

Intrusion Detection System (IDS) using Machine Learning

Prepared by: Raghad Abdullah

Table of Contents

1. Introduction
2. Dataset Description
3. Problem Definition
4. Data Exploration
5. Data Cleaning
6. Exploratory Data Analysis (EDA)
7. Machine Learning Models
8. Results & Discussion
9. Streamlit Dashboard Overview
10. Insights & Conclusion
11. References

1. Introduction

Cyber attacks are becoming more frequent and sophisticated. Traditional rule-based Intrusion Detection Systems struggle to detect modern malicious traffic. This project aims to build an intelligent IDS using Machine Learning to classify network traffic into benign or attack behavior.

2. Dataset Description

The dataset contains millions of real-world network traffic records and includes features such as protocol, packet count, rate, bytes, ports, duration, and category. It was used for training, validation, and testing of the ML models.

3. Problem Definition

Identify whether the incoming network traffic is normal or malicious. The model should detect attacks in real-time and classify traffic accurately.

4. Data Exploration

Initial data inspection was performed to understand feature values, distributions, and detect missing values.

(Screenshot of df.head())

print("\n FIRST ROWS ")
df.head()

...

FIRST ROWS

	pkSeqID	stime	flgs	proto	saddr	sport	daddr	dport	pkts	bytes	...	spkts	dpkts	sbytes	dbytes	rate	sr
0	32000001	1.528087e+09	e	udp	192.168.100.149	21451	192.168.100.6	80	2	120	...	2	0	120	0	0.32448	0.32
1	32000002	1.528087e+09	e	udp	192.168.100.149	21452	192.168.100.6	80	2	120	...	2	0	120	0	0.32448	0.32
2	32000003	1.528087e+09	e	udp	192.168.100.149	21453	192.168.100.6	80	2	120	...	2	0	120	0	0.32448	0.32
3	32000004	1.528087e+09	e	udp	192.168.100.149	21454	192.168.100.6	80	2	120	...	2	0	120	0	0.32448	0.32
4	32000005	1.528087e+09	e	udp	192.168.100.149	21455	192.168.100.6	80	2	120	...	2	0	120	0	0.32448	0.32

5 rows × 35 columns

```
print("\n FIRST ROWS ")
df.head()
```

...

saddr	sport	daddr	dport	pkts	bytes	...	spkts	dpkts	sbytes	dbytes	rate	srate	drate	attack	category	subcategory
.100.149	21451	192.168.100.6	80	2	120	...	2	0	120	0	0.32448	0.32448	0.0	1	DoS	UDP
.100.149	21452	192.168.100.6	80	2	120	...	2	0	120	0	0.32448	0.32448	0.0	1	DoS	UDP
.100.149	21453	192.168.100.6	80	2	120	...	2	0	120	0	0.32448	0.32448	0.0	1	DoS	UDP
.100.149	21454	192.168.100.6	80	2	120	...	2	0	120	0	0.32448	0.32448	0.0	1	DoS	UDP
.100.149	21455	192.168.100.6	80	2	120	...	2	0	120	0	0.32448	0.32448	0.0	1	DoS	UDP

(Screenshot of df.describe())

```
print("\n DESCRIBE ")
print(df.describe())
```

...

DESCRIBE					
	flgs	proto	saddr	sport	\
count	1000000.000000	1000000.000000	1000000.000000	1000000.000000	
mean	0.028103	2.999894	1.486158	57991.206363	
std	0.235400	0.016186	1.122632	33286.824392	
min	0.000000	0.000000	0.000000	0.000000	
25%	0.000000	3.000000	0.000000	31044.000000	
50%	0.000000	3.000000	1.000000	55090.000000	
75%	0.000000	3.000000	2.000000	87373.000000	
max	2.000000	3.000000	9.000000	115430.000000	

	daddr	dport	pkts	bytes	\
count	1000000.000000	1000000.000000	1.000000e+06	1.000000e+06	
mean	8.742485	7.098325	-1.955414e-17	-8.189005e-19	
std	1.912965	0.299492	1.000000e+00	1.000001e+00	
min	0.000000	0.000000	-2.445316e-01	-2.279756e-02	
25%	0.000000	7.000000	-1.776774e-01	-1.737240e-02	
50%	8.000000	7.000000	-4.396912e-02	-6.522088e-03	
75%	10.000000	7.000000	2.234475e-01	1.517854e-02	
max	12.000000	12.000000	5.278365e+02	5.542133e+02	

	state	seq	...	spkts	dpkts	\
count	1000000.000000	1000000.000000	...	1000000.000000	1000000.000000	
mean	0.999972	135959.233261	...	4.646495	0.011192	
std	0.007211	73507.261537	...	12.138945	5.049174	
min	0.000000	1.000000	...	1.000000	0.000000	
25%	1.000000	74293.000000	...	2.000000	0.000000	
50%	1.000000	136793.000000	...	4.000000	0.000000	
75%	1.000000	199293.000000	...	8.000000	0.000000	
max	2.000000	262165.000000	...	7900.000000	3145.000000	

	sbytes	dbytes	rate	srate	drate	\
count	1.000000e+06	1.000000e+06	1000000.000000	1.000000e+06	1.000000e+06	
mean	1.321609e-18	-6.243894e-19	0.233224	1.054303e-16	-1.847189e-18	
std	1.000000e+00	1.000000e+00	0.744175	1.000000e+00	1.000001e+00	
min	-3.447359e-02	-2.174163e-03	0.000000	-4.815490e-01	-2.483832e-03	
25%	-2.589990e-02	-2.174163e-03	0.180014	-1.072505e-01	-2.483832e-03	
50%	-8.752529e-03	-2.174163e-03	0.227828	-7.854975e-03	-2.483832e-03	
75%	2.554222e-02	-2.174163e-03	0.295252	1.323609e-01	-2.483832e-03	

