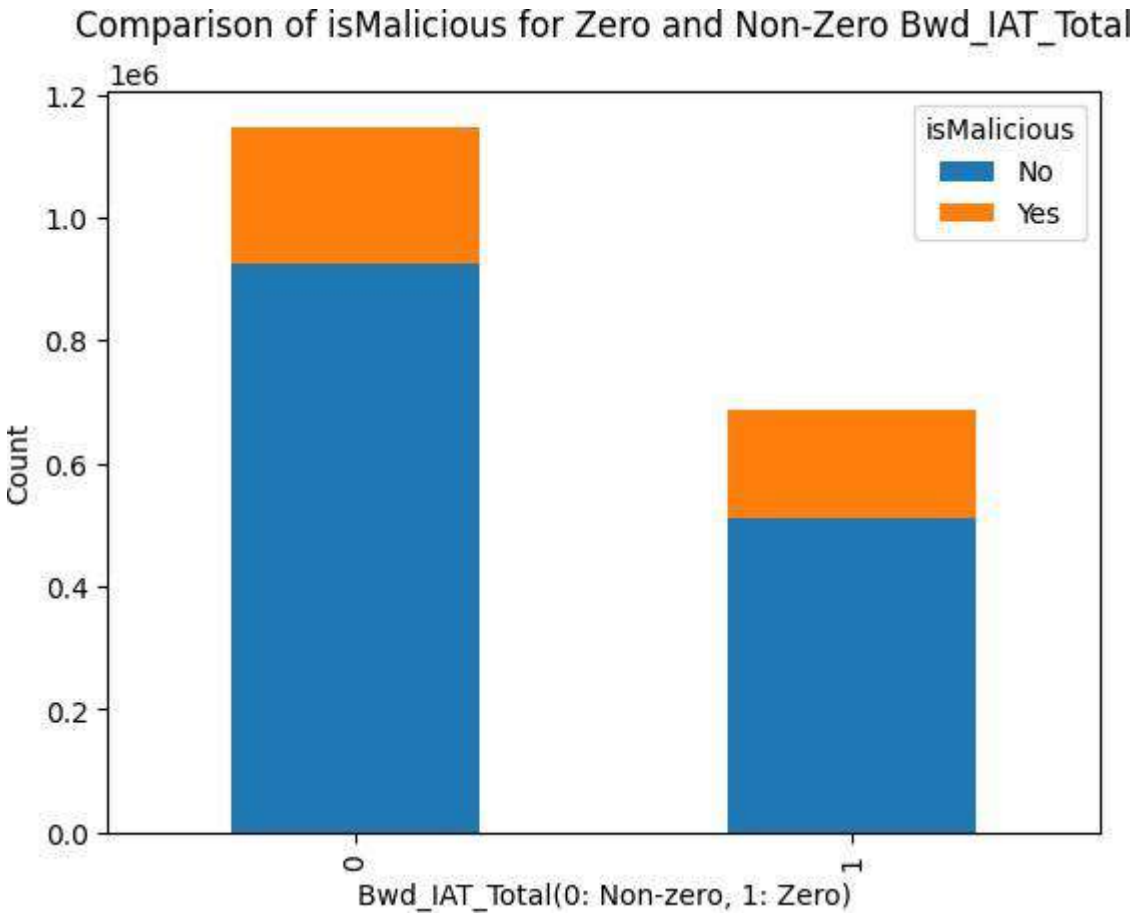


Figure 4.16.8 Stacked bar chart for Bwd IAT Total plotted for values which are zero and non-zero w.r.t isMalicious

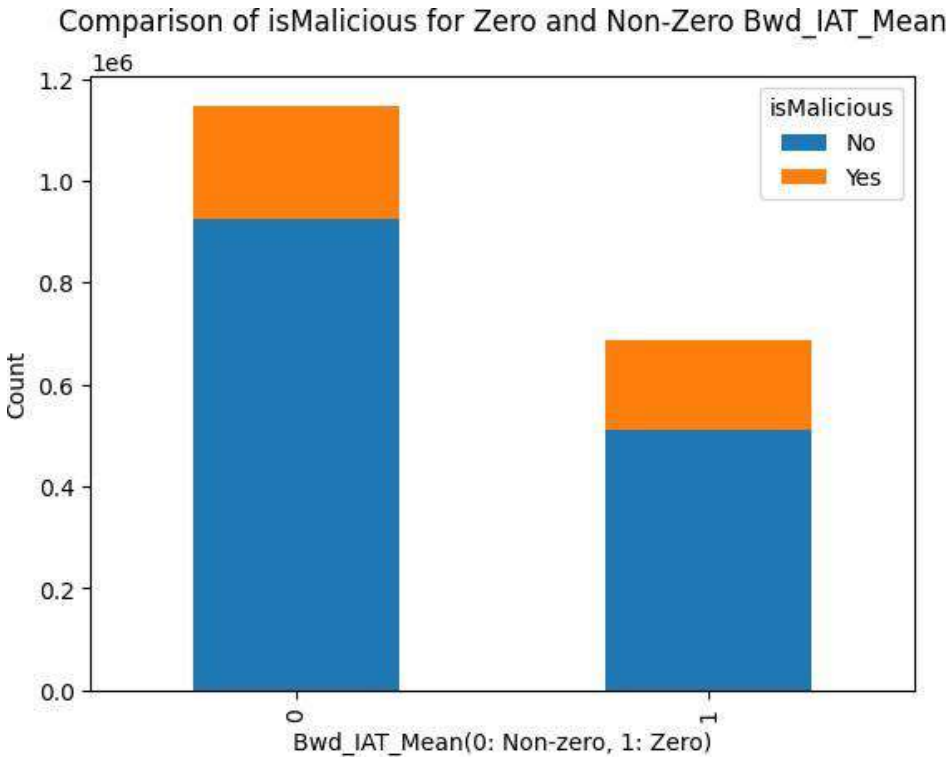


Figure 4.16.9 Stacked bar chart for Bwd IAT Mean plotted for values which are zero and non-zero w.r.t isMalicious

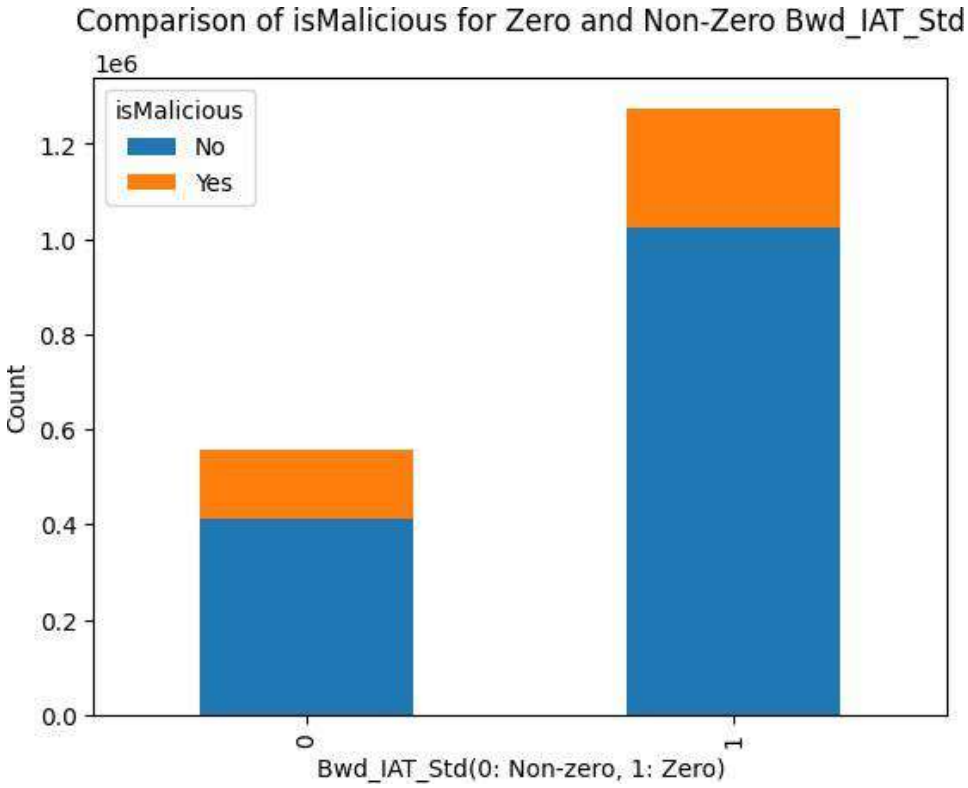


Figure 4.16.10 Stacked bar chart for Bwd IAT Std plotted for values which are zero and non-zero w.r.t isMalicious

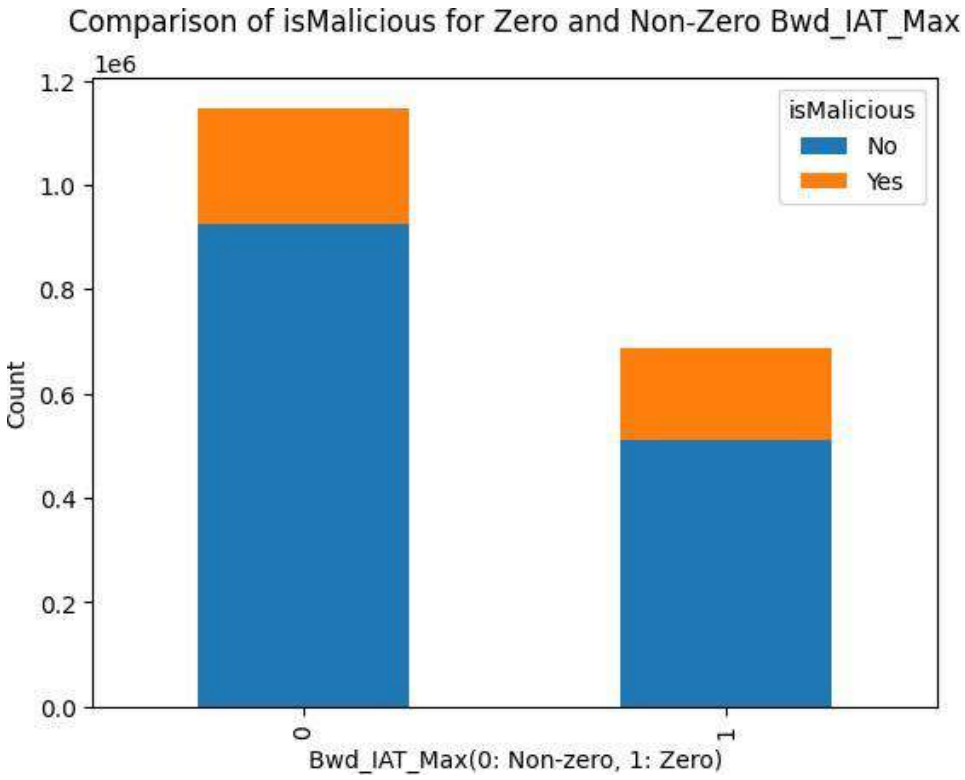


Figure 4.16.11 Stacked bar chart for Bwd IAT Max plotted for values which are zero and non-zero w.r.t isMalicious

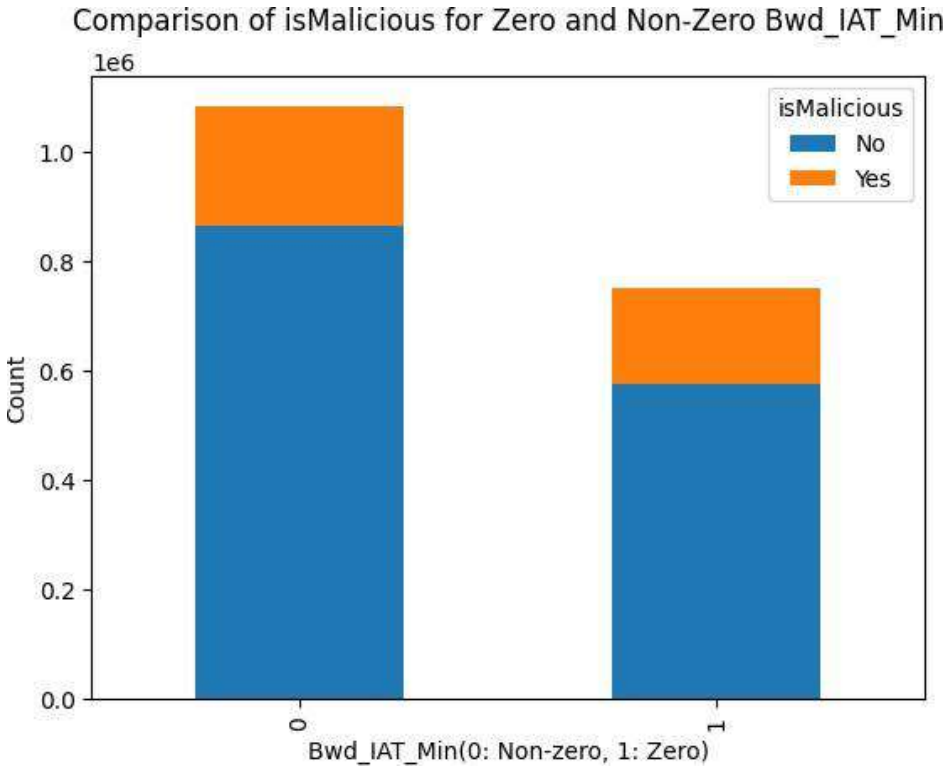


Figure 4.16.12 Stacked bar chart for Bwd IAT Min plotted for values which are zero and non-zero w.r.t isMalicious

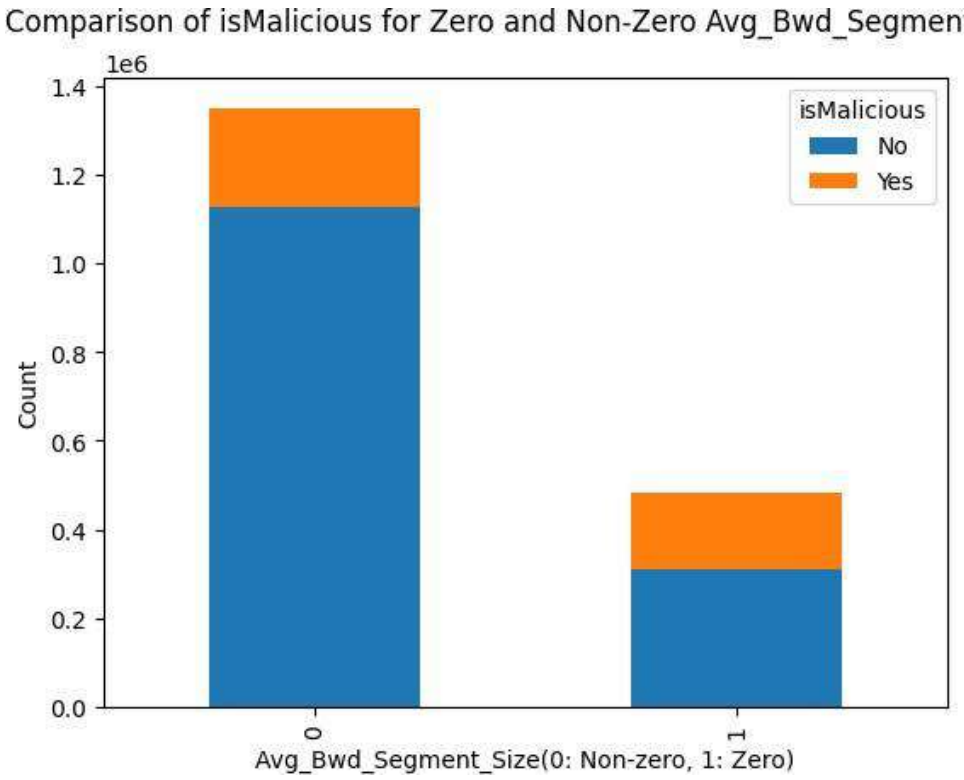


Figure 4.16.13 Stacked bar chart for Avg Bwd Segment Size plotted for values which are zero and non-zero w.r.t isMalicious

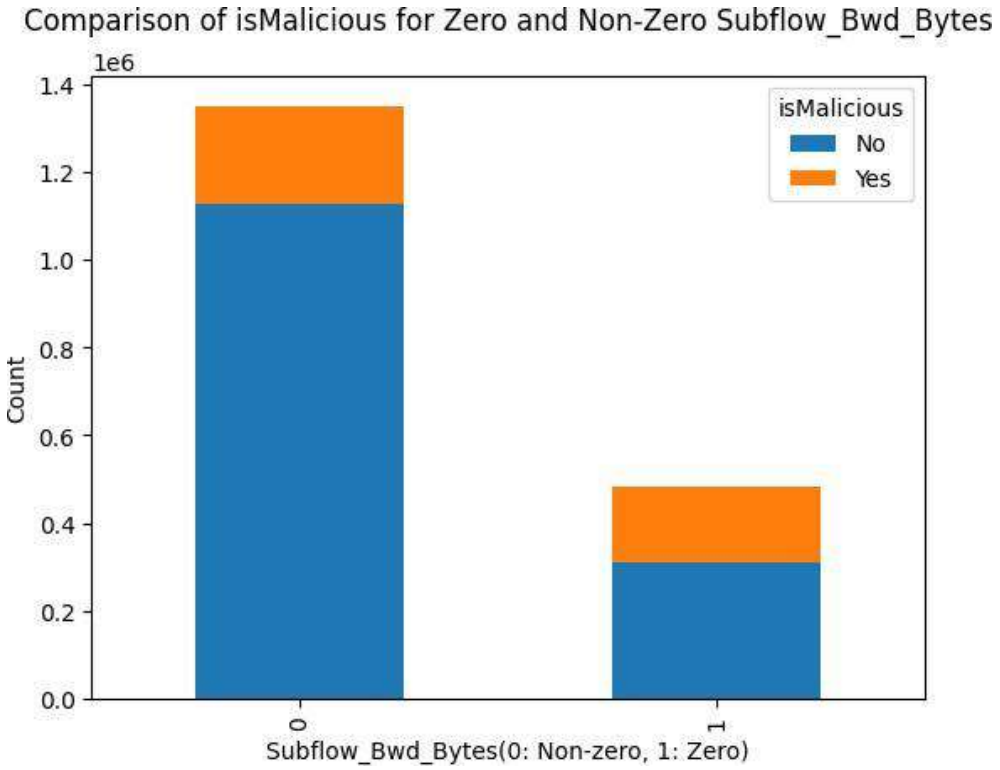


Figure 4.16.14 Stacked bar chart for Subflow Bwd Bytes plotted for values which are zero and non-zero w.r.t isMalicious

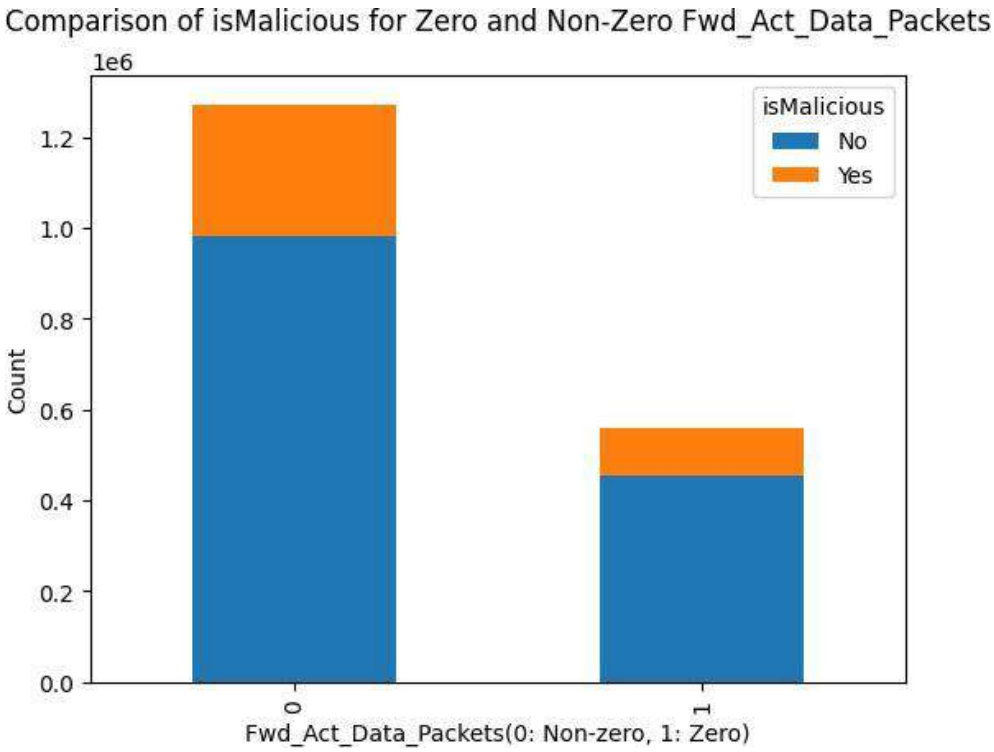


Figure 4.16.15 Stacked bar chart for Fwd Act Data Packets plotted for values which are zero and non-zero w.r.t isMalicious

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.

