Correlation between Network Traffic and Security Threat Detection

Contents

- Project Topic
  - Problem Definition
  - Related Research
- Dataset Overview and Analysis
  - Dataset Introduction and Analysis Overview
  - Data Preprocessing
  - Correlation Analysis
  - Regression Analysis
- Machine Learning
  - Machine Learning Overview
  - Data Preprocessing
  - Model Training
  - Model Performance Evaluation
- Conclusion & Insights

Project Topic

Problem Definition

- Goal: Analyze the correlation between various features of network traffic data and security threats.
- Dataset: A dataset with labels for benign (normal) and security threats (malicious).
- Analysis Focus: Investigating how features like temporal changes in traffic, protocol and port characteristics, flow duration, and others contribute to security threats.
- **Hypothesis: "Security threats emerge within specific network traffic patterns"**

Related Research

A Survey on Big Data for Network Traffic Monitoring and Analysis

arXiv:2003.01648v1 [cs.DC] 3 Mar 2020

arXiv:2402.04469v1 [cs.LG] 6 Feb 2024

IoT Network Traffic Analysis with Deep Learning

Mei Liu
School of IT
Deakin University
Australia
Email: merryliuny@gmail.com

Leon Yang
School of IT
Deakin University
Australia
Email: leon.yang@deakin.edu.au

***Abstract*—As IoT networks become more complex and generate massive amounts of dynamic data, it is difficult to monitor and detect anomalies using traditional statistical methods and machine learning methods. Deep learning algorithms can process and learn from large amounts of data and can also be trained using unsupervised learning techniques, meaning they don't require labelled data to detect anomalies. This makes it possible to detect new and unknown anomalies that may not have been detected before. Also, deep learning algorithms can be automated and highly scalable; thereby, they can run continuously in the backend and make it achievable to monitor large IoT networks instantly. In this work, we conduct a literature review on the most recent works using deep learning techniques and implement a model using ensemble techniques on the KDD Cup 99 dataset. The experimental results showcase the impressive performance of our deep anomaly detection model, achieving an accuracy of over 98%.**

1. Introduction

Anomalies, namely novelties or outliers, are individuals that are far from the nominal data population. In general, an anomaly can be thought of as an observation or pattern that does not conform to the expected behaviour or follows the same patterns as the rest of the data. Anomaly Detection (AD), is the process of identifying unusual patterns in data that do not conform to a well-defined notion of normal data [1]. The process involves learning the normal behaviour of the data and then identifying instances that deviate significantly from the model through statistical methods or machine learning techniques. For example, in a time series dataset, an anomaly might be a sudden change in the trend or a spike in the data that is not consistent with the rest of the series. In a classification problem, an anomaly could be an observation that does not fit into any defined classes or is significantly different from the other observations.

Statistically, the sparsely distributed areas indicate that the probability of data occurring in a certain area is relatively low, where the data falling in can be considered to be anomalies. In Figure 1, we illustrate the anomalies in two-dimensional data space, where the clusters in blue indicate normal data and red points represent anomalies far from normal. Given a dataset X = {X1, X2, ..., Xn}, the feature dimension of each sample is D, $x_i \in R^D$. Deep Anomaly Detection (DAD) aims to learn a mapping function, which maps the original space to a new representation space $\phi(\cdot)$: $X \mapsto Z$, where $Z \in R^K (K \ll D)$. If the probability density of a sample in the dataset is less than the threshold, a small enough value, the sample is considered an anomaly and the anomaly score of the sample $\tau(\cdot)$ can be computed in the new space. Such sparse anomalies can be applicable in many areas by analysing activity patterns to detect anomalous behaviours, manage industrial resources, or ensure production security.

Figure 1. A simple example of anomalies in 2d data space as the red cross points while the blue solid points are the normal data.