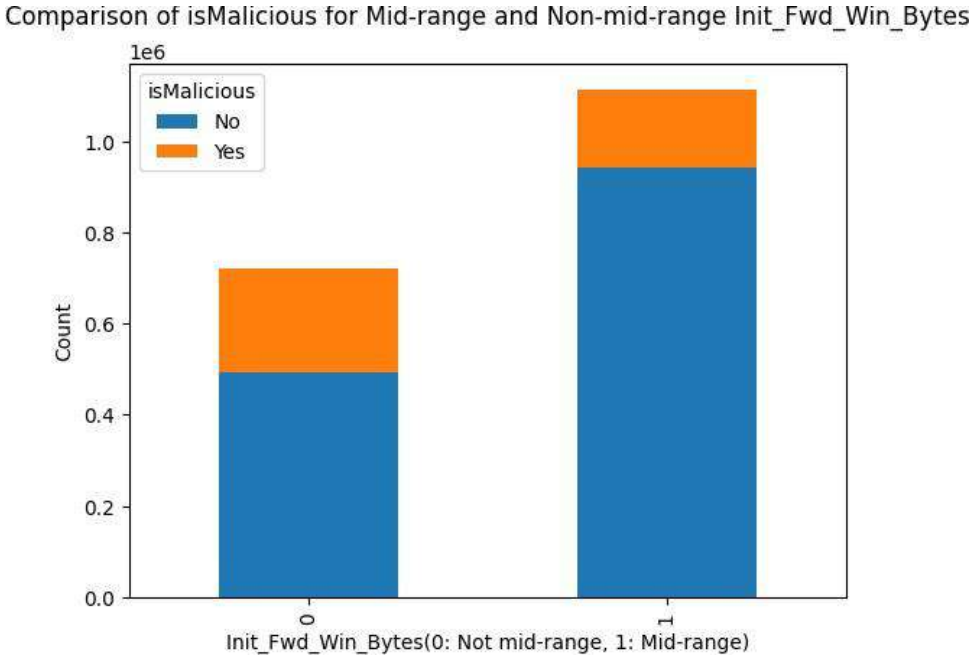


Figure 4.16.16 Stacked bar chart for Init Fwd Win Bytes plotted for values which are mid-range and not mid- range w.r.t isMalicious

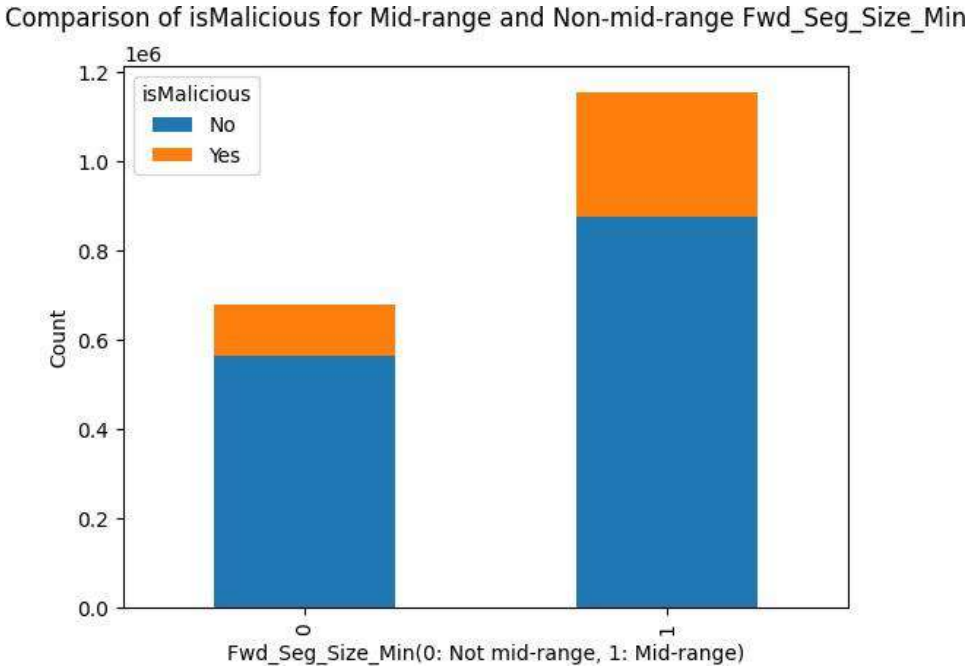


Figure 4.16.17 Stacked bar chart Fwd Seg Size Min plotted for values which are mid-range and not mid- range w.r.t isMalicious

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.

4.17 Fetching most 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.

4.18 Label encoding on newly sampled dataset: -

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 4.18.1: 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

4.19 Dropping the feature: 'Init Bwd Win Bytes': -

Based on the results of Pyramid chart, we observed 'Init Bwd Win Bytes' will not enable to train the classifier model for differentiating malicious events from benign events.

Thus, we dropped the feature from our dataset.

New shape of the sampled dataset: (1734343, 47).

4.20 Writing pre-processed data in a new file: -

The dataset was written in a new file: processed_dataaset.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.

4.21 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 4.21.1: Distribution of records in sampled dataset based on isMalicious

isMalicious	Number of records
0	1437467
1	296876

For multiclass_cic_df: -

Table 4.21.2: 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

4.22 Preparing the training data: -

- In order to overcome imbalanced nature of the dataset, we performed Undersampling.
- Undersampling was chosen for following reasons: -
    - To avoid creating duplicated data.
    - To ensure all classes have equal number of counts, thus, reducing bias in the training data.
- However, there is a major trade-off while selecting the approach. We ended up losing many records from the dataset.
- We performed undersampling based on feature: attack_id.
- After completion of undersampling, the shape of dataset is (82184, 47).
- Thus, we have 4 labels in field attack_id, each will have 20546 records.
- Two new dataframes were defined: -
    - binary_cic_df
    - multiclass_cic_df
- binary_cic_df has undersampled data without the field: attack_id.
- multiclass_cic_df has undersampled data without the field: isMalicious.
- Thus, shape of both binary_cic_df and multiclass_cic_df is (82184, 46).
- The data in binary_cic_df is written to a new file: binary_training_data.parquet
- The data in multiclass_cic_df is written to a new file: multiclass_training_data.parquet

4.23 Preparing the test data: -

- We fetch a new sampled dataset from the original dataset with different random_state=30. As the result, the sampled dataset for testing will have different set of records as compared to the dataset used for analysis, feature selection and model training.
- The size of sampled dataset is 20% of the original data.
- Shape of the dataset after drawing the sample is (1833516, 59).
- Now we perform similar data pre-processing steps as performed on the training dataset: -
- 13 duplicate entries were removed.
- All negative values in each feature were replaced with respective median values.
- A new feature was created: isMalicious, based on ClassLabel. isMalicious=1 means the event is Malicious. isMalicious=0 means the event is Benign.
- For following features, outliers were handled by performing winsorization: -
    - Init Fwd Win Bytes
    - Init Bwd Win Bytes
    - Fwd Seg Size Min
    - Bwd IAT Mean
- For the remaining features, outliers were handled my imputing them with respective median values.
- Following features were dropped because during analysis we found they had only 1 value in the complete dataset: -
    - Fwd PSH Flags
    - SYN Flag Count
    - URG Flag Count

    - Active Mean
    - Active Std
    - Active Max
    - Active Min
    - Idle Mean
    - Idle Std
    - Idle Max
    - Idle Min
    - Init Bwd Win Bytes
- The columns were renamed by replacing space with underscore (_).
- Records having following labels in the field ClassLabel are retained and rest are dropped: -
    - Benign
    - Botnet
    - Bruteforce
    - DDoS
- A new feature is defined: attack_id by performing label encoding on ClassLabel.
- Now we will build 4 test datasets: -

    - For binary classification: -
        - Balanced dataset: binary_balanced_test_data.parquet
        - Imbalanced dataset: binary_imbalanced_test_data.parquet

    - For multiclass classification: -
        - Balanced dataset: multiclass_balanced_test_data.parquet
        - Imbalanced dataset: multiclass_imbalanced_test_data.parquet

- For balanced datasets, following steps were carried out: -
    - We create a copy of original test dataset.
    - Perform undersampling based on attack_id.
    - Copy the undersampled data into two dataframes: -
        - balanced_binary_test_cic_df
        - balanced_multiclass_test_cic_df
    - Drop the column: attack_id in balanced_binary_test_cic_df.
    - Drop the column: isMalicious in balanced_multiclass_test_cic_df.

    Shape of balanced_binary_test_cic_df and balanced_multiclass_test_cic_df is (82768, 46).

- For imbalanced datasets, following steps were carried out: -
    - We create two copies of the test dataframe: -
        - imbalanced_binary_test_cic_df
        - imbalanced_multiclass_test_cic_df

    - Dropped the column: attack_id from imbalanced_binary_test_cic_df
    - Dropped the column: isMalicious from imbalanced_multiclass_test_cic_df

- Finally, the four dataframes were written in 4 files: -
    - balanced_binary_test_cic_df -> binary_balanced_test_data.parquet

    - imbalanced_binary_test_cic_df -> binary_imbalanced_test_data.parquet
    - balanced_multiclass_test_cic_df -> multiclass_balanced_test_data.parquet
    - imbalanced_multiclass_test_cic_df -> multiclass_imbalanced_test_data.parquet

- By the above approach, we will be able to classify any given event in test dataset for both binary classification and multiclass classification.
- This will enable us perform evaluation both in lab environment and real-time environment and compare how the model performs.

  Lab environment: Here we have control over the test dataset and its attributes, the class labels are balanced.

  Real-time environment: Here we replicate the scenario of real-world situation and observe how the classifier performs.

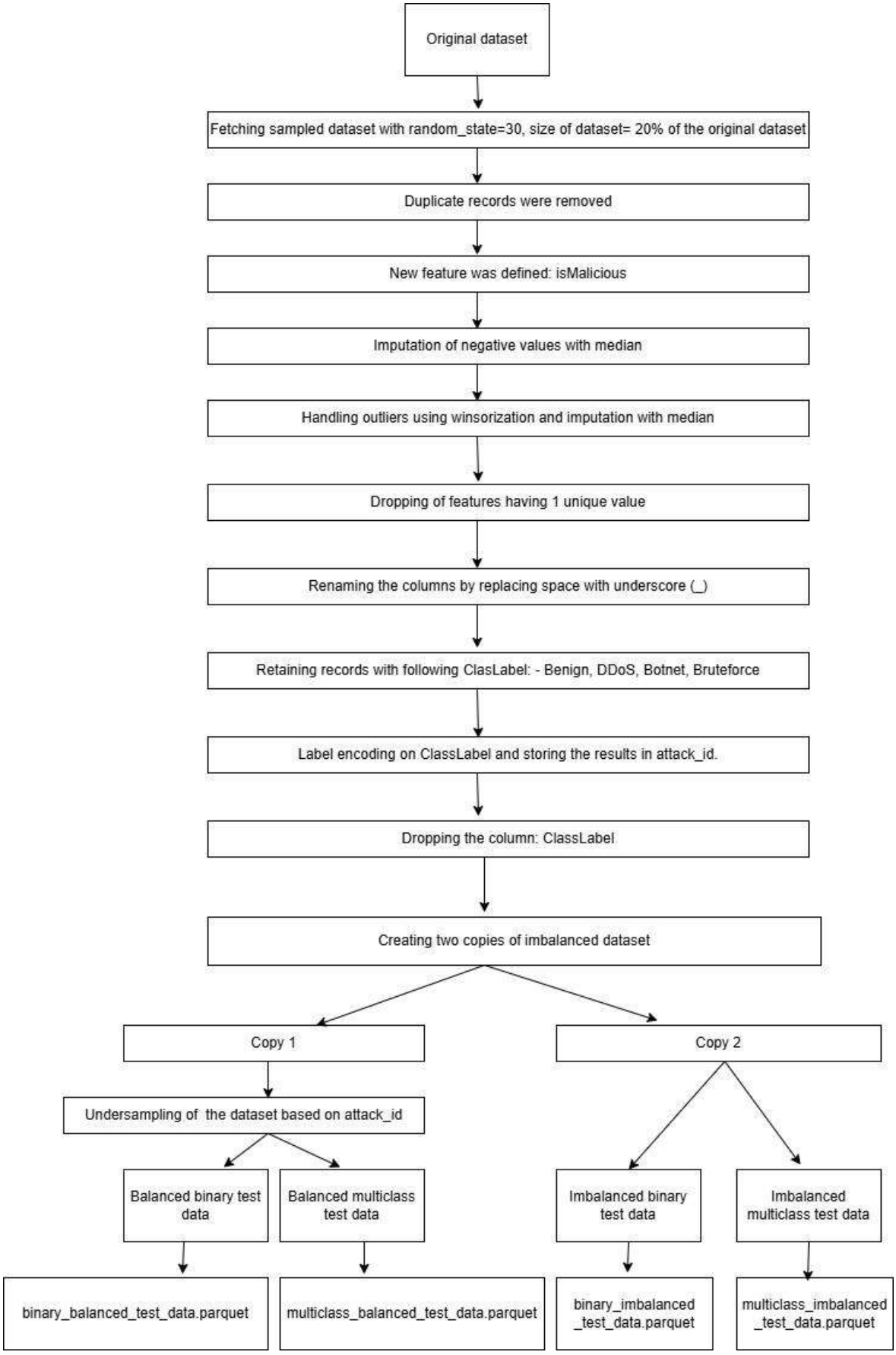


Figure 4.23.1 Flowchart to illustrate the steps for creating test datasets

Chapter 5

Research about heuristic optimization algorithms for feature selection

Heuristic algorithms involve two major components: -

1. Exploration
2. Exploitation

Exploration: -

- It is also called diversification.
- Here we generate diverse solutions which enables us to explore the search space and thus, it helps us to avoid getting stuck in local optima.
- Thus, it helps us to explore new solutions in different regions of search space and find global optima which may be far way from the current best solution discovered so far.

Exploitation: -

- It is also called intensification.
- Here we focus on a small region of the search space and exploit the information about a current solution which is better than the other solutions observed in the region.
- Thus, it helps us get the best solution present in the local search space.

Thus, heuristic algorithms have lesser likelihood than traditional approaches of getting stuck in local optima and more often return results that are global optima.

However, the success of heuristic algorithms is achieved by balancing both exploration and exploitation.

If an algorithm performs more of exploitation than exploration, we may quickly find the optimal results, but the probability of finding global optima reduces.

If an algorithm performs more of exploration than exploitation, it may become extremely slow to get optimal results, we may hope to get global optima at some point in time.

The algorithms also involve a process known as “Evolution”. In this process, the population in each generation is updated based on best solution observed, thus, enabling us to achieve more fitter individuals (subset of population) as we move from one generation to the other.

Thus, the goal to minimize the error of outcome by iterative trial and error and by using an objective function to make decisions and determine which is the best solution.

Heuristic search aims to give good solution each time it is used, however, it does not guarantee of finding the correct solution.

Thus, if we run the same algorithm on the same dataset multiple times, we may more often get different set of results, each of them having best fitness among other solutions when the search was carried out.

Heuristic algorithms are inspired from nature because the elements in nature continuously evolve based on the conditions and environment around them, enabling to tackle complex problems and also fulfilling the criteria of “Survival of the fittest” as the process continues. Moreover, since nature has evolved over many generations, the strength of population of elements gets better and closer to the

needs of current situations. In nature, constraints and conditions are many and more often non- linear, thus, the approaches to handle them become simpler but lengthy.

In Data Science and Machine Learning, we often find problems which have large search space, and constraints are non-linear and complex. As the result, taking inspiration from the processes used by nature and adapting them in our solutions helps us to leverage the power of naturally occurring processes and solve problems having large search spaces, multiple constraints and non-linear relationships.

There are more than 40 nature inspired algorithms. Some of them are: -

1. Particle Swarm Optimization
2. Artificial Bee Colony Optimization
3. Artificial Immune System
4. Ant Colony Optimization
5. Cat Swarm Optimization
6. Crow Search Optimization
7. Elephant Intelligence Behaviour
8. Grasshopper Optimization
9. Water wave Optimization
10. Brain storm Optimization
11. Whale Optimization
12. Grey Wolves Optimization
13. Insects – Firefly Optimization
14. Salp Swarm Optimization
15. Flower Pollination Algorithm
16. Bat Algorithm

Following algorithms were studied in depth in order to understand how they work and to solve the problem of feature selection: -

1. Artificial Bee Colony Optimization
2. Flower Pollination Algorithm

5.1 Artificial Bee Colony optimization: -

- It has three phases: -
    - Employed bee phase
    - Onlooker bee phase
    - Scout bee phase

- Parameters of the algorithm: -
    - Trial: Vector that keeps track of total number of failures irrespective of employed bee phase and onlooker bee phase.
    - Limit: It sets the threshold up to which a given solution can fail beyond which the solution may participate in Scout bee phase.
    - maxGenerations: Number of generations (or iterations) for which the process is carried out.

- Np: Number of food sources, which indicates number of feature subsets that can participate in a population.
- P: Random partner selected
- r: Random number
- f: Objective function
- fit: Fitness function

- Employed bee phase: -
  - Each employed bee is tagged to a food source.
  - Food source means a possible solution (In our case a subset of features).
  - From that food source its fitness value was initially computed.
  - Now, the employed bee generates a partner solution in the neighbourhood of that food source and compute its fitness value.
  - If the fitness value of the new food source is better than the current food source, then the new food source is added to the population and the original food source is removed from the population.
  - Thus, in Employed bee phase, Greedy selection is performed to determine the best solution.
  - And every bee tagged with a food source performs this process.
  - If the new food source is inferior to the current food source, then we increment the trial counter of current food source by 1.
  - If the new food source is superior than the current food source, then the trial counter of the new food source is 0.

- Onlooker bee phase: -
  - Prior to this process, we compute probability of each food source using the Equation (2).
  - A food source with higher fitness value will have higher probability.
  - For each food source, a new random number: r is generated between 0 and 1.
  - If the probability of a given food source is greater than r, then the food source is added to onlooker bee phase.
  - If the probability of a given food source is smaller than r, then the food source is not added to onlooker bee phase.
  - It may be possible that a given food source may not participate in the onlooker bee phase in a given generation. But it may participate in the onlooker bee phase in the next generation. This will depend on the random number generated for that food source in each generation.
  - For the food sources that participate in onlooker bee phase, they perform steps similar to employed bee phase.

- Scout bee phase: -
  - Solution which have trial greater than limit are the candidates to be discarded. Thus, if the trial value is greater than limit the solution can potentially enter the scout phase.
  - The trial counter of the abandoned solution is reset to 0.
  - Out of all solutions in the population which have trial counter greater than the limit, only one among them can participate in the Scout bee phase.

    - However, it is possible that we end up eliminating the best solution from the population due to the limit.
    - Thus, prior performing Scout bee phase, we need to memorize the best solution in the population.
    - in Scout bee phase: -
        - Case 1: In a given population, all the solutions have trial lesser than the limit. Thus, none of the solutions will participate in scout bee phase.
        - Case 2: In a given population, multiple solutions have trial greater than the limit. Thus, the solution with highest trial value is selected for scout bee phase.
        - Case 3: In a given solution, multiple solutions have trial greater than the limit, and have same value. Thus, randomly one of them are selected for scout bee phase.
    - Fitness is related to objective function using Equation (3).
- 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.

Flowchart of Artificial Bee Colony Optimization: -

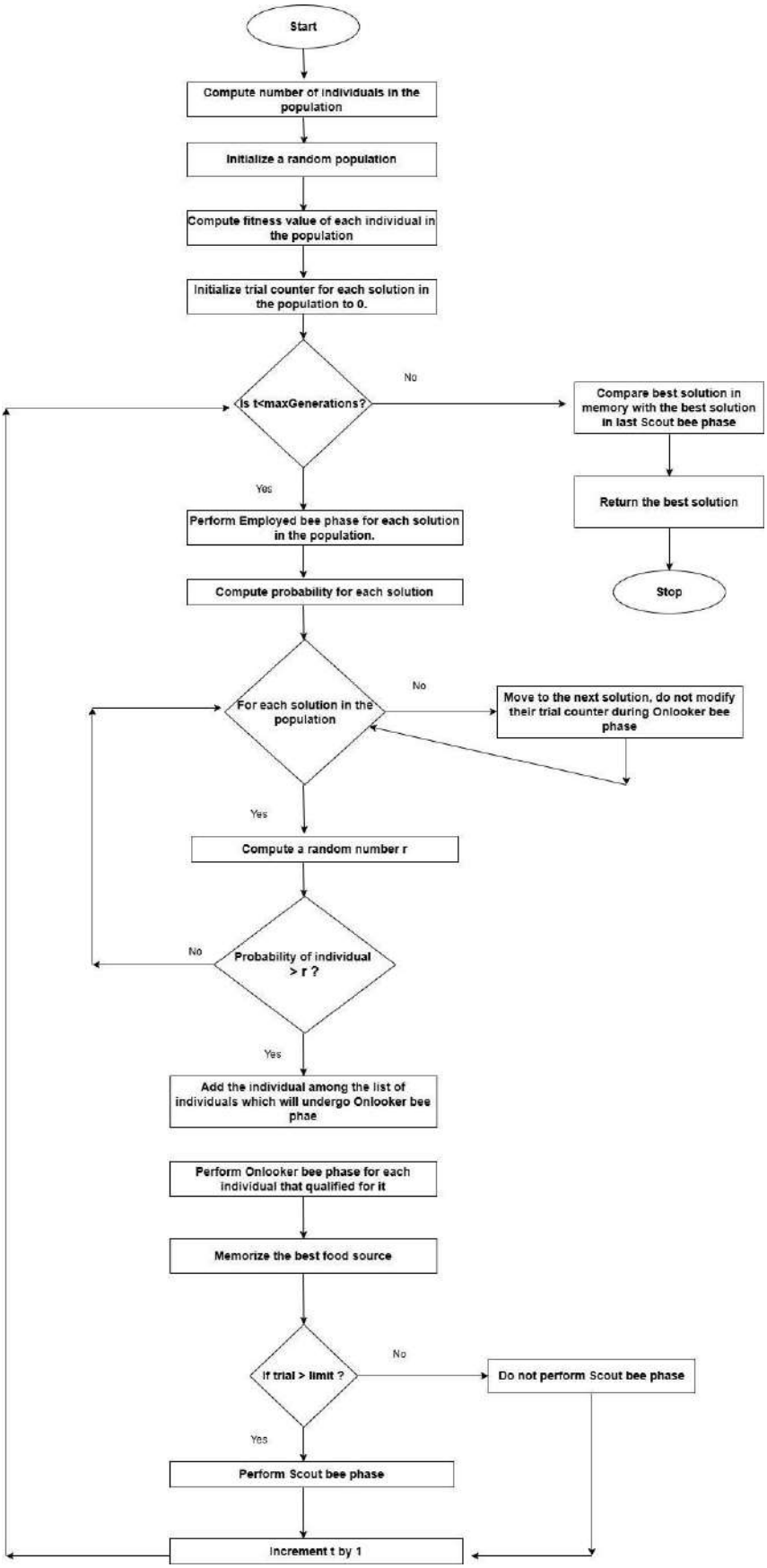


Figure 5.1.1 Flow chart of Artificial Bee Colony Optimization

5.2 Flower Pollination Algorithm: -

- It has two main components: -
    - Local pollination
    - Global pollination

- Parameters of the algorithm: -
    - maxGenerations: Number of generations (or iterations) for which the process is carried out
    - p: Switch probability which helps to decide whether we perform global pollination or local pollination.
    - λ: It helps to control the step size in Levy flight.
    - γ: Pulse emission rate which helps to determine the intensity of global pollination.
    - L(): Levy distribution computed using Levy flight.
    - g*: Current best solution
    - ε: Normal distribution

- Local pollination: -
    - We create an 1D array: ε whose dimension is equal to number of features.
    - The array is made using normal distribution of points in the range of 0 and 1.
    - Two unique solutions from the population are randomly selected: xj and xk.
    - A new solution is formed by performing local pollination using Equation (20).

- Global pollination: -
    - We create an 1D array: L whose dimension is equal to number of features.
    - The array is made using Levy distribution.
    - A new solution is formed be performing global pollination using Equation (19).

- Switch probability – p: -
    - If value of switch probability p is high, then we perform a greater number of global pollinations and lesser local pollinations.
    - If the value of switch probability p is small, then we perform a greater number of local pollinations and lesser global pollinations.
    - Thus, switch probability is a hyper parameter which needs to be set based on the requirements.

Flowchart of Flower Pollination Algorithm

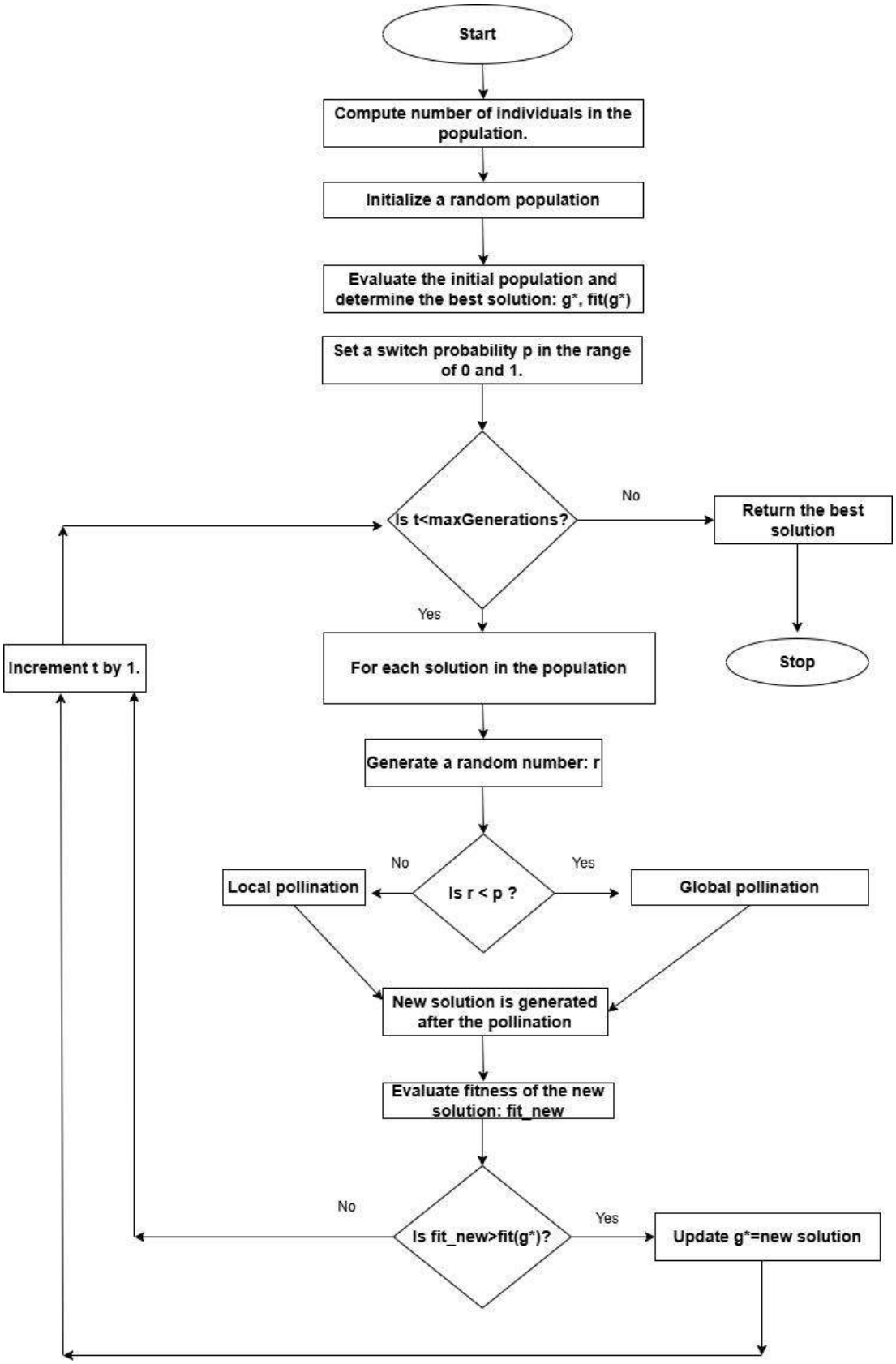


Figure 5.2.1 Flow chart of Flower Pollination Algorithm

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 Forests
5. Naïve Bayes
6. K-Nearest Neighbours

6.1 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.

6.2 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 efficient when we have large number of features.
4. It can be computationally expensive.
5. It is used to perform binary classification between two classes.
6. However, for multiclass classification, by using one vs one approach, we will need to train and evaluate multiple classifiers which leads to increased volume of computation and increased usage of time.
7. By one vs all approach, we need to again train multiple classifiers which again leads to the problem of increased volume of computation and increased usage of time.

6.3 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.

6.4 Random Forests: -

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.

6.5 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.

6.6 K-Nearest Neighbours: -

1. It is also represented as k-NN.
2. It classifies each input into with 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.
6. For each new data point for we need to perform classification, KNN computes its distance with all the training data points, and sorts the data points based on nearest neighbours of the test data.
7. The labels of K-nearest training datapoints determine the label for the test data based on voting and the label that gets maximum vote gets selected.
8. Thus, it gets extremely essential to select value of ‘k’ to get optimal performance from the algorithm.
9. Smaller value of k leads to generation of noise, because the model becomes sensitive to outliers.
10. Large value of k leads to model missing out important properties in the training dataset which are required for performing optimal classification.
11. It is commonly used in medical diagnosis and fraud detection.

Chapter 7

Research about different evaluation metrics used to evaluate results of 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: -
    - Why should the end user trust the trained model?
    - 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
13. Precision – Recall curve

Task 2: Definition and details about each metric: -

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

Table 7.1 Confusion matrix for binary classification between normal and attack

		Predictive values	
		Normal	**Attack**
Actual values	**Normal**	True Negative	False Positive
	Attack	False Negative	True Positive

    e. Representation of Confusion Matrix for Multi-class classifier: -

Table 7.2 Confusion matrix for multi-class classification between benign, attack_1 and attack_2

		Predicted values		
		Normal	**Attack_1**	**Attack_2**
Actual values	**Normal**	Cell 1	Cell 2	Cell 3
	Attack_1	Cell 4	Cell 5	Cell 6
	Attack_2	Cell 7	Cell 8	Cell 9

- True Positive: [Cell 5 + Cell 9]
- False Positive: [Cell 2 + Cell 3]
- True Negative: [Cell 1]
- False Negative: [Cell 4 + Cell 7]

f. Examples of above four components in the domain of cyber security: -
    i. True Positive: The model correctly predicts a malicious event as an attack.
    ii. True Negative: The model correctly predicts a normal event as normal.
    iii. False Positive: The model incorrectly predicts a normal event as an attack.
    iv. False Negative: The model incorrectly predicts an attack event as normal.

g. 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.

h. 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.

i. 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: -
    a. It gives overall correctness of the model.
    b. Accuracy is computed using Equation (4).
    c. It helps us understand how often the model predicts correctly.
    d. It is useful in scenarios when the dataset is balanced that is the target feature has balanced representation of all classes.
    e. It fails to justify false negative in imbalanced dataset.

3. Precision: -
    a. It gives accuracy of positive predictions.
    b. Precision is computed using Equation (5).
    c. It emphasizes on positive predictions made by the model.
    d. It is better than accuracy while working on imbalanced datasets since it demands minimization of False Positives to have higher score.

4. Recall: -
    a. It is also called Sensitivity.
    b. It gives model's ability to find all positive cases.
    c. Recall is computed using Equation (6).
    d. 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.
    e. Thus, if a model has high recall, it means it mostly identifies malicious events correctly.
    f. It works well on imbalanced datasets.

5. F1-Score: -
    a. It takes both precision and recall as inputs to give the output.
    b. F1-Score is computed using Equation (7).
    c. In binary classification, if F1-Score is close to 1, then the model has high accuracy and recall, which indicates model has good performance.
    d. It is useful when we work on imbalanced dataset.
    e. 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.

Table 7.3 Case 1 for confusion matrix for binary classification

		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.

Table 7.4 Case 2 for confusion matrix for binary classification

		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.

Table 7.5 Case 3 for confusion matrix for binary classification

		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.

Table 7.6 Case 4 for confusion matrix for binary classification

		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: -
    a. ROC: Reverse Operating Characteristics
    b. Here, we plot true positive rate (on y axis) and false positive rate (on x axis).
    c. The area under the curve is used to measure the model's performance.
    d. True Positive Rate is computed using Equation (8).
    e. False Positive Rate is computed using Equation (9).

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

8. Balanced accuracy: -
    a. Balanced accuracy is computed using Equation (10).
    b. True Negative Rate is computed using Equation (11).
    c. It is useful when classes of the dataset are imbalanced.
    d. Example: -
        i. Case 1: -
            1. True Positive Rate = 0
            2. True Negative Rate = 1
            3. Balanced accuracy = 0.5

        ii. Case 2: -
            1. Tue Positive Rate = 1
            2. True Negative Rate = 0
            3. Balanced accuracy = 0.5

        iii. Case 3: -
            1. True Positive Rate = 1
            2. True Negative Rate = 0.8125
            3. Balanced accuracy = 0.90625

        iv. Case 4: -
            1. True Positive Rate = 0.5
            2. True Negative Rate = 1
            3. Balanced accuracy = 0.75

    e. If the score is closer to 1, then the model has higher performance.
    f. It ranges between 0 to 1.

9. Matthews Correlation Coefficient (MCC): -
    a. It ranges between -1 to +1.
    b. MCC is computed using Equation (12).
    c. If MCC=0, the classifier performs random classification.
    d. It is used for binary class and multi-class classification.

10. Negative predictive value: -
    a. It tells how likely the event is not malicious if it is classified as benign.

b. Negative predictive value is computed using Equation (13).
c. Example: -
    i. Case 1: - NPV = 16 / (16 + 4) = 0.8
    ii. Case 2: - NPV = 0 / (0 + 0) = Not defined ~ 0
    iii. Case 3: - NPV = 13 / (13 + 0) = 1
    iv. Case 4: - NPV = 16 / (16 + 2) = 0.89

11. False discovery rate: -
    a. False Discovery Rate is computed using Equation (14).
    b. Here we determine out of all the events that the model classified as malicious; how many were incorrect.
    c. Thus, this helps us determine the noise generated by the model.
    d. Example: -
        i. Case 1: - FDR = 0 / (0 + 0) = Not defined ~ 0
        ii. Case 2: - FDR = 16 / (16 + 4) = 0.8
        iii. Case 3: - FDR = 3 / (3 + 4) = 0.428
        iv. Case 4: - FDR = 0 / (0 + 2) = 0
    e. We can also use it to for feature selection, to determine features that are associated with malicious events.

12. Cohen – kappa: -
    a. It takes into account that model may correctly classify the some of the events purely by chance.
    b. It is also called Kappa Score (k).
    c. Kappa score is computed using Equation (15).
    d. k=1: There is complete agreement between the models.
    e. k<=0: There is no agreement between the models.
    f. Example: -

Table 7.7 Results of Cohen's Kappa for the four cases

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

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

Table 7.8 Interpretation of Cohen's kappa score

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

h. As per our 4 cases: -
    i. Case 1 and case 2 have no agreement. Thus, the two models are far away from expected model.
    ii. Case 3 and case 4 have substantial agreement. Thus, the two models are closer to the expected model.

13. Precision – Recall curve: -
    a. It is useful for imbalanced dataset.
    b. 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.
    c. Then we plot the graph with Recall on x-axis and Precision on y-axis.
    d. The area under Precision Recall curve is the indicator of performance.
    e. Larger the area, better the model performs.

Chapter 8

Feature selection and training of models

8.1 Scaling of independent features: -

Some of the approaches for scaling and standardization of features: -

1. Standard Scaler
2. Robust Scaler
3. Min-Max Scaler
4. MaxAbs Scaler

In this project, we used two approaches and performed independent scaling operations: -

1. Standard Scaler
2. Robust Scaler

Thus, two approaches are used during feature selection process and the results of feature selection are used for corresponding model training and model evaluation.

8.2 Selection of machine learning algorithm: -

We have used K-NN as the Machine Learning algorithm in the project, both for feature selection and for training the models for classification.