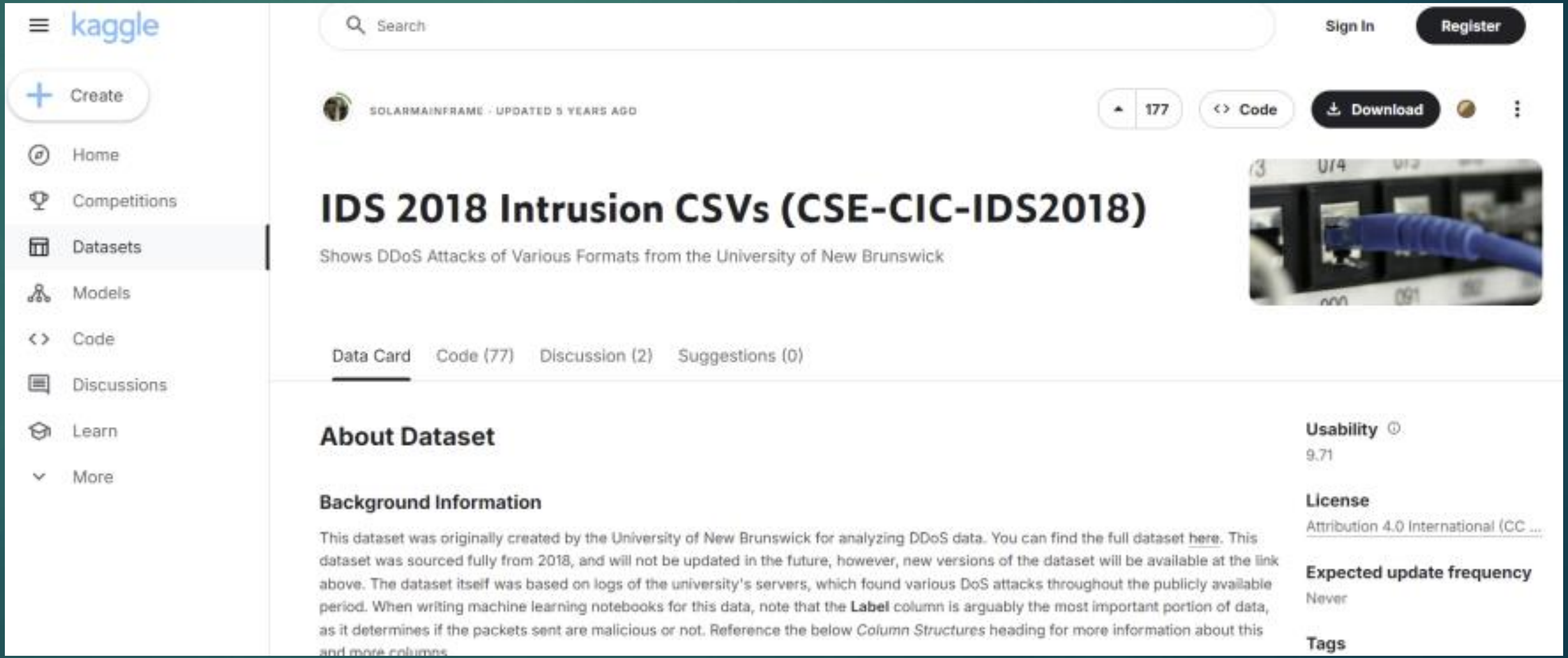
Based on the availability of data labels, we usually divide Anomaly Detection tasks into three types: supervised, semi-supervised, and unsupervised. Supervised AD uses labels of nominal and anomalous data instances to train binary or multi-class classifiers; semi-supervised techniques use the existing normal single label to separate outliers without the anomalous instances involved in the training process; in training with unsupervised deep AD techniques, there are both normal and anomalous instances in the data, but the normal instances are often much larger than the anomalies. This method detects outliers based on the intrinsic properties of the data instances and is usually used

- A Survey on Big Data for Network Traffic Monitoring and Analysis
- IoT Network Traffic Analysis with Deep Learning
- These studies suggest that security threats can be detected using data-driven methods such as big data analysis, machine learning, and deep learning.

Dataset Overview and Analysis

Dataset Introduction and Analysis Overview

- CSE-CIC-IDS2018 dataset
- Kaggle



Dataset Introduction and Analysis Overview

- Dataset Shape: (1,048,575, 80)

	Dst Port	Protocol	Timestamp	Flow Duration	Tot Fwd Pkts	Tot Bwd Pkts	TotLen Fwd Pkts	TotLen Bwd Pkts	Fwd Pkt Len Max	Fwd Pkt Len Min	...	Fwd Seg Size Min	Active Mean	Active Std	Active Max	Active Min	Idle Mean	Idle Std	Idle Max	Idle Min	Label
0	0	0	14/02/2018 08:31:01	112641719	3	0	0	0	0	0	...	0	0.0	0.0	0	0	56320859.5	139.300036	56320958	56320761	Benign
1	0	0	14/02/2018 08:33:50	112641466	3	0	0	0	0	0	...	0	0.0	0.0	0	0	56320733.0	114.551299	56320814	56320652	Benign
2	0	0	14/02/2018 08:36:39	112638623	3	0	0	0	0	0	...	0	0.0	0.0	0	0	56319311.5	301.934596	56319525	56319098	Benign
3	22	6	14/02/2018 08:40:13	6453966	15	10	1239	2273	744	0	...	32	0.0	0.0	0	0	0.0	0.000000	0	0	Benign
4	22	6	14/02/2018 08:40:23	8804066	14	11	1143	2209	744	0	...	32	0.0	0.0	0	0	0.0	0.000000	0	0	Benign
5	22	6	14/02/2018 08:40:31	6989341	16	12	1239	2273	744	0	...	20	0.0	0.0	0	0	0.0	0.000000	0	0	Benign
6	0	0	14/02/2018 08:39:28	112640480	3	0	0	0	0	0	...	0	0.0	0.0	0	0	56320240.0	203.646753	56320384	56320096	Benign
7	0	0	14/02/2018 08:42:17	112641244	3	0	0	0	0	0	...	0	0.0	0.0	0	0	56320622.0	62.225397	56320666	56320578	Benign
8	80	6	14/02/2018 08:47:14	476513	5	3	211	463	211	0	...	32	0.0	0.0	0	0	0.0	0.000000	0	0	Benign
9	80	6	14/02/2018 08:47:15	475048	5	3	220	472	220	0	...	32	0.0	0.0	0	0	0.0	0.000000	0	0	Benign

Dataset Introduction and Analysis Overview

- **Key Columns Introduction**
  - **Dst Port**: Destination port number
  - **Protocol**: Protocol number (e.g., TCP = 6, UDP = 17)
  - **Timestamp**: Time of the traffic
  - **Flow Duration**: Duration of the traffic (μs)
  - **Tot Fwd Pkts / Tot Bwd Pkts**: Number of forward / backward packets
  - **TotLen Fwd Pkts / TotLen Bwd Pkts**: Forward / backward packet length (in Bytes)
  - **Fwd Pkt Len Max / Fwd Pkt Len Min**: Maximum / Minimum forward packet length
  - **Fwd Seg Size Min**: Minimum forward segment size

Dataset Introduction and Analysis Overview

- **Additional Key Columns**
  - **Active Mean, Active Std, Active Max, Active Min**: Statistics of active transmission time (mean, standard deviation, max, min)
  - **Idle Mean, Idle Std, Idle Max, Idle Min**: Statistics of idle time (mean, standard deviation, max, min)

Dataset Introduction and **Analysis Overview**

- **Correlation Analysis**
  - **Network Traffic and Security Threat Correlation**
  - Example: Pearson, Spearman correlations, etc.
- **Regression Analysis**
  - **Univariate**: The relationship between individual features and security threats.
  - **Multivariate**: The relationship between multiple features and security threats.
- **Machine Learning**
  - Based on the variables with high correlation from the correlation analysis and regression, machine learning models will be trained and evaluated for performance.

Data Preprocessing

- **Data Cleaning:** Remove missing values, constant columns, and NaN values.
- **Labeling**: Label benign traffic as ‘Benign (0)’ and all attack traffic as ‘Attack (1)’.

```
# 결측치 제거
network_data_clean = network_data.dropna()

# 상수 열 또는 NaN 포함 열 제거
numeric_cols = numeric_cols.loc[:, numeric_cols.nunique(dropna=True) > 1]
numeric_cols = numeric_cols.dropna(axis=1)
```

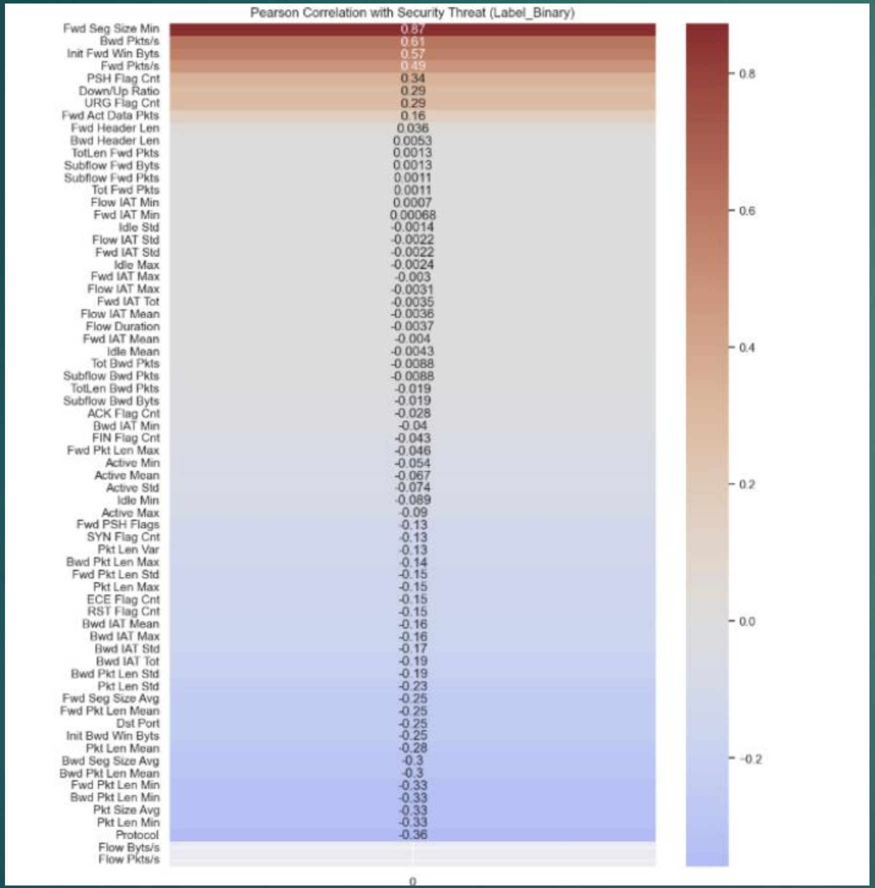
```
# 라벨링
network_data_clean.loc[:, 'Label_Binary'] = network_data_clean['Label'].apply(lambda x: 0 if x == 'Benign' else 1)
```

Correlation Analysis

- Pearson Correlation Coefficient and Spearman Correlation Coefficient Calculation