Reasons for choosing K-NN algorithm: -

1. It is a lazy learner, thus computationally less intensive in training phase.
2. Since we are using heuristic algorithms for feature selection, by definition they are computationally expensive.
3. As the result, in order to perform recurring training of ML model and evaluation during feature selection efficiently, K-NN algorithm was used.
4. It also allows us to use same algorithm for both binary classification and multiclass classification.

From scikit-learn library, we import KNeigborsClassifier model, which by default has k (n_neighbors)=5. We have used the default configuration in feature selection and model training.

High level flowchart for feature selection, training model and evaluation of the models.

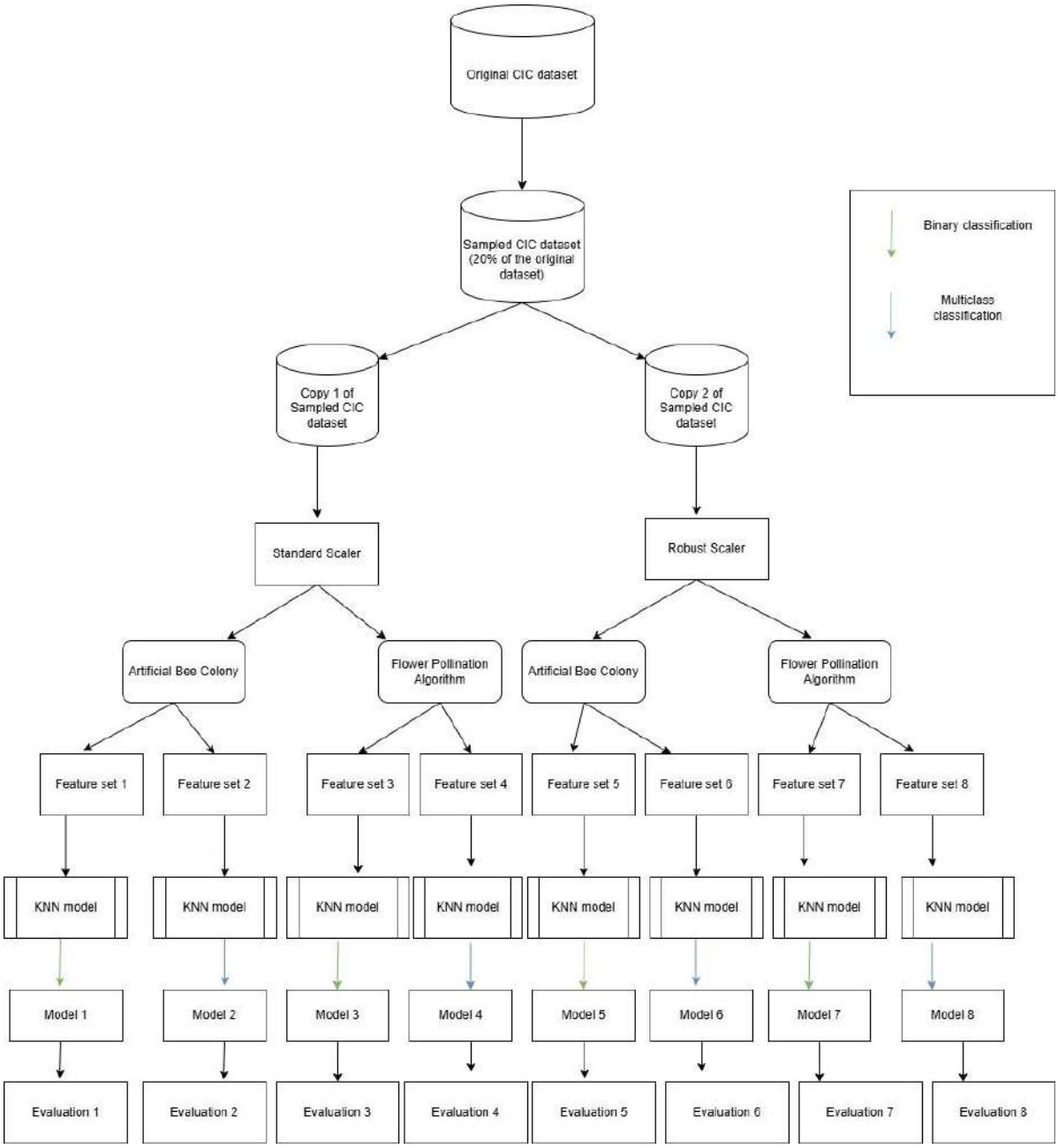


Figure 8.2.1 High level flow chart of the process used for feature selection, training of models and evaluation

8.3 Fitness function: -

In feature selection process, we have used Recall as the fitness function.

Reason: -

1. CIC is a cyber security dataset; thus, correct classification of malicious events is more critical and important than correct classification of benign events.
2. Recall is useful while working on imbalanced dataset, since it focuses on maximizing True positives and minimizing False negatives. By implication, in the case of imbalanced dataset where distribution of events is skewed against malicious (or attack) events, recall helps to

maximize the correct classification of malicious events and reduce incorrect classification of malicious events as benign.

Feature selection has dual objectives: -

1. Increase the performance of models to ensure higher number of correct classifications.
2. Decrease the number of features selected to train the model, to overcome overfitting during training process and reduce complexity of the models.

Since the two objectives are in opposite directions and sometimes tend to conflict with each other, it was felt essential to modify the fitness function.

In heuristic algorithms, we improve results by performing exploration and exploitation over many iterations (known as generations). In practice, heuristic algorithms are executed over 100 to 1000 iterations, and the goal is to get more optimal search results as the number of iterations increase.

However, due to constraint of time and computation, we have restricted the number of iterations (generations) for feature selection to 5.

As the result, to prioritize smaller subset of features during feature selection, we have introduced **penalty** in the fitness function.

The penalty is computed by below formula: -

Penalty = (Length of current subset of features/Total number of independent features) * (Current generation/Maximum number of generations) * 2

In our dataset, after pre-processing, Total number of independent features = 45.

Let us consider the current subset of features have count = 10

For generation 1: - penalty = (10/45) * (1/5) * 2 = 0.088

For generation 2: - penalty = (10/45) * (2/5) * 2 = 0.178

For generation 3: - penalty = (10/45) * (3/5) * 2 = 0.267

For generation 4: - penalty = (10/45) * (4/5) * 2 = 0.356

For generation 5: - penalty = (10/45) * (5/5) * 2 = 0.444

Thus, for same subset of features the penalty value gets increased as we progressed further in generation.

This will help us to reduce the overall fitness of solutions whose subset length increases or remains constant as we progress to the next generation. And it enables to push the algorithm to find more subset of features having optimal results, that is lower feature count and better recall value.

Thus, the new fitness value of a subset of features will be computed by taking difference between recall and penalty, and the solution with maximum fitness value will get selected to train the model.

Let us consider the below scenario observed at the end of 5th generation: -

1. Subset 1: -
    a. Number of features = 35
    b. Recall = 0.98
    c. Penalty = (35/45) * (5/5) * 2 = 1.56

d. Updated fitness = 0.98 – 1.56 = -0.59

2. Subset 2: -
    a. Number of features = 15
    b. Recall = 0.90
    c. Penalty = (15/45) * (5/5) * 2 = 0.67
    d. Updated fitness = 0.90 – 0.67 = 0.23

Thus, we observed: -

- Recall for subset 1 is greater than recall for subset 2.
- Number of features in subset 2 is lesser than number of features in subset 1.
- Updated fitness of subset 2 is greater than updated fitness of subset 1.
- As the result, among the two feature subsets, subset 2 will be selected over subset 1.

Observations: -

1. Model complexity increases when we have more number features used for training. As the complexity of model increases, its likelihood of overfitting and inability to handle unseen data also increases.

2. There is always a trade-off for choosing between lower number of features and better performance of model. Thus, we need to determine if we can get a subset of features which have smaller count but performance close to subset of features having larger count. This will enable us to train the models efficiently and still get results closer to expected success.

Chapter 9

Evaluation of models

9.1 Evaluation of models

For each pair of heuristic algorithm and scaler type, we fetched two sets of independent results and evaluated them together.

This is because although heuristic algorithm's objective is to return good results, it does not guarantee returning best result for a given problem every time its executed. Moreover, it also does not guarantee returning same results for a given problem after each execution.

In real-world scenario, we may want to run and test the algorithm over multiple number of times in order to get a good sample size of results for making inference, but due to limitations of time and compute resources, we restricted to two results and evaluated the same.

Evaluation of results obtained using Artificial Bee Colony algorithm

1. Binary classification using Standard Scaler
2. Multi-class classification using Standard Scaler
3. Binary classification using Robust Scaler
4. Multi-class classification using Robust Scaler

Evaluation of results obtained using Artificial Bee Colony algorithm

1. Binary classification using Standard Scaler
2. Multi-class classification using Standard Scaler
3. Binary classification using Robust Scaler
4. Multi-class classification using Robust Scaler

9.2 Evaluation of results obtained using Artificial Bee Colony algorithm

9.2.1. Binary classification using Standard Scaler

Table 9.2.1 Results obtained from Binary classification by using ABC algorithm for feature selection and Standard scaler to scale independent features

	Balanced test dataset		**Imbalanced test dataset**	
	Solution 1	**Solution 2**	**Solution 1**	**Solution 2**
Number of Features	13	11	13	11
Confusion matrix	[[20103, 589], [476, 61600]]	[[20006, 686], [314, 61762]]	[[1404119, 33045], [80723, 216214]]	[[1382983, 54181], [76129, 220808]]
Accuracy	0.987	0.988	0.934	0.925
Precision	0.991	0.989	0.867	0.803
Recall	0.992	0.995	0.728	0.744
F1-Score	0.991	0.992	0.792	0.772
AUC Score	0.982	0.981	0.853	0.853

Balanced accuracy	0.982	0.981	0.853	0.853
MCC	0.966	0.968	0.757	0.728
NPV	0.977	0.985	0.946	0.948
FDR	0.009	0.011	0.133	0.197
Cohen Kappa	0.966	0.968	0.753	0.727

Balanced test dataset: Comparison of ROC curves

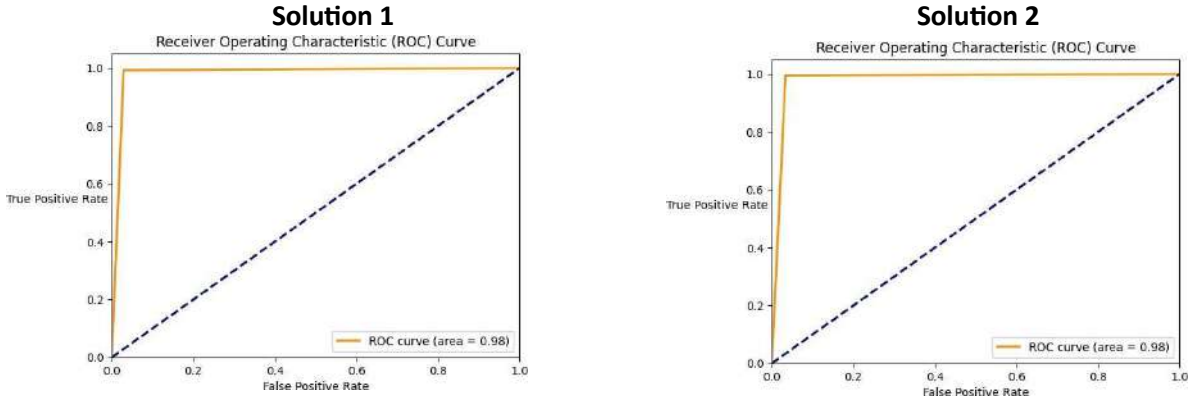


Figure 9.2.1.1 ROC curve on balanced test dataset for solution 1 and solution 2 obtained for Binary classification using ABC algorithm for feature selection and Standard scaler to scale independent features

Balanced test dataset: Comparison of Precision – Recall curves

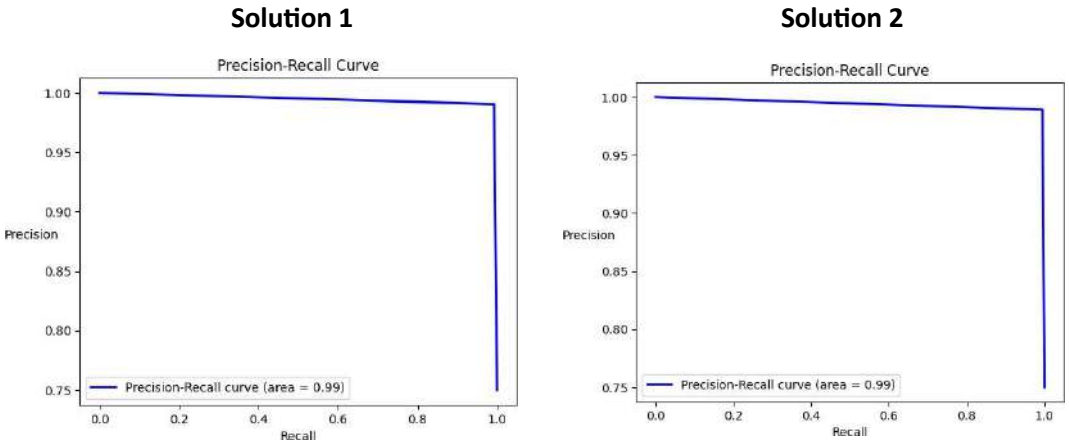


Figure 9.2.1.2 Precision - Recall curve on balanced test dataset for solution 1 and solution 2 obtained for Binary classification using ABC algorithm for feature selection and Standard scaler to scale independent features

Imbalanced test dataset: Comparison of ROC curves

Solution 1 **Solution 2**

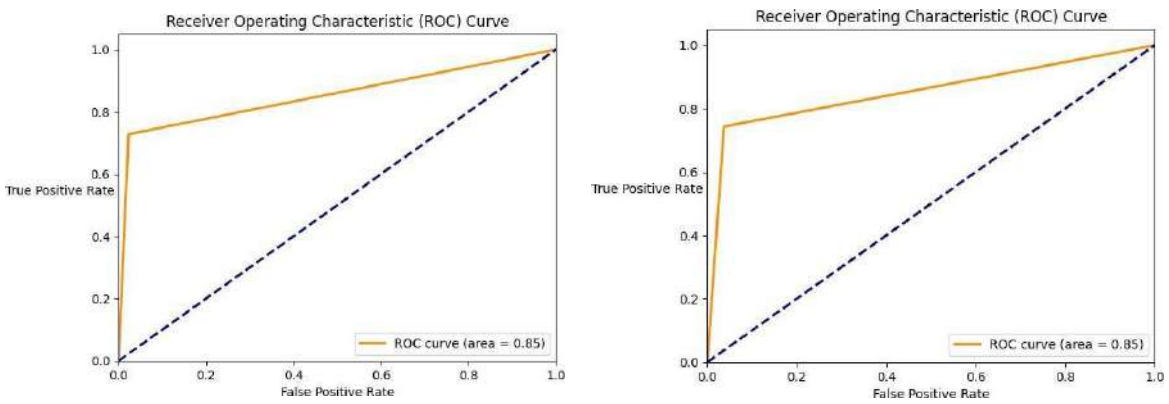


Figure 9.2.1.3 ROC curve on imbalanced test dataset for solution 1 and solution 2 obtained for Binary classification using ABC algorithm for feature selection and Standard scaler to scale independent features

Imbalanced test dataset: Comparison of Precision – Recall curves

Solution 1 **Solution 2**

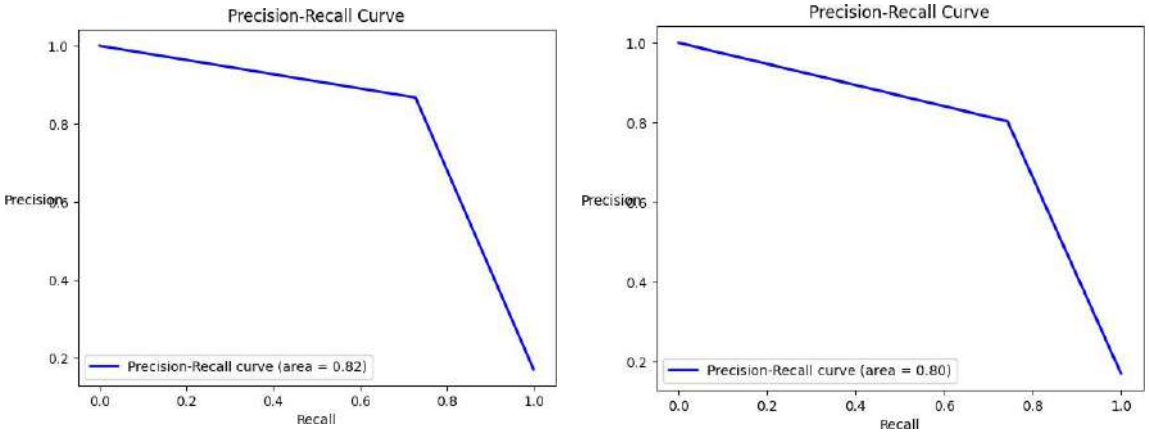


Figure 9.2.1.4 Precision - Recall curve on imbalanced test dataset for solution 1 and solution 2 obtained for Binary classification using ABC algorithm for feature selection and Standard scaler to scale independent features

Balanced dataset: -

1. Both solutions have almost **equal and very high accuracy**. Thus, both models perform correct prediction around 98% of the times.
2. Both solutions have **very high precision**, thus, the false positives for both models were less. Solution 1 has slightly higher precision than solution 2.
3. Both solutions have **very high recall**, thus, both models correctly classify most of the malicious events (True Positives). Solution 2 has slightly higher recall than solution 1.
4. Both solutions have F1-score close to 1, thus, both models have **good performance.**
5. Both solutions have almost **equal and very high AUC scores** which is closer to 1. Thus, both models have great ability to differentiate between benign and malicious events.
6. Both solutions have **almost equal and very high balanced accuracy** which is closer to 1. Thus, both models have high precision and recall.

7. Both solutions have **almost equal and very high MCC** which is closer to 1. Thus, both models have very close agreements with actual labels of each event.
8. Both solutions have **very high negative predictive value**. Thus, for both models when an event was classified as benign, around 98% times it was correct and the event was actually benign (that is not malicious). Solution 2 has higher negative predictive value than solution 1.
9. Both solutions have **very low false discovery rate**. Thus, for both models when an event was classified as malicious, around 10% of the times it was incorrect and the event was actually benign. Thus, both the models have noise close to 10%.
10. Both solutions have Cohen Kappa in the range of 0.81 to 0.99. Thus, the models have **near perfect agreement** and are closer to the expected model.
11. Both solutions have **ROC curve closer to axes**, and the **elbow is closer to coordinate (0, 1)** which indicates both the models have larger value of area in their respective ROC curve. Thus, both the models are good classifiers.
12. Both solutions have **Precision – Recall curve have very high value for both precision and recall**. Thus, both the models are good classifiers.