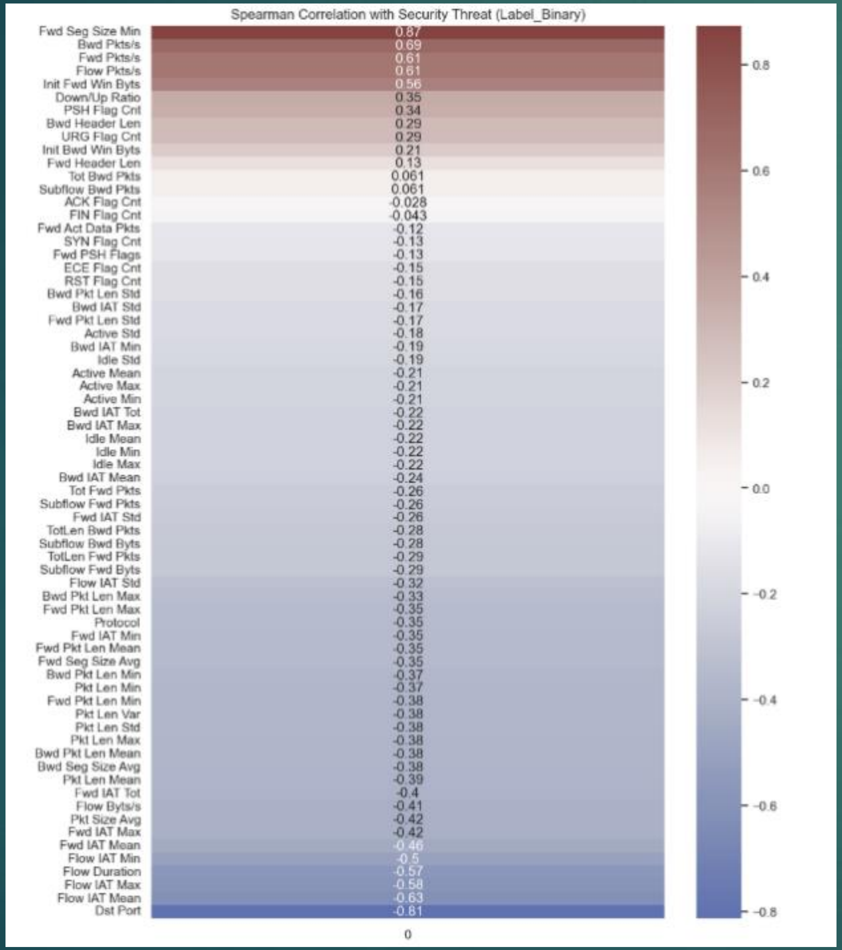
```
# ==============================================================================
# 2. 상관관계 분석
# ==============================================================================
import seaborn as sns
import matplotlib.pyplot as plt

# 수치형 피처만 선택
numeric_cols = network_data_clean.select_dtypes(include=['float64', 'int64']).drop(columns=['Label_Binary'])

# 상수열 및 NaN 열 제거
numeric_cols = numeric_cols.loc[:, numeric_cols.nunique(dropna=True) > 1]
numeric_cols = numeric_cols.dropna(axis=1)

# Pearson 상관계수
pearson_corr = numeric_cols.corrwith(network_data_clean['Label_Binary'], method='pearson')

# Spearman 상관계수
spearman_corr = numeric_cols.corrwith(network_data_clean['Label_Binary'], method='spearman')
```

Correlation Analysis



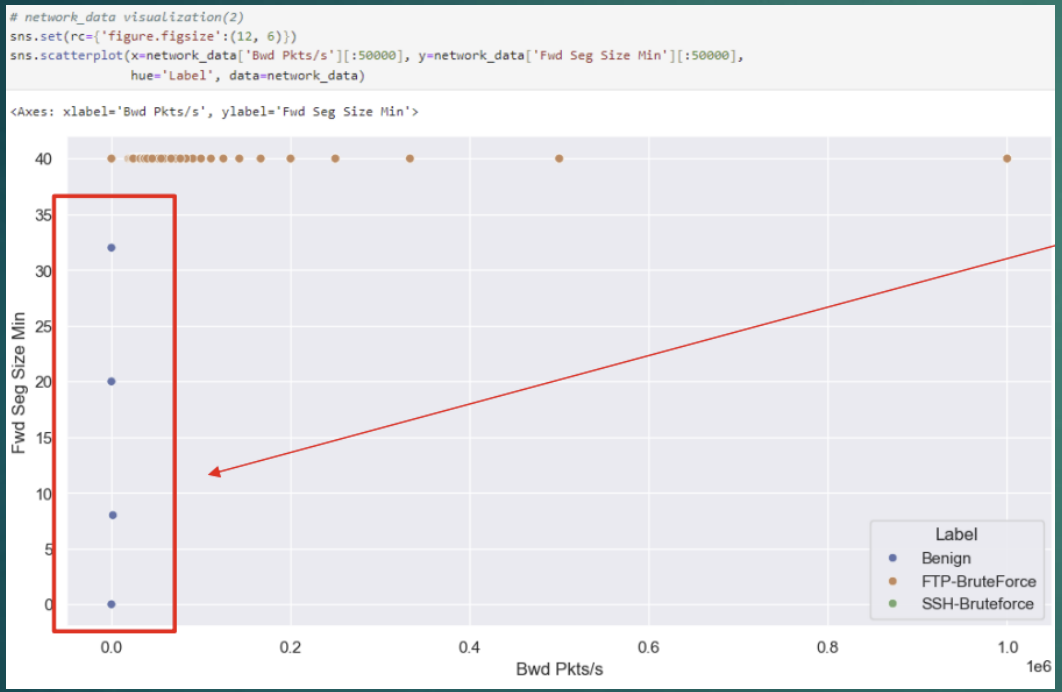
- **Pearson Correlation Coefficient Visualization**
  - **Observation**: The **Fwd Seg Size Min** variable shows the highest correlation.
  - **Additional Insights**: **Bwd Pkts/s**, **Init Fwd Win Bytes**, and **Fwd Pkts/s** also show relatively high correlations.

Correlation Analysis



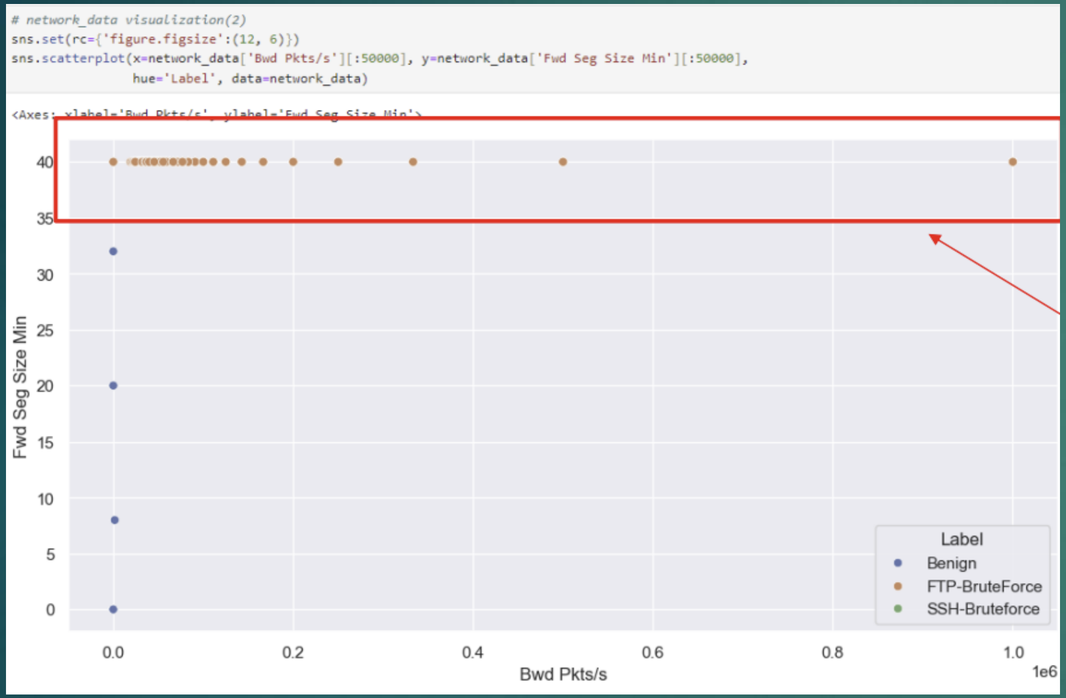
- **Spearman Correlation Coefficient Visualization**
  - **Observation**: The **Fwd Seg Size Min** variable again shows the highest correlation.
  - **Additional Insights**: **Bwd Pkts/s** and **Fwd Pkts/s** also demonstrate relatively high correlations.

Correlation Analysis



- **Distribution of Fwd Seg Size Min and Bwd Pkts/s for Normal vs Attack Networks**
- **Normal Network (Blue)**:
  - **Bwd Pkts/s** is mostly distributed at lower values.
  - **Fwd Seg Size Min** is distributed across the entire range.
- **Interpretation**: Normal traffic tends to have slow reception speeds, and the size of transmitted segments varies widely.

Correlation Analysis



- **Attack Network (Orange):**
  - **Bwd Pkts/s** varies across a broader range.
  - **Fwd Seg Size Min** tends to concentrate around a value of 40.
- **Interpretation**: Attack traffic often has a fixed segment size (40), and the reception speed is inconsistent.

Regression Analysis

- **Perform Regression Analysis on the Top 10 Features with High Correlation**
- **Regression Coefficient (coef)**: Represents how much the log odds of an attack change when the corresponding feature increases by one unit.
- **P-value**: Represents the statistical significance, where a value below 0.05 indicates statistical significance.
- **Odds Ratio**: The exponentiated value of the regression coefficient.

Regression Analysis

	feature	coef	p-value	odds_ratio
0	Fwd Seg Size Min	0.985981	0.000000	2.680441e+00
1	Bwd Pkts/s	0.000034	0.000000	1.000034e+00
2	Init Fwd Win Byts	0.000119	0.000000	1.000119e+00
3	Fwd Pkts/s	0.000007	0.000000	1.000007e+00
4	Protocol	-0.307995	0.000000	7.349189e-01
5	PSH Flag Cnt	1.540149	0.000000	4.665285e+00
7	Pkt Size Avg	-0.015790	0.000000	9.843339e-01
8	Bwd Pkt Len Min	-2.019751	0.002343	1.326886e-01
6	Pkt Len Min	-11.014815	0.935002	1.645609e-05
9	Fwd Pkt Len Min	-17.788376	0.962523	1.881943e-08

- **Regression Analysis Results Table**
- **Coef:**
  - A positive coefficient (+) indicates that as the feature increases, the model is more likely to predict an attack (1).
  - A negative coefficient (-) indicates that as the feature increases, the model is more likely to predict a normal network (0).
  - The feature **Fwd Seg Size Min** has the highest absolute coefficient, showing a strong correlation with attacks (1).

Regression Analysis

	feature	coef	p-value	odds_ratio
0	Fwd Seg Size Min	0.985981	0.000000	2.680441e+00
1	Bwd Pkts/s	0.000034	0.000000	1.000034e+00
2	Init Fwd Win Byts	0.000119	0.000000	1.000119e+00
3	Fwd Pkts/s	0.000007	0.000000	1.000007e+00
4	Protocol	-0.307995	0.000000	7.349189e-01
5	PSH Flag Cnt	1.540149	0.000000	4.665285e+00
7	Pkt Size Avg	-0.015790	0.000000	9.843339e-01
8	Bwd Pkt Len Min	-2.019751	0.002343	1.326886e-01
6	Pkt Len Min	-11.014815	0.935002	1.645609e-05
9	Fwd Pkt Len Min	-17.788376	0.962523	1.881943e-08

- **Regression Analysis Results Table (Continued)**
- **Odds Ratio**:
  - For **Fwd Seg Size Min**, a 1-unit increase in segment size increases the likelihood of an attack network by approximately **2.68 times**.

Regression Analysis

	feature	coef	p-value	odds_ratio
0	Fwd Seg Size Min	0.985981	0.000000	2.680441e+00
1	Bwd Pkts/s	0.000034	0.000000	1.000034e+00
2	Init Fwd Win Byts	0.000119	0.000000	1.000119e+00
3	Fwd Pkts/s	0.000007	0.000000	1.000007e+00
4	Protocol	-0.307995	0.000000	7.349189e-01
5	PSH Flag Cnt	1.540149	0.000000	4.665285e+00
7	Pkt Size Avg	-0.015790	0.000000	9.843339e-01
8	Bwd Pkt Len Min	-2.019751	0.002343	1.326886e-01
6	Pkt Len Min	-11.014815	0.935002	1.645609e-05
9	Fwd Pkt Len Min	-17.788376	0.962523	1.881943e-08

- **Regression Analysis Results Table (Continued)**
- **Odds Ratio**:
  - For **PSH Flag Cnt**, adding 1 more PSH flag increases the likelihood of an attack network (1) by approximately **4.7 times**.