Imbalanced dataset: -

1. Both the solutions have **a very high accuracy**. Thus, both models perform correct prediction around 93% of the times.
2. Both solutions have **high precision**, greater than 80%. Thus, the model incorrectly classifies some of the normal events as malicious, resulting in false positives.
3. Both solutions have **average recall**, around 70%. Thus, the model incorrectly classifies multiple malicious events as benign.
4. Both solutions have **average F1-score**, around 78%. This is because the model's performance drops while correctly predicting both normal and malicious events.
5. Both solutions **have equal and high AUC score**: 85%. Thus, both models have consistent ability to differentiate between benign and malicious events.
6. Both solutions have **equal and high balanced accuracy**: 85%. Thus, both models have high precision and recall.
7. Both solutions have **average MCC**, around 70%. MCC score is greater than 0 and closer to 1. Thus, both models have some agreement with the actual labels.
8. Both solutions have **very high negative predictive value**: 94%. Thus, for both models when an event was classified as benign, around 94% times it was correct and the event was actually benign (that is not malicious). Solution 2 has higher negative predictive value than solution 1.
9. Both solutions have **low false discovery rate**. Thus, for solution 1 if an event is classified as malicious, 13% times it is misclassified and the event was actually normal (benign). For solution 2, if an event is classified as malicious, 19% of times it is misclassified and the event was actually normal (benign). As the result, model for solution 1 was better than the model for solution 2.
10. Both solutions have Cohen Kappa score in the range of 0.61 and 0.80. Thus, the models have **substantial agreement with the expected model**.
11. Both solutions have **ROC curve closer to axes**, and the **elbow is closer to coordinate (0, 0.8)** which indicates both the models have relatively large value of area in their respective ROC curve and are medium fit.

12. Both solutions have **Precision – Recall curve with large area**. The area of P-R curve for solution 1 is greater than the area of P-R curve for solution 2. Thus, model for solution 1 performs better than the model for solution 2.

9.2.2. Multi-class classification using Standard Scaler

Table 9.2.2 Results obtained from Multiclass classification by using ABC algorithm for feature selection and Standard scaler to scale independent features

	Balanced dataset		**Imbalanced dataset**	
	Solution 1	**Solution 2**	**Solution 1**	**Solution 2**
Number of features	27	30	27	30
Confusion matrix	[[19998, 74, 32, 588], [70, 20616, 0, 6], [87, 2, 20603, 0], [104, 0, 1, 20587]]	[[19939, 60, 23, 630], [143, 20542, 1, 6], [82, 0, 20605, 5], [84, 0, 2, 20606]]	[[1392586, 10923, 4728, 28927], [155, 28994, 2, 12], [167, 3, 20521, 1], [53871, 94, 22, 193095]]	[[1399394, 9646, 7061, 21063], [319, 28827, 2, 15], [162, 0, 20526, 4], [54696, 1163, 20, 191203]]
Accuracy	0.988	0.988	0.943	0.946
Precision	0.989	0.989	0.845	0.864
Recall	0.996	0.995	0.817	0.813
F1-Score	0.992	0.992	0.831	0.838
Balanced accuracy	0.981	0.98	0.893	0.894
MCC	0.969	0.967	0.797	0.807
NPV	0.987	0.985	0.963	0.962
FDR	0.011	0.011	0.155	0.136
Cohen Kappa	0.984	0.983	0.803	0.811

Balanced dataset: -

1. Both solutions have **almost equal and very high accuracy.** Thus, both models perform correct prediction around 98% of the times.
2. Both solutions have **very high precision**, thus, the false positives for both models were less.
3. Both solutions have **very high recall**, thus, both models correctly classify most of the malicious events (True Positives).
4. Both solutions have F1-score close to 1, thus, both models **have good performance**.
5. Both solutions have **almost equal and very high balanced accuracy** which is closer to 1. Thus, both models have high precision and recall.
6. Both solutions have **almost equal and very high MCC** which is closer to 1. Thus, both models have very close agreements with actual labels of each event.

7. Both solutions **have very high negative predictive value**. Thus, for both models when an event was classified as benign, around 98% times it was correct and the event was actually benign (that is not malicious). Solution 2 has higher negative predictive value than solution 1.
8. Both solutions **have very low false discovery rate**. Thus, for both models when an event was classified as malicious, around 1% of the times it was incorrect and the event was actually benign. Thus, both the models have noise close to 1%.
9. Both solutions have Cohen Kappa in the range of 0.81 to 0.99. Thus, the models have **near perfect agreement** and are closer to the expected model.

Imbalanced dataset: -

1. Both the solutions have a **very high accuracy**. Thus, both models perform correct prediction around 94% of the times.
2. Both solutions **have high precision**, greater than 84%. Thus, the model incorrectly classifies some of the normal events as malicious, resulting in false positives.
3. Both solutions have **high recall**, around 81%. Thus, the model incorrectly classifies some malicious events as benign.
4. Both solutions have **average F1-score**, around 83%. This is because the model's performance drops while correctly predicting both normal and malicious events.
5. Both solutions have almost **equal and high balanced accuracy**: 89%. Thus, both models have high precision and recall.
6. Both solutions have **high MCC**, around 80%. MCC score is greater than 0 and closer to 1. Thus, both models have some agreement with the actual labels.
7. Both solutions have **almost equal and very high negative predictive value**: 96%. Thus, for both models when an event was classified as benign, around 96% times it was correct and the event was actually benign (that is not malicious). Solution 1 has slightly higher negative predictive value than solution 1.
8. Both solutions have **low false discovery rate**. Thus, for solution 1 if an event is classified as malicious, 15% times it is misclassified and the event was actually normal (benign). For solution 2, if an event is classified as malicious, 13% of times it is misclassified and the event was actually normal (benign). As the result, model for solution 2 was better than the model for solution 1.
9. Both solutions have Cohen Kappa score greater than 0.80. Thus, the models **have near perfect agreement with the expected model**.

9.2.3. Binary classification using Robust Scaler

Table 9.2.3 Results obtained from Binary classification by using ABC algorithm for feature selection and Robust scaler to scale independent features

	Balanced dataset		Imbalanced dataset	
	Solution 1	**Solution 2**	**Solution 1**	**Solution 2**
Number of features	14	16	14	16
Confusion matrix	[[20207, 485], [224, 61852]]	[[20304, 388], [301, 61775]]	[[824867, 612288], [120690, 176247]]	[[920408, 516756], [246720, 50217]]
Accuracy	0.991	0.992	0.577	0.56
Precision	0.992	0.994	0.224	0.089
Recall	0.996	0.995	0.594	0.169
F1-Score	0.994	0.994	0.325	0.116
AUC Score	0.986	0.988	0.584	0.405
Balanced accuracy	0.986	0.988	0.584	0.405
MCC	0.977	0.978	0.127	-0.153
NPV	0.989	0.985	0.872	0.789
FDR	0.008	0.006	0.776	0.911
Cohen Kappa	0.977	0.978	0.101	-0.14

Balanced test dataset: Comparison of ROC curves

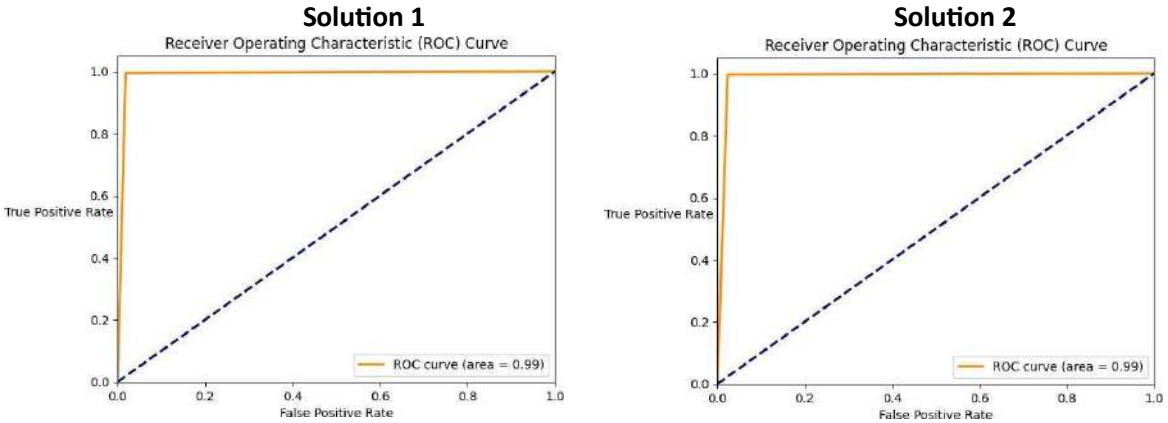


Figure 9.2.3.1 ROC curve on balanced test dataset for solution 1 and solution 2 obtained for Binary classification using ABC algorithm for feature selection and Robust scaler to scale independent features

Balanced test dataset: Comparison of Precision – Recall curves

Solution 1 **Solution 2**

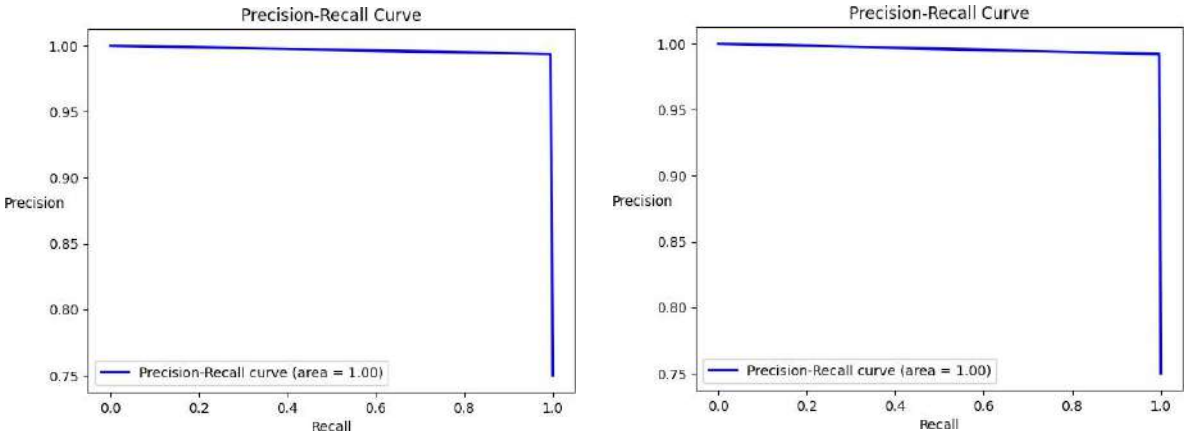


Figure 9.2.3.2 Precision - Recall curve on balanced test dataset for solution 1 and solution 2 obtained for Binary classification using ABC algorithm for feature selection and Robust scaler to scale independent features

Imbalanced test dataset: Comparison of ROC curves

Solution 1 **Solution 2**

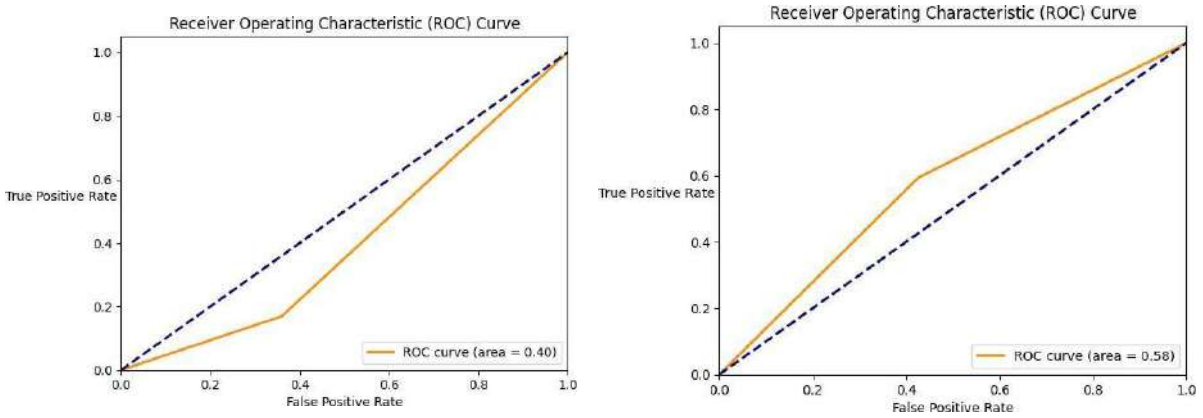


Figure 9.2.3.3 ROC curve on imbalanced test dataset for solution 1 and solution 2 obtained for Binary classification using ABC algorithm for feature selection and Robust scaler to scale independent features

Imbalanced test dataset: Comparison of Precision – Recall curves

Solution 1 **Solution 2**

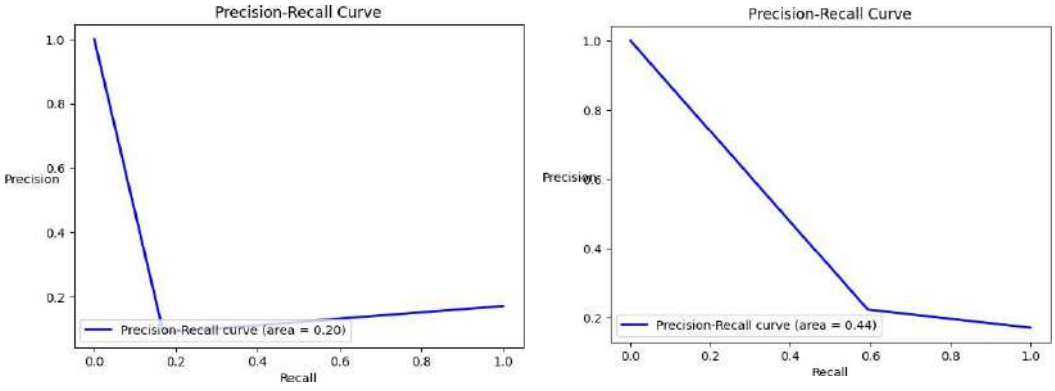


Figure 9.2.3.4 Precision - Recall curve on imbalanced test dataset for solution 1 and solution 2 obtained for Binary classification using ABC algorithm for feature selection and Robust scaler to scale independent features

Balanced dataset: -

1. Both solutions have **very high accuracy**. Thus, both models perform correct prediction around 99% of the times.
2. Both solutions have **very high precision**, thus, the false positives for both models were less. Solution 2 has slightly higher precision than solution 1.
3. Both solutions have **very high recall**, thus, both models correctly classify most of the malicious events (True Positives). Solution 1 has slightly higher recall than solution 1.
4. Both solutions have F1-score close to 1, thus, both models **have good performance**.
5. Both solutions have almost **equal and very high AUC scores** which is closer to 1. Thus, both models have great ability to differentiate between benign and malicious events.
6. Both solutions have **almost equal and very high balanced accuracy** which is closer to 1. Thus, both models have high precision and recall.
7. Both solutions have **almost equal and very high MCC** which is closer to 1. Thus, both models have very close agreements with actual labels of each event.
8. Both solutions have **very high negative predictive value**. Thus, for both models when an event was classified as benign, around 98% times it was correct and the event was actually benign (that is not malicious). Solution 1 has higher negative predictive value than solution 2.
9. Both solutions have **very low false discovery rate**. Thus, for both models when an event was classified as malicious, around 8% of the times it was incorrect and the event was actually benign. Thus, both the models have noise less than 8%.
10. Both solutions have Cohen Kappa in the range of 0.81 to 0.99. Thus, the models have **near perfect agreement and are closer to the expected model**.
11. Both solutions **have ROC curve closer to axes**, and the **elbow is closer to coordinate (0, 1)** which indicates both the models have larger value of area in their respective ROC curve. Thus, both the models are good classifiers.
12. Both solutions have **Precision – Recall curve have very high value for both precision and recall**. Thus, both the models are good classifiers.

Imbalanced dataset: -

1. Both the solutions have a very low accuracy. Thus, both models perform correct prediction around 56% of the times.
2. Both solutions have very low precision. Thus, the model incorrectly classifies many normal events as malicious, resulting in false positives. Model for solution 1 has better precision than the model for solution 2.
3. Both solutions have low recall. Thus, the model incorrectly classifies multiple malicious events as benign. Model for solution 1 has better recall than the model for solution 2.
4. Both solutions have low F1-score. This is because the model's performance is significantly low while correctly predicting both normal and malicious events.
5. Both solutions have low AUC score. Thus, both models are inconsistent to differentiate between benign and malicious events.
6. Both solutions have low balanced accuracy. Thus, both models have low precision and recall.
7. Both solutions have low MCC. MCC score for solution 1 is positive but closer to 0, thus its performance is very close to random guessing. MCC score of solution 2 is closer to -1, thus, it has total disagreement between the model's predictions and the actual labels.

8. Both solutions have high negative predictive value. Model for solution 1 has NPV: 87% and model for solution 2 has NPV: 78%. Thus, for both models when an event was classified as benign, around 87%% times and 78% times respectively it was correct and the event was actually benign (that is not malicious). Solution 1 has higher negative predictive value than solution 2.
9. Both solutions have high false discovery rate. Thus, for solution 1 if an event is classified as malicious, 77% times it is misclassified and the event was actually normal (benign). For solution 2, if an event is classified as malicious, 91% of times it is misclassified and the event was actually normal (benign).
10. Both solutions have Cohen Kappa score in the range of 0 and 0.1. Thus, the models have no agreement with the expected model.
11. Both solutions have very small area under the ROC curve. For solution 1, the ROC curve is below random guessing line. For solution 2, the ROC curve is just above the random guessing line. Thus, both the classifiers are very bad fit.
12. Both solutions have Precision – Recall curve with small area. Thus, both models have very poor precision and recall.

9.2.4. Multi-class classification using Robust Scaler

Table 9.2.4 Results obtained from Multiclass classification by using ABC algorithm for feature selection and Robust scaler to scale independent features

	Balanced dataset		**Imbalanced dataset**	
	Solution 1	**Solution 2**	**Solution 1**	**Solution 2**
Number of features	32	25	32	25
Confusion matrix	[[20260, 61, 37, 334], [54, 20631, 0, 7], [24, 0, 20668, 0], [183, 4, 2, 20503]]	[[20294, 51, 24, 323], [44, 20641, 0, 7], [17, 1, 20674, 0], [141, 0, 5, 20546]]	[[809752, 354838, 206919, 65655], [638, 27591, 757, 177], [20021, 0, 348, 323], [150550, 24460, 515, 71557]]	[[1264989, 68723, 24664, 78788], [916, 28178, 36, 33], [16201, 0, 346, 4145], [108326, 2932, 129, 135695]]
Accuracy	0.992	0.993	0.532	0.828
Precision	0.993	0.994	0.137	0.488
Recall	0.996	0.997	0.368	0.567
F1-Score	0.994	0.995	0.199	0.525
Balanced accuracy	0.987	0.989	0.465	0.724
MCC	0.978	0.981	-0.051	0.422
NPV	0.987	0.99	0.825	0.91
FDR	0.007	0.006	0.863	0.512
Cohen Kappa	0.989	0.99	0.075	0.444

Balanced dataset: -

1. Both solutions have **almost equal and very high accuracy**. Thus, both models perform correct prediction around 99% of the times.
2. Both solutions have **very high precision**, thus, the false positives for both models were less.
3. Both solutions have **very high recall**, thus, both models correctly classify most of the malicious events (True Positives).
4. Both solutions have F1-score close to 1, thus, both models **have good performance**.
5. Both solutions have **almost equal and very high balanced accuracy** which is closer to 1. Thus, both models have high precision and recall.
6. Both solutions have **almost equal and very high MCC** which is closer to 1. Thus, both models have very close agreements with actual labels of each event.
7. Both solutions have **very high negative predictive value**. Thus, for both models when an event was classified as benign, around 99% times it was correct and the event was actually benign (that is not malicious). Solution 2 has higher negative predictive value than solution 1.
8. Both solutions have **very low false discovery rate**. Thus, for both models when an event was classified as malicious, around 0.7% of the times it was incorrect and the event was actually benign. Thus, both the models have noise less than 1%.
9. Both solutions have Cohen Kappa in the range of 0.81 to 0.99. Thus, the models **have near perfect agreement and are closer to the expected model**.