Note

- **Correlation and Regression Analysis Insights**
  - **Fwd Seg Size Min** is the most strongly correlated feature with the attack network (1).
- Based on this, machine learning classification can be performed using **Fwd Seg Size Min** to distinguish between normal and attack networks.

Machine Learning

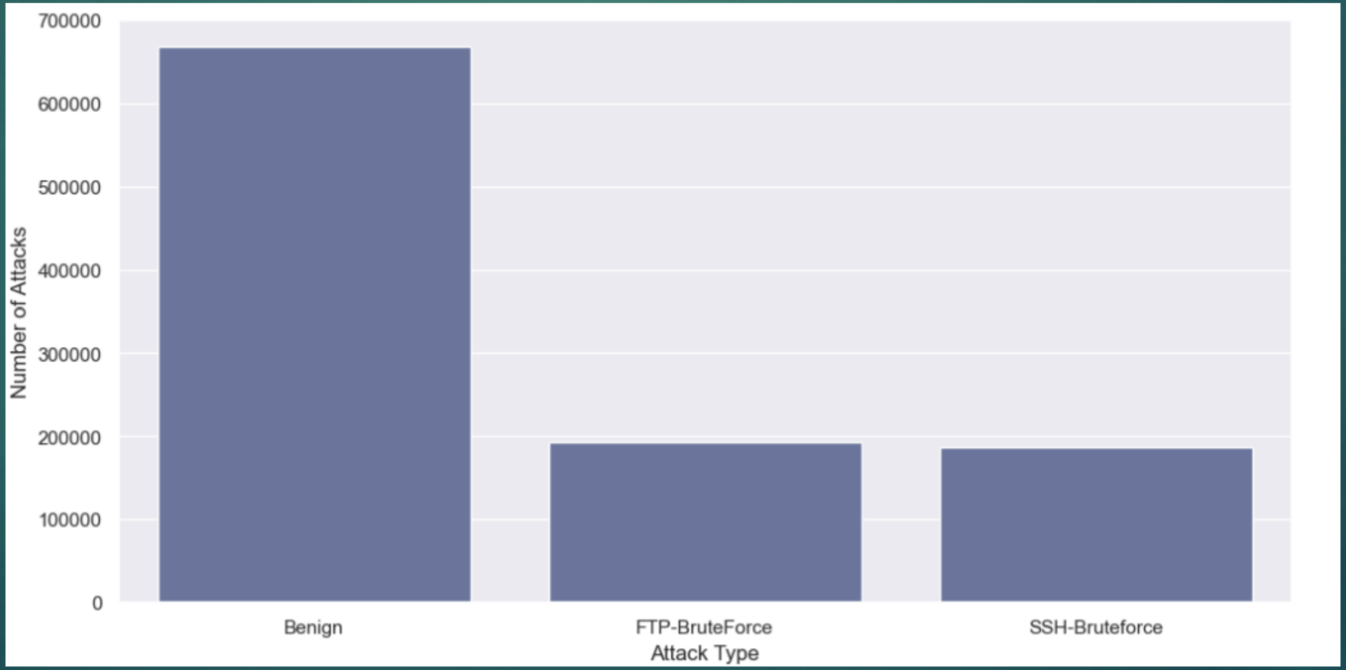
Data Preprocessing

- Remove columns with missing or NaN values.
- Label the data as:
  - **0**: Normal Network (Benign)
  - **1, 2**: Attack Networks

```
# encode the column labels
label_encoder = LabelEncoder()
cleaned_data.loc[:, 'Label'] = label_encoder.fit_transform(cleaned_data['Label'])
cleaned_data['Label'].unique()

array([0, 1, 2], dtype=object)

# encoded labels
cleaned_data['Label'].value_counts()

Label
0    665355
1    193354
2    187589
Name: count, dtype: int64
```