Imbalanced dataset: -

1. **Solution 1 has low accuracy**: 0.532 and **solution 2 has relatively higher accuracy**: 0.828. Thus, overall correctness of model for solution 2 is better than the model for solution 1.
2. Both solutions have **low precision**. Thus, the model incorrectly classifies many normal events as malicious, resulting in false positives.
3. Both solutions have **low recall.** Thus, the model incorrectly classifies many malicious events as benign.
4. Both solutions have **low F1-score**. This is because the model's performance drops while correctly predicting both normal and malicious events.
5. Solution 1 has **low balanced accuracy**: 0.465, and solution 2 has relatively higher balanced accuracy: 0.724. Although solution 2 has lower precision and recall and still has higher balanced accuracy. It may be due to imbalanced nature of the dataset and its higher accuracy also supports the results.
6. Both solutions have **low MCC**, closer to 0. MCC **for solution 1** is negative and closer to -1, which indicates there is **strong disagreement between the model's predictions and the actual labels**. MCC **for solution 2** is positive and closer to 0, which indicates that **the model performs similar to random guessing.**
7. Both solutions have **high negative predictive value**. Thus, for both models when an event was classified as benign, for solution 1 it was correct 82.5% times and for solution 2 it was correct 91% times and the event was actually benign (that is not malicious). Solution 2 has higher negative predictive value than solution 1.
8. Solution 1 has **high false discovery rate**, and solution 2 has low false discovery rate. Thus, for solution 1 if an event is classified as malicious, 86.3% times it is misclassified and the event was actually normal (benign). For solution 2, if an event is classified as malicious, 51.2% of times it is misclassified and the event was actually normal (benign). As the result, model for solution 2 was better than the model for solution 1.

9. Solution 1 has Cohen Kappa score less than 0.1, thus, the model for **solution 1 has no agreement with the expected model**. Solution 2 has Cohen Kappa score in the range of 0.41 to 0.60, thus, the model for **solution 2 has moderate agreement with the expected model**.

9.3 Evaluation of results obtained using Flower Pollination Algorithm

9.3.1. Binary classification using Standard Scaler

Table 9.3.1 Results obtained from Binary classification by using FPA algorithm for feature selection and Standard scaler to scale independent features

	Balanced dataset		**Imbalanced dataset**	
	Solution 1	**Solution 2**	**Solution 1**	**Solution 2**
Number of features	20	25	20	25
Confusion matrix	[[20045, 647], [356, 61720]]	[[20012, 680], [206, 61870]]	[[1401121, 36043], [69228, 227649]]	[[1400635, 36529], [60777, 236160]]
Accuracy	0.988	0.989	0.939	0.944
Precision	0.99	0.989	0.863	0.866
Recall	0.994	0.997	0.767	0.795
F1-Score	0.992	0.993	0.812	0.829
AUC Score	0.981	0.982	0.871	0.885
Balanced accuracy	0.981	0.982	0.871	0.885
MCC	0.968	0.971	0.778	0.797
NPV	0.983	0.99	0.953	0.958
FDR	0.01	0.011	0.137	0.134
Cohen Kappa	0.968	0.971	0.776	0.796

Balanced test dataset: Comparison of ROC curves

Solution 1 **Solution 2**

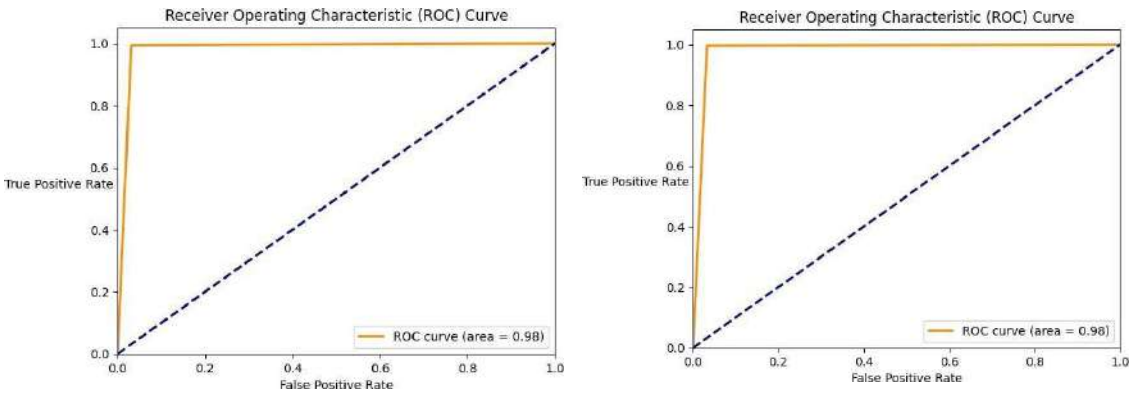


Figure 9.3.1.1 ROC curve on balanced test dataset for solution 1 and solution 2 obtained for Binary classification using FPA algorithm for feature selection and Standard scaler to scale independent features

Balanced test dataset: Comparison of Precision – Recall curves

Solution 1 **Solution 2**

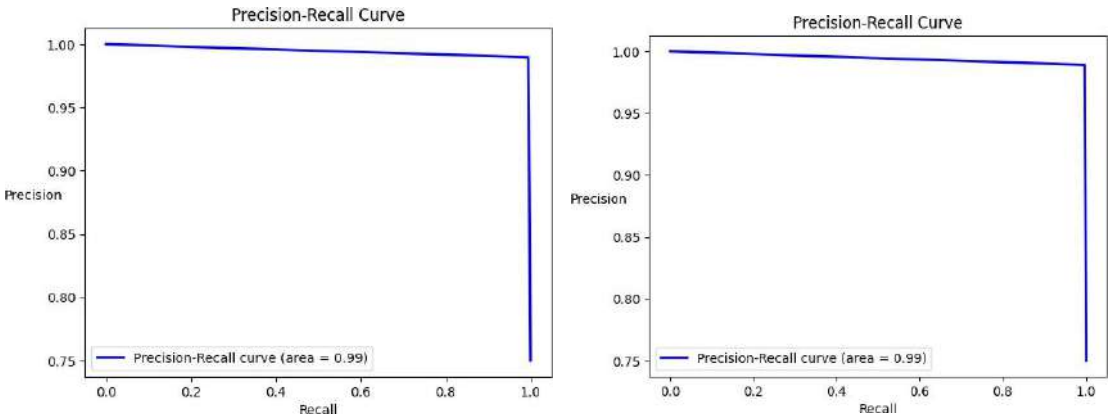


Figure 9.3.1.2 Precision - Recall curve on balanced test dataset for solution 1 and solution 2 obtained for Binary classification using FPA algorithm for feature selection and Standard scaler to scale independent features

Imbalanced test dataset: Comparison of ROC curves

Solution 1 **Solution 2**

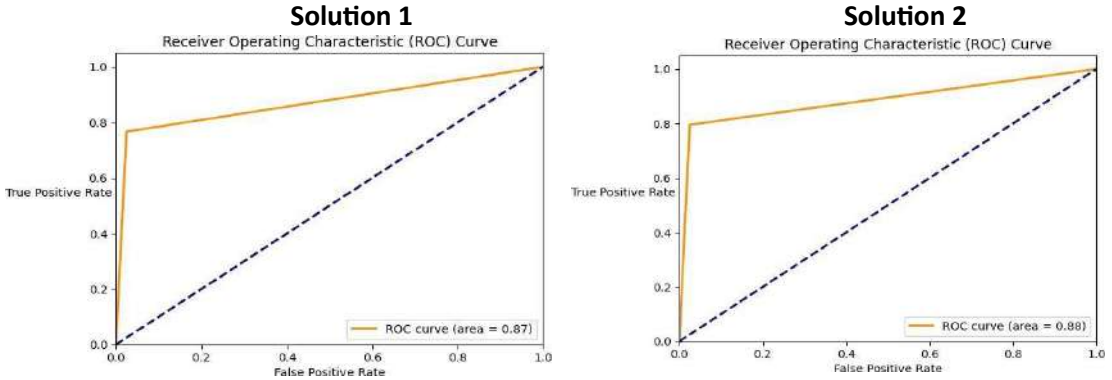


Figure 9.3.1.3 ROC curve on imbalanced test dataset for solution 1 and solution 2 obtained for Binary classification using FPA algorithm for feature selection and Standard scaler to scale independent features

Imbalanced test dataset: Comparison of Precision – Recall curves

Solution 1 **Solution 2**

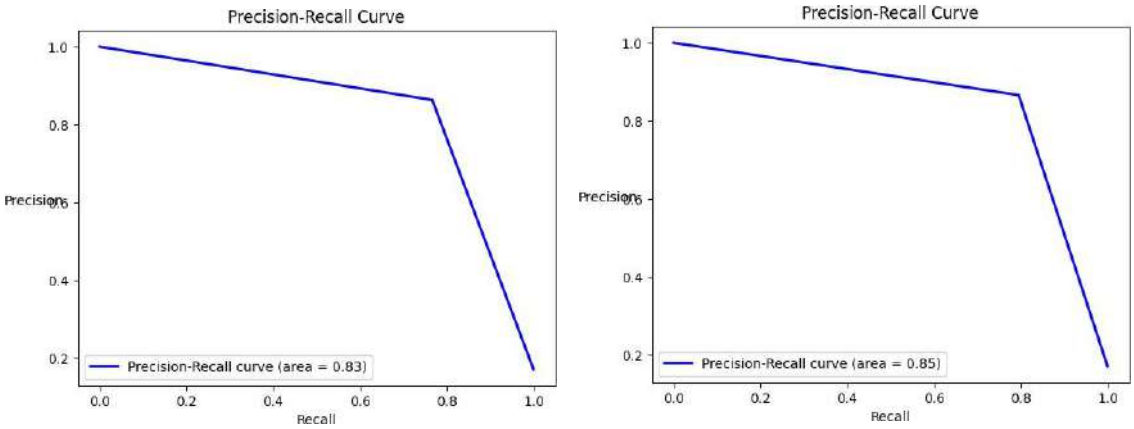


Figure 9.3.1.4 Precision - Recall curve on imbalanced test dataset for solution 1 and solution 2 obtained for Binary classification using FPA algorithm for feature selection and Standard scaler to scale independent features

Balanced dataset: -

1. Both solutions have **almost equal and very high accuracy**. Thus, both models perform correct prediction around 98% of the times.
2. Both solutions have **very high precision**, thus, the false positives for both models were less. Solution 1 has slightly higher precision than solution 2.
3. Both solutions have **very high recall,** thus, both models correctly classify most of the malicious events (True Positives). Solution 2 has slightly higher recall than solution 1.
4. Both solutions have F1-score close to 1, thus, both models **have good performance**.
5. Both solutions have **almost equal and very high AUC scores** which is closer to 1. Thus, both models have great ability to differentiate between benign and malicious events.
6. Both solutions have **almost equal and very high balanced accuracy** which is closer to 1. Thus, both models have high precision and recall.
7. Both solutions **have almost equal and very high MCC** which is closer to 1. Thus, both models have very close agreements with actual labels of each event.
8. Both solutions **have very high negative predictive value**. Thus, for both models when an event was classified as benign, around 98% times it was correct and the event was actually benign (that is not malicious). Solution 2 has slightly higher negative predictive value than solution 1.
9. Both solutions **have very low false discovery rate**. Thus, for both models when an event was classified as malicious, around 1% of the times it was incorrect and the event was actually benign. Thus, both the models have noise close to 1%.
10. Both solutions have Cohen Kappa in the range of 0.81 to 0.99. Thus, the **models have near perfect agreement and are closer to the expected model**.
11. Both solutions **have ROC curve closer to axes**, and the **elbow is closer to coordinate (0, 1)** which indicates both the models have larger value of area in their respective ROC curve. Thus, both the models are good classifiers.
12. Both solutions have **Precision – Recall curve have large high value for both precision and recall**. Thus, both the models are good classifiers.

Imbalanced dataset: -

1. Both the solutions have **a very high accuracy**. Thus, both models perform correct prediction around 94% of the times.
2. Both solutions have **high precision**, greater than 86%. Thus, the model incorrectly classifies some of the normal events as malicious, resulting in false positives.
3. Both solutions have **average recall**, greater than 75%. Thus, the model incorrectly classifies multiple malicious events as benign. Solution 2 has better recall than solution 1.
4. Both solutions have **high F1-score**, around 82%. This is because the model's performance drops while correctly predicting both normal and malicious events.
5. Both solutions have **high AUC score**, around 87%. Thus, both models have strong ability to differentiate between benign and malicious events.
6. Both solutions have **high balanced accuracy**: 87%. Thus, both models have high precision and recall.
7. Both solutions have **average MCC**, around 79%. MCC score is greater than 0 and closer to 1. Thus, both models have some agreement with the actual labels.

8. Both solutions have **very high negative predictive value** around 95%. Thus, for both models when an event was classified as benign, around 95% times it was correct and the event was actually benign (that is not malicious). Solution 2 has slightly higher negative predictive value than solution 1.
9. Both solutions have **low false discovery rate** around 13%. Thus, for solution 1 if an event is classified as malicious, 13% times it is misclassified and the event was actually normal (benign). For solution 2, if an event is classified as malicious, 13% of times it is misclassified and the event was actually normal (benign).
10. Both solutions have Cohen Kappa score in the range of 0.61 and 0.80. Thus, the **models have substantial agreement with the expected model**.
11. Both solutions have **ROC curve closer to axes**, and the **elbow is closer to coordinate (0, 0.8)** which indicates both the models have relatively large value of area in their respective ROC curve and are medium fit
12. Both solutions have **Precision – Recall curve with large area**. The area of P-R curve for solution 2 is greater than the area of P-R curve for solution 1. Thus, model for solution 2 performs better than the model for solution 1.

9.3.2. Multi-class classification using Standard Scaler

Table 9.3.2 Results obtained from Multiclass classification by using FPA algorithm for feature selection and Standard scaler to scale independent features

	Balanced dataset		**Imbalanced dataset**	
	Solution 1	**Solution 2**	**Solution 1**	**Solution 2**
Number of Features	14	19	14	19
Confusion matrix	[[19980, 58, 17, 637], [145, 20539, 0, 8], [33, 0, 20659, 0], [106, 1, 1, 20584]]	[[19954, 205, 25, 508], [133, 20555, 0, 4], [71, 0, 20621, 0], [307, 1, 2, 20382]]	[[1398768, 10113, 2113, 26170], [589, 28570, 0, 4], [527, 0, 20165, 0], [[96845, 563, 20, 149654]]	[[1416627, 10478, 2477, 7582], [329, 28828, 0, 6], [496, 1, 20195, 0], [64697, 471, 24, 181890]]
Accuracy	0.988	0.985	0.921	0.95
Precision	0.989	0.988	0.838	0.918
Recall	0.995	0.992	0.669	0.779
F1-Score	0.992	0.99	0.744	0.843
Balanced accuracy	0.981	0.978	0.821	0.882
MCC	0.968	0.96	0.705	0.818
NPV	0.986	0.975	0.935	0.956
FDR	0.011	0.012	0.162	0.082
Cohen Kappa	0.984	0.98	0.707	0.819

Balanced dataset: -

1. Both solutions have **almost equal and very high accuracy**. Thus, both models perform correct prediction around 98% of the times.
2. Both solutions have **very high precision**, thus, the false positives for both models were less.
3. Both solutions have **very high recall**, thus, both models correctly classify most of the malicious events (True Positives).
4. Both solutions have **F1-score close to 1**, thus, both models have good performance.
5. Both solutions have **almost equal and very high balanced accuracy** which is closer to 1. Thus, both models have high precision and recall.
6. Both solutions have **almost equal and very high MCC** which is closer to 1. Thus**, both models have very close agreements with actual labels of each event**.
7. Both solutions have **very high negative predictive value**. Thus, for both models when an event was classified as benign, around 98% times it was correct and the event was actually benign (that is not malicious). Solution 1 has slightly higher negative predictive value than solution 2.
8. Both solutions have **very low false discovery rate**. Thus, for both models when an event was classified as malicious, around 1% of the times it was incorrect and the event was actually benign. Thus, both the models have noise close to 1%.
9. Both solutions have Cohen Kappa in the range of 0.81 to 0.99. Thus, the **models have near perfect agreement and are closer to the expected mode**l.

Imbalanced dataset: -

1. Both the solutions have a **very high accuracy.** Thus, both models perform correct prediction around 93% of the times.
2. Both solutions have **high precision**, greater than 84%. Thus, the model incorrectly classifies some of the normal events as malicious, resulting in false positives. Model for solution 2 has higher precision than the model for solution 1.
3. Both solutions have **low recall,** model for solution 1 has recall around 67% and model for solution 2 has recall around 78%. Thus, the model incorrectly classifies many malicious events as benign.
4. Both solutions have **low F1-score**, model for solution 1 has F1-score around 74% and model for solution 2 has F1-score around 84%. This is because the model's performance drops while correctly predicting both normal and malicious events.
5. Both solutions **have almost equal and high balanced accuracy** around 82%. Thus, both models have high precision and recall. However, this may be observed due to imbalanced nature of the dataset and high accuracy of both models.
6. Both solutions have **low MCC**, model for solution 1 has MCC around 70% and model for solution 2 has MCC around 82%. MCC score is greater than 0 and closer to 1. Thus, both models have some agreement with the actual labels.
7. Both solutions have **very high negative predictive value** around 94%. Thus, for both models when an event was classified as benign, around 94% times it was correct and the event was actually benign (that is not malicious). Solution 2 has slightly higher negative predictive value than solution 1.

8. Both solutions have **low false discovery rate**. Thus, for solution 1 if an event is classified as malicious, 16% times it is misclassified and the event was actually normal (benign). For solution 2, if an event is classified as malicious, 8% of times it is misclassified and the event was actually normal (benign). As the result, model for solution 2 was better than the model for solution 1.
9. Solution 1 has Cohen Kappa score in the range of 0.61 to 0.80, thus, for **solution 1** model there is **substantial agreement with the expected model**. Solution 2 has Cohen Kappa score in the range of 0.81 to 0.99, thus, for **solution 2** model there is **near perfect agreement with the expected model**.