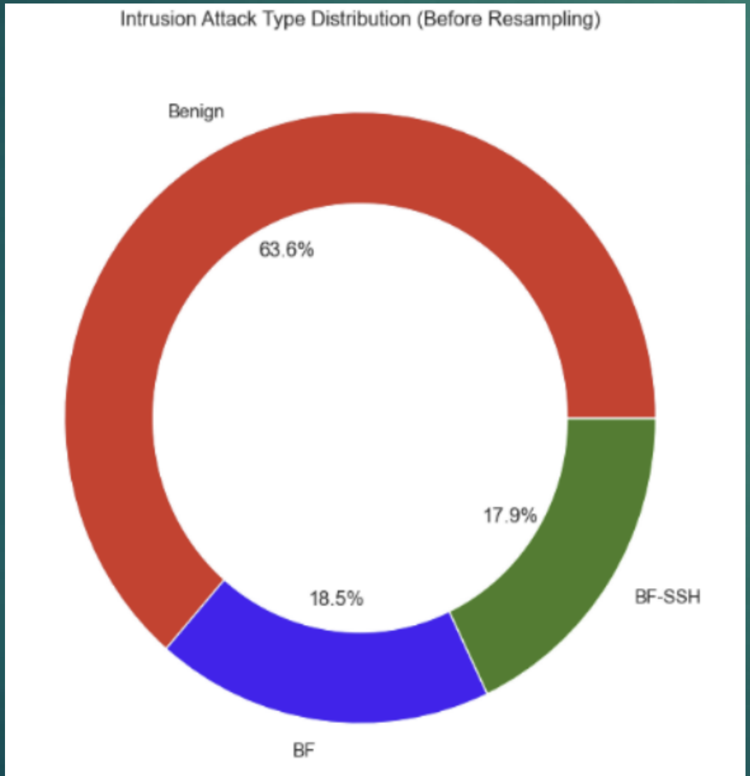
Data Preprocessing

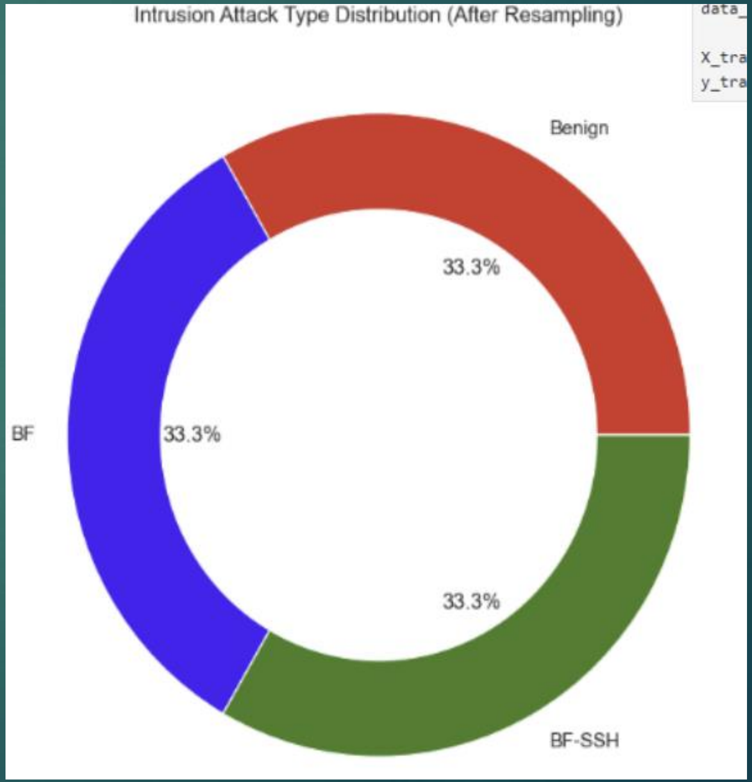
- The dataset has a higher number of normal networks compared to attack networks, so resampling is performed to balance the class distribution.



Data Preprocessing

- Resample the dataset





Data Preprocessing

- Remove unnecessary columns and scale features to a [0, 1] range.
- Perform **one-hot encoding** to create binary columns:
  - **Class 0 (Benign)**: [1, 0, 0]
  - **Class 1 (FTP-BruteForce)**: [0, 1, 0]
  - **Class 2 (SSH-Bruteforce)**: [0, 0, 1]

Model Training

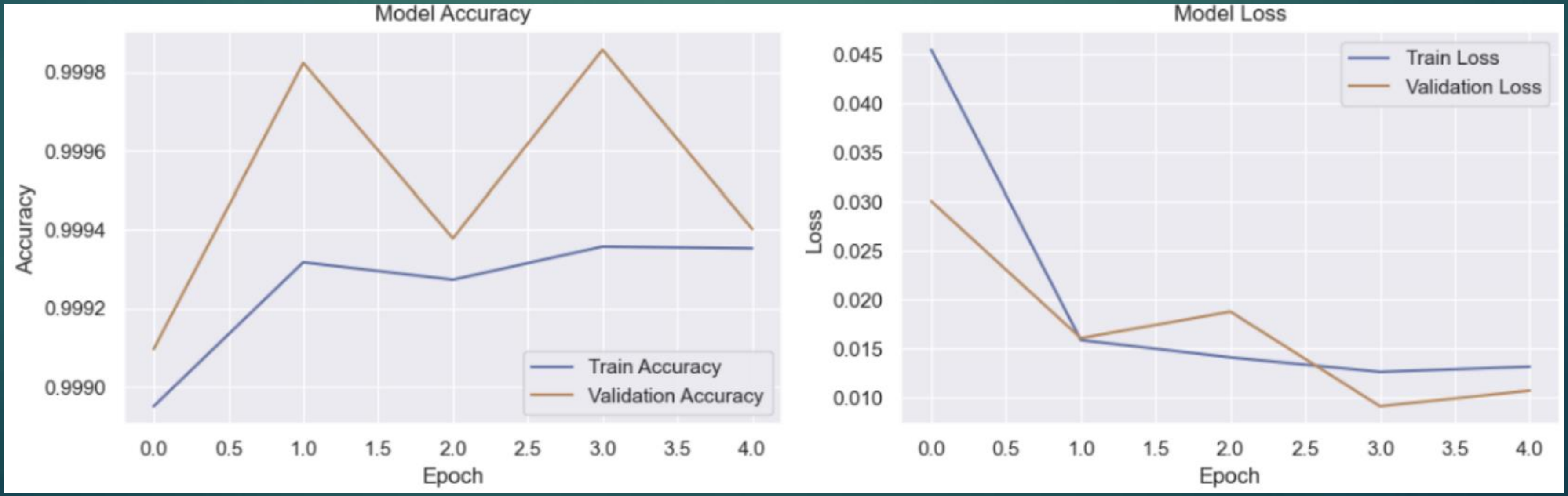
Layer (type)	Output Shape	Param #
conv1d (Conv1D)	(None, 73, 32)	128
batch_normalization (BatchNormalization)	(None, 73, 32)	128
max_pooling1d (MaxPooling1D)	(None, 36, 32)	0
conv1d_1 (Conv1D)	(None, 36, 64)	6,208
batch_normalization_1 (BatchNormalization)	(None, 36, 64)	256
max_pooling1d_1 (MaxPooling1D)	(None, 18, 64)	0
flatten (Flatten)	(None, 1152)	0
dense (Dense)	(None, 64)	73,792
dropout (Dropout)	(None, 64)	0
dense_1 (Dense)	(None, 3)	195