9.3.3. Binary classification using Robust Scaler

Table 9.3.3 Results obtained from Binary classification by using FPA algorithm for feature selection and Robust scaler to scale independent features

	Balanced dataset		**Imbalanced dataset**	
	Solution 1	**Solution 2**	**Solution 1**	**Solution 2**
Number of Features	18	18	18	18
Confusion matrix	[[20301, 391], [307, 61769]]	[[20417, 275], [207, 61869]]	[[1245124, 192040], [242030, 54907]]	[[975308, 461856], [177082, 119855]]
Accuracy	0.992	0.994	0.75	0.632
Precision	0.994	0.996	0.222	0.206
Recall	0.995	0.997	0.185	0.404
F1-Score	0.994	0.996	0.202	0.273
AUC Score	0.988	0.992	0.526	0.541
Balanced accuracy	0.988	0.992	0.526	0.541
MCC	0.977	0.984	0.055	0.066
NPV	0.985	0.99	0.837	0.846
FDR	0.006	0.004	0.778	0.794
Cohen Kappa	0.977	0.984	0.055	0.06

Balanced test dataset: Comparison of ROC curves

Solution 1 **Solution 2**

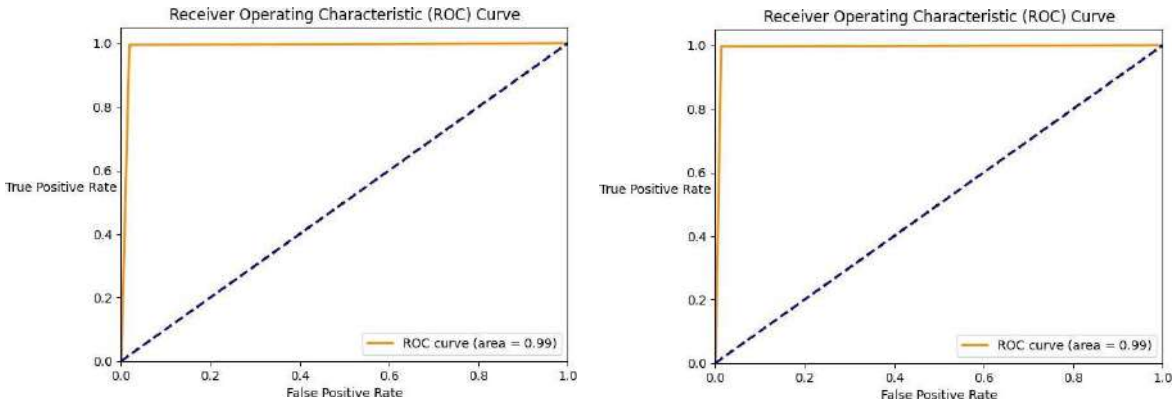


Figure 9.3.3.1 ROC curve on balanced test dataset for solution 1 and solution 2 obtained for Binary classification using FPA algorithm for feature selection and Robust scaler to scale independent features

Balanced test dataset: Comparison of Precision – Recall curves

Solution 1 **Solution 2**

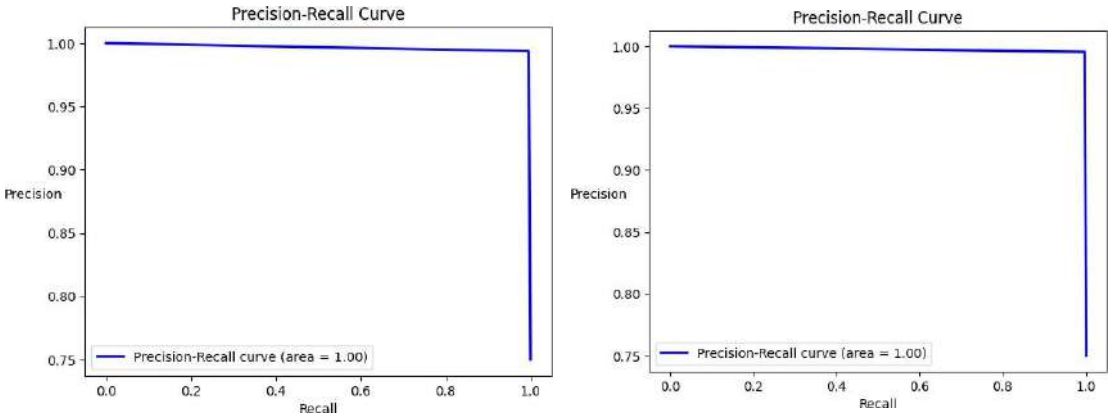


Figure 9.3.3.2 Precision - Recall curve on balanced test dataset for solution 1 and solution 2 obtained for Binary classification using FPA algorithm for feature selection and Robust scaler to scale independent features

Imbalanced test dataset: Comparison of ROC curves

Solution 1 **Solution 2**

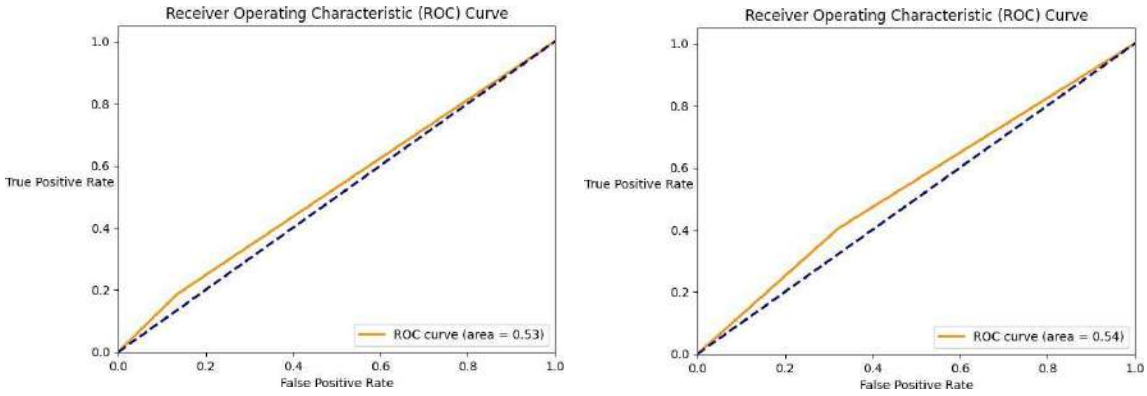


Figure 9.3.3.3 ROC curve on imbalanced test dataset for solution 1 and solution 2 obtained for Binary classification using FPA algorithm for feature selection and Robust scaler to scale independent features.

Imbalanced test dataset: Comparison of Precision – Recall curves

Solution 1 **Solution 2**

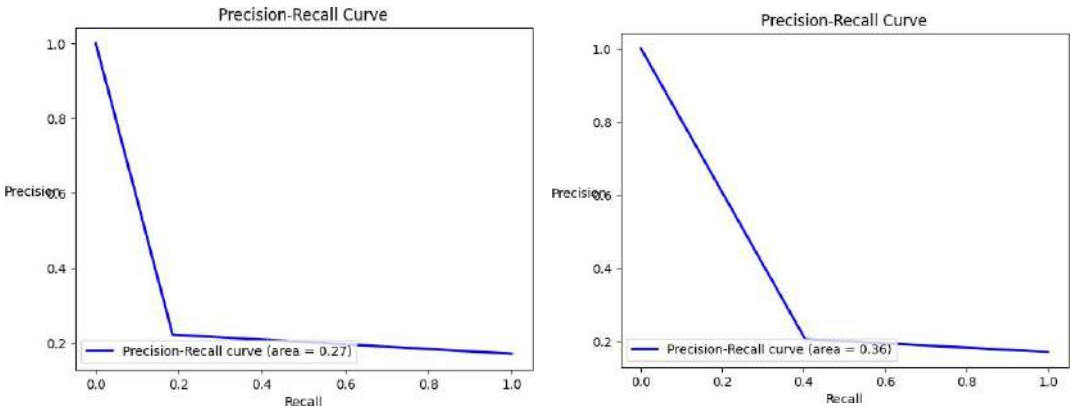


Figure 9.3.3.4 Precision - Recall curve on balanced test dataset for solution 1 and solution 2 obtained for Binary classification using FPA algorithm for feature selection and Robust scaler to scale independent features.

Balanced dataset: -

1. Both solutions have **very high accuracy**. Thus, both models perform correct prediction around 99% of the times.
2. Both solutions have **very high precision**, thus, the false positives for both models were less. Solution 2 has slightly higher precision than solution 1.
3. Both solutions have **very high recall**, thus, both models correctly classify most of the malicious events (True Positives). Solution 2 has slightly higher recall than solution 2.
4. Both solutions have **F1-score close to 1**, thus, both models have good performance.
5. Both solutions have **almost very high AUC scores** which is closer to 1. Thus, both models have great ability to differentiate between benign and malicious events.
6. Both solutions have **very high balanced accuracy** which is closer to 1. Thus, both models have high precision and recall.
7. Both solutions have **very high MCC** which is closer to 1. Thus, both models have very close agreements with actual labels of each event.
8. Both solutions have **very high negative predictive value**. Thus, for both models when an event was classified as benign, around 98% times it was correct and the event was actually benign (that is not malicious). Solution 2 has slightly higher negative predictive value than solution 1.
9. Both solutions have **very low false discovery rate**. Thus, for both models when an event was classified as malicious, less than 0.6% of the times it was incorrect and the event was actually benign. Thus, both the models have noise less than 0.6%.
10. Both solutions have Cohen Kappa in the range of 0.81 to 0.99. Thus, the models have **near perfect agreement and are closer to the expected model**.
11. Both solutions have **ROC curve closer to axes**, and the **elbow is closer to coordinate (0, 1)** which indicates both the models have larger value of area in their respective ROC curve. Thus, both the models are good classifiers.
12. Both solutions have **Precision – Recall curve have very large value** for both precision and recall. Thus, both the models are good classifiers.

Imbalanced dataset: -

1. Both the solutions have a **very low accuracy.** Thus, both models perform correct prediction in the range of 60 to 75%.
2. Both solutions have **very low precision**. Thus, the model incorrectly classifies many normal events as malicious, resulting in false positives. Model for solution 1 has better precision than the model for solution 2.
3. Both solutions have **very low recall**. Thus, the model incorrectly classifies multiple malicious events as benign. Model for solution 2 has better recall than the model for solution 1.
4. Both solutions have **low F1-score**. This is because the model's performance is significantly low while correctly predicting both normal and malicious events.
5. Both solutions have **low AUC score**. Thus, both models are inconsistent to differentiate between benign and malicious events.
6. Both solutions have **low balanced accuracy**. Thus, both models have low precision and recall.
7. Both solutions have **low MCC**. MCC score for both the solutions is positive and closer to 0, thus their **performance is very close to random guessing**.
8. Both solutions have **high negative predictive value**. Model for both solutions have NPV around 84% Thus, for both models when an event was classified as benign, around 84% times it was correct and the event was actually benign (that is not malicious). Solution 2 has slightly higher negative predictive value than solution 1.
9. Both solutions have **average false discovery rate**. Thus, for solution 1 if an event is classified as malicious, around 78% times it is misclassified and the event was actually normal (benign). For solution 2, if an event is classified as malicious, 79% of times it is misclassified and the event was actually normal (benign).
10. Both solutions have Cohen Kappa score in the range of 0 and 0.1. Thus, the **models have no agreement with the expected model**.
11. Both solutions have **very small area under the ROC curve**. For solution 1, the ROC curve is very close to random guessing line. For solution 2, the ROC curve has relatively more distance from the random guessing line. Thus, both the classifiers are very bad fit.
12. Both solutions have **Precision – Recall curve with very small area**. Thus, both models have very poor precision and recall.

9.3.4. Multi-class classification using Robust Scaler

Table 9.3.4 Results obtained from Multiclass classification by using FPA algorithm for feature selection and Robust scaler to scale independent features

	Balanced dataset		**Imbalanced dataset**	
	Solution 1	**Solution 2**	**Solution 1**	**Solution 2**
Number of Features	28	25	28	25
Confusion matrix	[[20339, 41, 25, 287], [117, 20572, 0, 3], [48, 0, 20644, 0], [163, 2, 6, 20521]]	[[20349, 51, 20, 272], [77, 20611, 0, 4], [14, 0, 20678, 0], [138, 5, 2, 20547]]	[[833874, 161274, 35569, 406447], [461, 26706, 5, 1991], [16542, 0, 0, 4150], [145300, 3734, 60, 97988]]	[[712300, 134029, 265956, 324879], [578, 27464, 68, 1053], [19610, 0, 349, 733], [91661, 2172, 859, 152390]]
Accuracy	0.992	0.993	0.556	0.516
Precision	0.994	0.994	0.171	0.199
Recall	0.995	0.996	0.434	0.617
F1-Score	0.995	0.995	0.246	0.301
Balanced Accuracy	0.989	0.99	0.507	0.556
MCC	0.978	0.982	0.011	0.085
NPV	0.984	0.989	0.837	0.864
FDR	0.006	0.006	0.829	0.801
Cohen Kappa	0.989	0.991	0.068	0.138

Balanced dataset: -

1. Both solutions have **almost equal and very high accuracy**. Thus, both models perform correct prediction around 99% of the times.
2. Both solutions have **very high precision**, thus, the false positives for both models were less.
3. Both solutions have **very high recall**, thus, both models correctly classify most of the malicious events (True Positives).
4. Both solutions have **F1-score close to 1**, thus, both models have good performance.
5. Both solutions have **almost equal and very high balanced accuracy** which is closer to 1. Thus, both models have high precision and recall.
6. Both solutions have **very high MCC** which is closer to 1. Thus, **both models have very close agreements with actual labels of each event**.
7. Both solutions have **very high negative predictive value**. Thus, for both models when an event was classified as benign, around 98% times it was correct and the event was actually benign (that is not malicious). Solution 2 has slightly higher negative predictive value than solution 1.
8. Both solutions **have equal and very low false discovery rate**. Thus, for both models when an event was classified as malicious, around 0.6% of the times it was incorrect and the event was actually benign. Thus, both the models have noise less than 1%.

9. Both solutions have Cohen Kappa in the range of 0.81 to 0.99. Thus, the **models have near perfect agreement and are closer to the expected model**.

Imbalanced dataset: -

1. Both solution 1 and solution 2 have very low accuracy. Solution 1 has accuracy around 55% and solution 2 has accuracy around 51%. Thus, overall correctness of model for solution 1 is better than the model for solution 2.
2. Both solutions have low precision. Thus, the model incorrectly classifies many normal events as malicious, resulting in false positives.
3. Both solutions have low recall. Thus, the model incorrectly classifies many malicious events as benign.
4. Both solutions have low F1-score. This is because the model's performance drops while correctly predicting both normal and malicious events.
5. Solution 1 has low balanced accuracy: 0.507, and solution 2 has relatively higher balanced accuracy: 0.556. Although solution 2 has lower precision and recall and still has higher balanced accuracy. It may be due to imbalanced nature of the dataset and its higher accuracy also supports the results.
6. Both solutions have low MCC, closer to 0. MCC values for both solutions are positive and closer to 0, which indicates that both models perform similar to random guessing.
7. Both solutions have high negative predictive value. Thus, for both models when an event was classified as benign, for solution 1 it was correct around 83% times and for solution 2 it was correct around 86% times and the event was actually benign (that is not malicious). Solution 2 has higher negative predictive value than solution 1.
8. Both solutions have high false discovery rate. Thus, for both models when if an event is classified as malicious, greater than 80% times it is misclassified and the event was actually normal (benign). As the result both models generate lots of noise.
9. Solution 1 has Cohen Kappa score less than 0.1, thus, the model for solution 1 has no agreement with the expected model. Solution 2 has Cohen Kappa score in the range of 0.10 to 0.20, thus, the model for solution 2 has slight agreement with the expected model.

9.4 Comparison of results between Balanced test datasets and Imbalanced test datasets

We observed the model's performance on balanced test datasets is very high compared to the performance on imbalanced test datasets, irrespective of the heuristic algorithms and scaling methods used.

Based on the above observations, we can infer that our models require further fine-tuning and improvement in order to deal with real-world scenarios and real-world test datasets and reduce the bias towards majority class.

Since there is very little or sometimes no difference in the performance of models on balanced test datasets, thus, the remaining comparisons will be done only on imbalanced test datasets.

9.5 Comparison of results between Standard Scaler and Robust Scaler

9.5.1. Binary classification using Artificial Bee Colony algorithm

Table 9.5.1 Comparison of results between Standard Scaler and Robust Scaler for Binary classification using ABC for feature selection

	Standard Scaler		**Robust Scaler**	
	Solution 1	**Solution 2**	**Solution 1**	**Solution 2**
Number of Features	13	11	14	16
Confusion matrix	[[1404119, 33045], [80723, 216214]]	[[1382983, 54181], [76129, 220808]]	[[824867, 612288], [120690, 176247]]	[[920408, 516756], [246720, 50217]]
Accuracy	0.934	0.925	0.577	0.56
Precision	0.867	0.803	0.224	0.089
Recall	0.728	0.744	0.594	0.169
F1-Score	0.792	0.772	0.325	0.116
AUC Score	0.853	0.853	0.584	0.405
Balanced Accuracy	0.853	0.853	0.584	0.405
MCC	0.757	0.728	0.127	-0.153
NPV	0.946	0.948	0.872	0.789
FDR	0.133	0.197	0.776	0.911
Cohen Kappa	0.753	0.727	0.101	-0.14

9.5.2. Multiclass classification using Artificial Bee Colony algorithm

Table 9.5.2 Comparison of results between Standard Scaler and Robust Scaler for Multiclass classification using ABC for feature selection

	Standard Scaler		**Robust Scaler**	
	Solution 1	**Solution 2**	**Solution 1**	**Solution 2**
Number of Features	27	30	32	25
Confusion matrix	[[1392586, 10923, 4728, 28927], [155, 28994, 2, 12], [167, 3, 20521, 1], [53871, 94, 22, 193095]]	[[1399394, 9646, 7061, 21063], [319, 28827, 2, 15], [162, 0, 20526, 4], [54696, 1163, 20, 191203]]	[[809752, 354838, 206919, 65655], [638, 27591, 757, 177], [20021, 0, 348, 323], [150550, 24460, 515, 71557]]	[[1264989, 68723, 24664, 78788], [916, 28178, 36, 33], [16201, 0, 346, 4145], [108326, 2932, 129, 135695]]

Accuracy	0.943	0.946	0.532	0.828
Precision	0.845	0.864	0.137	0.488
Recall	0.817	0.813	0.368	0.567
F1-Score	0.831	0.838	0.199	0.525
Balanced accuracy	0.893	0.894	0.465	0.724
MCC	0.797	0.807	-0.051	0.422
NPV	0.963	0.962	0.825	0.91
FDR	0.155	0.136	0.863	0.512
Cohen Kappa	0.803	0.811	0.075	0.444

9.5.3. Binary classification using Flower Pollination algorithm

Table 9.5.3 Comparison of results between Standard Scaler and Robust Scaler for Binary classification using FPA for feature selection

	Standard Scaler		**Robust Scaler**	
	Solution 1	**Solution 2**	**Solution 1**	**Solution 2**
Number of features	20	25	18	18
Confusion matrix	[[1401121, 36043], [69228, 227649]]	[[1400635, 36529], [60777, 236160]]	[[1245124, 192040], [242030, 54907]]	[[975308, 461856], [177082, 119855]]
Accuracy	0.939	0.944	0.75	0.632
Precision	0.863	0.866	0.222	0.206
Recall	0.767	0.795	0.185	0.404
F1-Score	0.812	0.829	0.202	0.273
AUC Score	0.871	0.885	0.526	0.541
Balanced accuracy	0.871	0.885	0.526	0.541
MCC	0.778	0.797	0.055	0.066
NPV	0.953	0.958	0.837	0.846
FDR	0.137	0.134	0.778	0.794
Cohen Kappa	0.776	0.796	0.055	0.06

9.5.4. Multiclass classification using Flower Pollination algorithm

Table 9.5.4 Comparison of results between Standard Scaler and Robust Scaler for Multiclass classification using FPA for feature selection

	Standard Scaler		**Robust Scaler**	
	Solution 1	**Solution 2**	**Solution 1**	**Solution 2**
Number of Features	14	19	28	25
Confusion matrix	[[1398768, 10113, 2113, 26170], [589, 28570, 0, 4], [527, 0, 20165, 0], [[96845, 563, 20, 149654]]	[[1416627, 10478, 2477, 7582], [329, 28828, 0, 6], [496, 1, 20195, 0], [64697, 471, 24, 181890]]	[[833874, 161274, 35569, 406447], [461, 26706, 5, 1991], [16542, 0, 0, 4150], [145300, 3734, 60, 97988]]	[[712300, 134029, 265956, 324879], [578, 27464, 68, 1053], [19610, 0, 349, 733], [91661, 2172, 859, 152390]]
Accuracy	0.921	0.95	0.556	0.516
Precision	0.838	0.918	0.171	0.199
Recall	0.669	0.779	0.434	0.617
F1-Score	0.744	0.843	0.246	0.301
Balanced accuracy	0.821	0.882	0.507	0.556
MCC	0.705	0.818	0.011	0.085
NPV	0.935	0.956	0.837	0.864
FDR	0.162	0.082	0.829	0.801
Cohen Kappa	0.707	0.819	0.068	0.138

Observations and interpretations from the above 4 results: -

Across all metrics, the models trained and evaluated using Standard Scaler performed relatively much better than the models trained and evaluated using Robust Scaler. In Standard Scaler, we remove the mean scale all data points to unit variance.

$z = (x - \mu)/sd$

where: -

x is the given data point

μ is the mean of the data points

sd is the standard deviation of the data points

Thus, Standard Scaler ensures that all features have same variance, enabling to manage those features which prior scaling have variance with different magnitudes.

In Robust Scaler, we remove the median of all data points and scale the data according to quantile range. It helps to ensure that the distribution of dataset remains unchanged.

However, it does not bring all features to have same variance.

In KNN algorithm, distance measure is used to find similarity of test data as per the groups identified in training data.

As the result, when we use Standard Scaler, all features have uniform influence on the distance. But, when we use Robust Scaler, features with larger variance have greater influence on the distance than the features with smaller variance.

Thus, the models trained using Standard Scaler have better results than the models trained using Robust Scaler especially when the dataset has features with varying variance.

9.6 Comparison of results between Artificial Bee Colony algorithm and Flower Pollination Algorithm

9.6.1. Binary classification using Standard Scaler

Table 9.6.1 Comparison of results between ABC and FPA for Binary classification using Standard Scaler for feature scaling

	Artificial Bee Colony Optimization		**Flower Pollination Algorithm**	
	Solution 1	**Solution 2**	**Solution 1**	**Solution 2**
Number of Features	13	11	20	25
Confusion matrix	[[1404119, 33045], [80723, 216214]]	[[1382983, 54181], [76129, 220808]]	[[1401121, 36043], [69228, 227649]]	[[1400635, 36529], [60777, 236160]]
Accuracy	0.934	0.925	0.939	0.944
Precision	0.867	0.803	0.863	0.866
Recall	0.728	0.744	0.767	0.795
F1-Score	0.792	0.772	0.812	0.829
AUC Score	0.853	0.853	0.871	0.885
Balanced accuracy	0.853	0.853	0.871	0.885
MCC	0.757	0.728	0.778	0.797
NPV	0.946	0.948	0.953	0.958
FDR	0.133	0.197	0.137	0.134
Cohen Kappa	0.753	0.727	0.776	0.796

9.6.2. Multiclass classification using Standard Scaler

Table 9.6.2 Comparison of results between ABC and FPA for Multiclass classification using Standard Scaler for feature scaling

	Artificial Bee Colony Optimization		Flower Pollination Algorithm	
	Solution 1	**Solution 2**	**Solution 1**	**Solution 2**
Number of Features	27	30	14	19
Confusion matrix	[[1392586, 10923, 4728, 28927], [155, 28994, 2, 12], [167, 3, 20521, 1], [53871, 94, 22, 193095]]	[[1399394, 9646, 7061, 21063], [319, 28827, 2, 15], [162, 0, 20526, 4], [54696, 1163, 20, 191203]]	[[1398768, 10113, 2113, 26170], [589, 28570, 0, 4], [527, 0, 20165, 0], [[96845, 563, 20, 149654]]	[[1416627, 10478, 2477, 7582], [329, 28828, 0, 6], [496, 1, 20195, 0], [64697, 471, 24, 181890]]
Accuracy	0.943	0.946	0.921	0.95
Precision	0.845	0.864	0.838	0.918
Recall	0.817	0.813	0.669	0.779
F1-Score	0.831	0.838	0.744	0.843
Balanced accuracy	0.893	0.894	0.821	0.882
MCC	0.797	0.807	0.705	0.818
NPV	0.963	0.962	0.935	0.956
FDR	0.155	0.136	0.162	0.082
Cohen Kappa	0.803	0.811	0.707	0.819

9.6.3. Binary classification using Robust Scaler

Table 9.6.3 Comparison of results between ABC and FPA for Binary classification using Robust Scaler for feature scaling

	Artificial Bee Colony Optimization		Flower Pollination Algorithm	
	Solution 1	**Solution 2**	**Solution 1**	**Solution 2**
Number of Features	14	16	18	18
Confusion matrix	[[824867, 612288], [120690, 176247]]	[[920408, 516756], [246720, 50217]]	[[1245124, 192040], [242030, 54907]]	[[975308, 461856], [177082, 119855]]
Accuracy	0.577	0.56	0.75	0.632
Precision	0.224	0.089	0.222	0.206
Recall	0.594	0.169	0.185	0.404
F1-Score	0.325	0.116	0.202	0.273
AUC Score	0.584	0.405	0.526	0.541
Balanced accuracy	0.584	0.405	0.526	0.541
MCC	0.127	-0.153	0.055	0.066

NPV	0.872	0.789	0.837	0.846
FDR	0.776	0.911	0.778	0.794
Cohen Kappa	0.101	-0.14	0.055	0.06

9.6.4. Multiclass classification using Robust Scaler

Table 9.6.4 Comparison of results between ABC and FPA for Multiclass classification using Robust Scaler for feature scaling

	Artificial Bee Colony Optimization		**Flower Pollination Algorithm**	
	Solution 1	**Solution 2**	**Solution 1**	**Solution 2**
Number of Features	32	25	28	25
Confusion matrix	[[809752, 354838, 206919, 65655], [638, 27591, 757, 177], [20021, 0, 348, 323], [150550, 24460, 515, 71557]]	[[1264989, 68723, 24664, 78788], [916, 28178, 36, 33], [16201, 0, 346, 4145], [108326, 2932, 129, 135695]]	[[833874, 161274, 35569, 406447], [461, 26706, 5, 1991], [16542, 0, 0, 4150], [145300, 3734, 60, 97988]]	[[712300, 134029, 265956, 324879], [578, 27464, 68, 1053], [19610, 0, 349, 733], [91661, 2172, 859, 152390]]
Accuracy	0.532	0.828	0.556	0.516
Precision	0.137	0.488	0.171	0.199
Recall	0.368	0.567	0.434	0.617
F1-Score	0.199	0.525	0.246	0.301
Balanced accuracy	0.465	0.724	0.507	0.556
MCC	-0.051	0.422	0.011	0.085
NPV	0.825	0.91	0.837	0.864
FDR	0.863	0.512	0.829	0.801
Cohen Kappa	0.075	0.444	0.068	0.138

Observations and interpretations from the above 4 results: -

For both classifiers, on most of the metrics the models using Flower Pollination algorithm have performed better than models using Artificial Bee Colony optimization algorithm.

Since both algorithms were executed for same number of generations, based on the above results we can infer that Flower Pollination Algorithm has faster convergence towards optimal solution than Artificial Bee Colony optimization algorithm.

The reason for Flower Pollination Algorithm performing better than Artificial Bee Colony optimization algorithm could be observed due to the method of exploring new solutions in search space.

1. In Flower Pollination Algorithm, the new solutions are explored using Levy flight, which enables to search for new solutions far away from current best solutions in lesser time.

2. However, in Artificial Bee Colony optimization algorithm, we mostly search for partner solutions in the neighbourhood of each current solution to perform Greedy search. And only in Scout Bee phase, we search for randomly new solution for one of the food sources.
   As the result, the velocity to search for optimal solutions far away from each other in the search space is relatively greater in Flower Pollination algorithm than Artificial Bee Colony optimization algorithm.

However, the sample size of results is too small to make conclusive judgement between the performance of the two algorithms.

9.7 Results observed in literature survey for the models trained using Heuristic feature selection methods

Following are the results of some of the research papers from literature survey: -

9.7.1. An Adapted Ant Colony Optimization for Feature Selection

Table 9.7.1 Results observed in paper 1 of literature survey

Dataset	ACO with SVM		ACO with KNN=5
	Accuracy	F-Score	F-Score
WDBC	0.941	0.972	0.947
Dermatology	0.968	0.975	0.953
Ionosphere	0.882	0.927	0.897
Arhythmia	0.597	0.69	0.679
Wine	0.857	0.98	0.942
Hepatitis	0.788	0.923	0.887
Spambase	0.998	0.999	0.996
Madellon	0.486	0.549	0.721

9.7.2. Multi-Label Feature Selection Based on Improved Ant Colony Optimization with Dynamic Redundancy and Label Dependence

As the number of selected features increases, average precision increases and hamming loss decreases.

9.7.3. Intrusion detection in KDD99 Dataset using SVM-PSO and Feature Reduction with Information Gain

Table 9.7.2 Results observed in paper 3 of literature survey

Attack	Accuracy	Specificity	Sensitivity
Probing	99.3	84.2	99.9
DoS	99.4	99.9	97.1
Remote to User	98.7	89.4	96.2
User to Root	98.5	25	98.6

9.7.4. Towards support-vector machine based ant colony optimization algorithms for intrusion detection

Table 9.7.3 Results of dataset 1 observed in paper 4 of literature survey

Dataset 1	Accuracy	Sensitivity	Specificity	Precision
DoS	100	99.23	99.69	100
Probe	99.95	98.26	97.85	99.23
U2R &R2L	100	99.5	98.7	99.93

Table 9.7.4 Results of dataset 2 observed in paper 4 of literature survey

Dataset 2	Accuracy	Sensitivity	Specificity	Precision
DoS	99.9	99.6	98.89	99.63
Probe	99.62	99.02	98.25	99.65
U2R &R2L	99.23	99.11	99.22	99.62

9.7.5. Evolving optimized decision rules for intrusion detection using particle swarm paradigm

Detection rate is computed using Equation (16).

Error rate is computed using Equation (17).

Detection rate=98.3

Error rate=1.7

9.7.6. An Artificial Immune System for Classification with Local Feature Selection

Table 9.7.5 Results observed in paper 6 of literature survey

Dataset	Size	Total features	Classes	Accuracy %
Ionosphere	351	34	2	93.78
Glass	214	9	6	70.79
Breast Cancer	699	9	2	96.35
Iris	150	4	4	94.67
Wine	178	13	4	97.5
Diabetes	768	8	2	73.14
Heart Stalog	270	13	2	81.67
Sonar	208	60	2	82.62
Celveland	303	13	2	81.77

9.7.7. Artificial Bee Colony Optimization for Feature Selection in Opinion Mining

Table 9.7.6 Results observed in paper 7 of literature survey

Algorithm used with ABC	Accuracy	Precision	Recall
Naïve Bayes	88.5	0.887	0.885
RIDOR	78.5	0.785	0.785
FURIA	93.75	0.938	0.938

9.7.8. Data feature selection based on Artificial Bee Colony algorithm

Table 9.7.7 Results observed in paper 8 of literature survey

Dataset	Accuracy in percentage			
	PSO	ACO	GA	ABC
Image Segmentation	94.42	92.42	94.26	91.13
Auto	68.78	72.2	69.27	82.93
Breast Cancer	73.08	73.08	73.08	75.87
Diabetes	75.65	75.65	75.65	71.48
Glass	71.03	71.03	71.03	71.5
Heart-C	83.17	80.86	80.2	83.17
Heart-Statlog	82.96	81.11	73.7	84.81
Hepatic	86.45	83.23	83.26	87.1
Iris	96.66	96.66	96.66	97.33

Labor	89.47	92.98	89.47	98.26

9.8 Comparison of results obtained with the models in literature survey.

We compared the results with outcomes of Imbalanced datasets using Standard Scaler as the scaling method.

As per results of literature survey, it was observed that for different datasets, there is large variance in outcome of metrics. For example: -

In paper 1 and paper 6: -

Datasets such as Ionsphere, Breast cancer, Wine, Dermatology, KDD99, the results are superior than the outcomes observed in the project.

But for other datasets such as Madellon, Hepatitis, Diabetes, Opinion dataset, Heart Stalog, the results are inferior than the outcomes observed in the project.

In paper 3, SVM is used with Particle Swarm Optimization, and in paper 4, SVM is used with Ant Colony optimization, both have superior results than the outcomes observed in the project.

In paper 6, Artificial Bee Colony optimization approach is used with three different algorithms, the results are inferior than the outcomes observed in the project.

Moreover, all the research papers used relatively smaller size dataset compared to the dataset size used in the project.

Most of the research papers, used 6 metrics to evaluate the performance of models: -

1. Accuracy
2. Precision
3. Recall
4. F1-Score
5. Specificity
6. Sensitivity

Conclusions

Following are the major high level outcomes learnt from the project: -

1. With the help of heuristic algorithms for feature selection, performance of threat detection classifiers can be increased such that it sets a very high benchmark for advance solutions like Deep Learning and Graph Neural Networks.

2. The choice of approach used for scaling and standardization of features plays a key role in determining the output of classifiers. Thus, it is extremely essential to study and understand the pattern of independent features and accordingly determine the scaling approach.

3. Making suitable adjustments in machine learning algorithms and heuristic algorithms helps to handle imbalanced datasets which is essential to achieve higher quality and reliable results in real-world scenarios.

4. Experimentation of heuristic algorithms with different machine learning algorithms is essential for robust evaluation and getting high quality outcomes.

5. Trade-off between number of features selected and performance of the model needs to be addressed in order to obtain optimal results.

Following are the observations from the project: -

1. Flower Pollination Algorithm performs better than Artificial Bee Colony optimization for feature selection to train binary and multiclass classifiers, due to its characteristic to search for optimal solutions far away in the search space.

2. Standard Scaler is better choice for scaling and standardization of independent features in comparison to Robust Scaler when there is large variance in values of each feature.

3. Robust visualizations and analysis of features of training dataset helps to retain key features and drop the redundant features, which enables to get faster convergence during feature selection process.

Following are the areas of future research and scope in the project: -

1. To implement other heuristic algorithms for feature selection and compare their performance and also determine the reasons for difference in outcomes.

2. To run heuristic algorithms for multiple number of iterations and then compare how the performance varies at different iteration intervals.

3. To use different machine learning algorithms and understand their impact on feature selection and performance of the classifiers.

4. To perform tests using different values of parameters in heuristic algorithms and machine learning algorithms, determine the ways to achieve optimal set of parameters in lesser time.

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
   Maurico 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

11. A Journal Paper: Pattern Recognition Letters
Safinaz AbdEl-Fattah Sayed, Emad Nabil and Amr Badr, 'A Binary Clonal Flower Pollination for Feature Selection', Pattern Recognition Letters, Pattern Recognition Letters, Vol. 77, pp. 21-27, March 2016

12. A Journal Paper: Springer International Publishing, Switzerland
Douglas Rodrigues, Xin-She Yang, Andre Nunes de Souza and Joao Paulo Papa, 'Binary Flower Pollination Algorithm and Its Application to Feature Selection', Recent Advances in Swarm Intelligence and Evolutionary Computation, Springer Cham. 2015, Vol. 585, pp. 85-100, December 2014

13. A Journal Paper: Engineering Optimization
Xin-She Yang, Mehmet Karamanoglu and Xingshi He, 'Flower Pollination Algorithm: A Novel Approach for Multiobjective Optimization', Engineering Optimization, Vol. 46, No. 9, pp. 1222-1237, August 2014

14. A Journal Paper: Neural Computing and Applications
Ersin Korkmaz and Ali Payidar Akgungor, 'Comparison of artificial bee colony and flower pollination algorithms in vehicle delay models at signalized interactions', Neural Computing and Applications, Vol. 32, No. 8, pp-3581-3597, July 2018

15. A Journal Paper: Applied Artificial Intelligence
Mridul Chawla and Manoj Duhan, 'Levy Flights in Metaheuristics Optimization Algorithms – A Review', Applied Artificial Intelligence, Vol. 32, No. 9-10, pp. 802-821, September 2018

16. A Journal Paper: International Journal of Advanced Computer Science and Applications
Muhammad Iqbal Abu Latiffi, Mohd. Ridzwan Yaakub and Ibrahim Said Ahmad, 'Flower Pollination Algorithm for Feature Selection in Tweets Sentiment Analysis', IJACSA, Vol. 13, No. 5, pp. 429-436, 2022

17. A Conference Paper: 4th International Conference on Information Systems Security and Privacy
Iman Sharafaldin, Arash Habibi and Ali A. Ghorbani, 'Towards Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization', 4th International Conference on Information Systems Security and Privacy (ICISSP), Portugal, January 2018

18. Book: Scrivener Publishing
Kuldeep Singh Kaswan, Jagit Singh Dhatterwal and Avadhesh Kumar, Swarm Intelligence An Approach from Natural to Artificial, Scrivener Publishing, Wiley, 2023

19. Book: Elsevier publications
Xin-She Yang, Nature-Inspired Optimization Algorithms, Elsevier publications, 2014

Check list of items for the Final report

a) Is the Cover page in proper format? Y / N

b) Is the Title page in proper format? Y / N

c) Is the Certificate from the Supervisor in proper format? Has it been signed? Y / N

d) Is Abstract included in the Report? Is it properly written? Y / N

e) Does the Table of Contents page include chapter page numbers? Y / N

f) Does the Report contain a summary of the literature survey? Y / N

i. Are the Pages numbered properly? Y / N

ii. Are the Figures numbered properly? Y / N

iii. Are the Tables numbered properly? Y / N

iv. Are the Captions for the Figures and Tables proper? Y / N

v. Are the Appendices numbered? Y / N

g) Does the Report have Conclusion / Recommendations of the work? Y / N

h) Are References/Bibliography given in the Report? Y / N

i) Have the References been cited in the Report? Y / N

j) Is the citation of References / Bibliography in proper format? Y / N