Total params: 80,707 (315.26 KB)
Trainable params: 80,515 (314.51 KB)
Non-trainable params: 192 (768.00 B)

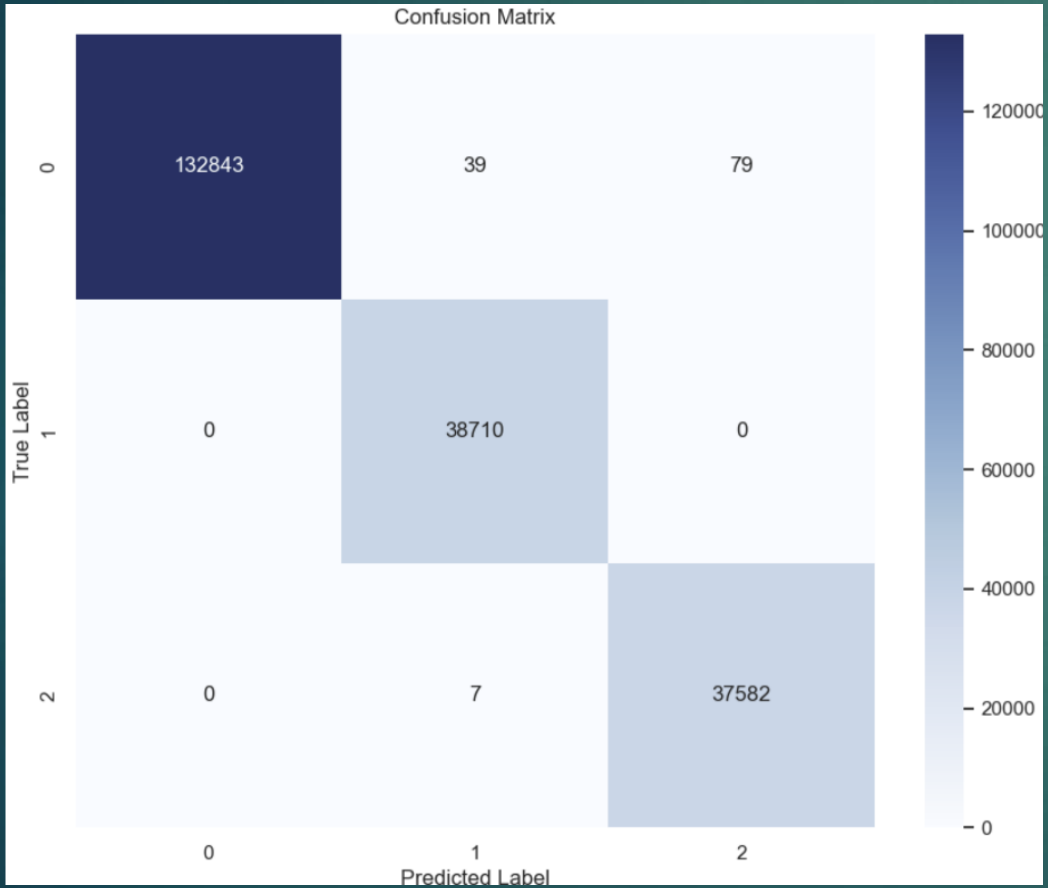
- **Model Information**
  - A **1D CNN model** is used with two convolution blocks to classify normal networks (0) and attack networks (1 - brute force, 2 - brute force SSH).
- **Model Training**

Model Performance Evaluation

- The overall **accuracy** of the model is **99.94%.**
- **Training and Validation Accuracy** visualization:
  - Both training and validation accuracy stabilize between **0.9990 to 0.9998** between epochs 1–5.
  - The **loss** steadily decreases as the epochs progress.



Model Performance Evaluation



- **Confusion Matrix Visualization**:
  - **Normal (Class 0)**: Out of 132,961 samples, 132,843 were classified correctly (118 misclassified).
  - **Attack 1 (Class 1)**: 38,710 samples classified correctly.
  - **Attack 2 (Class 2)**: Out of 37,589 samples, 37,582 were classified correctly (7 misclassified).
- The classification accuracy is high.

Conclusion & Insights

- **Hypothesis: "Security threats emerge within certain network traffic patterns"**
- **Verification**:
  - "Network attacks can be distinguished from normal networks using **Fwd Seg Size Min**. When the transmission segment size is fixed (Fwd Seg Size Min is concentrated at specific values) or the reception speed is inconsistent (Bwd Pkts/s varies widely), attacks are more likely to occur.“
  - "Additionally, attacks are more likely to appear in structures where the **PSH** (Push) flag count is high, indicating immediate forwarding of received data to the application."

Conclusion & Insights

- Based on the correlation and regression analysis results, **Fwd Seg Size** Min is most effective for distinguishing between attack and normal networks.
- **Technical Indicators Highly Associated with Attack Networks**:
  - **Fwd Seg Size Min** (minimum size of forward direction packets)
  - **Bwd Pkts/s** (packets per second in the backward direction)
  - **PSH Flag Cnt** (number of packets with PSH flag)