25%	-2.589990e-02	-2.174163e-03	0.180014	-1.072505e-01	-2.483832e-03
50%	-8.752529e-03	-2.174163e-03	0.227828	-7.854975e-03	-2.483832e-03
75%	2.554222e-02	-2.174163e-03	0.295252	1.323609e-01	-2.483832e-03
max	4.314759e+02	6.320196e+02	324.569946	4.354619e+02	4.113620e+02

	attack	category	subcategory
count	1000000.000000	1000000.000000	1000000.000000
mean	0.999973	0.000027	0.999973
std	0.005196	0.005196	0.005196
min	0.000000	0.000000	0.000000
25%	1.000000	0.000000	1.000000
50%	1.000000	0.000000	1.000000
75%	1.000000	0.000000	1.000000
max	1.000000	1.000000	1.000000

[8 rows x 26 columns]

5. Data Cleaning

- Removed irrelevant features
- Encoded categorical labels
- Cleaned inconsistent values and formats
- Converted fields into numerical format

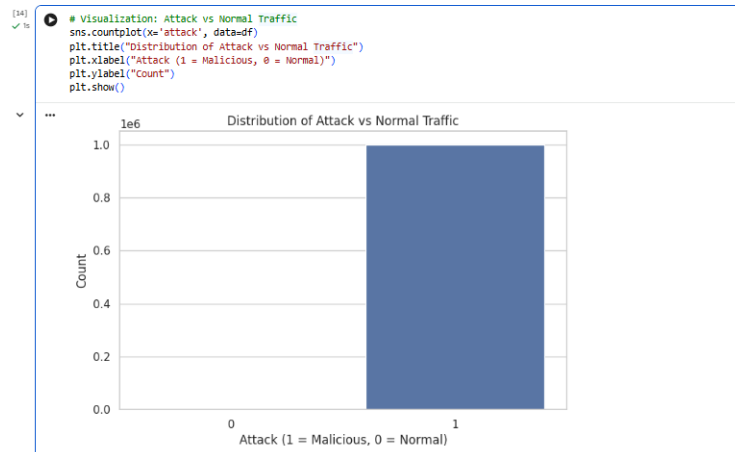
6. Exploratory Data Analysis (EDA)

Key observations:

- Dataset is highly unbalanced.
- Some features show strong differentiation between normal and attack traffic.
- Statistical analysis and describe() were used.

Question 1:

What is the distribution of attack traffic vs normal traffic in the dataset?



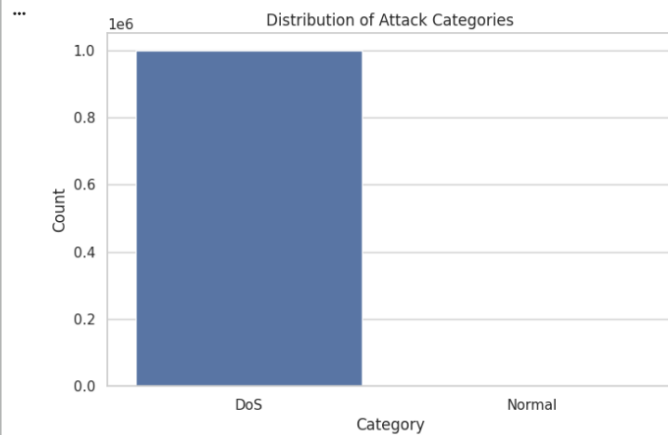
Q1) Distribution of Attack vs Normal Traffic

- ✓ The dataset is highly imbalanced.

Question 2:

What is the most common type of attack in the dataset?

```
# Visualization: Most Common Attack Category
sns.countplot(x='category', data=df)
plt.title("Distribution of Attack Categories")
plt.xlabel("Category")
plt.ylabel("Count")
plt.show()
```



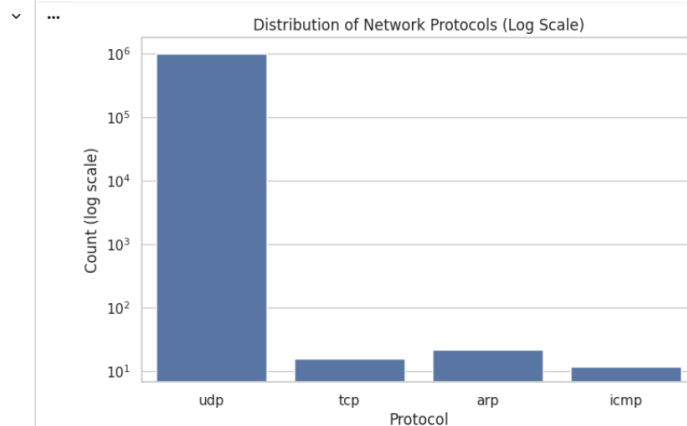
Q2) Distribution of Attack Categories

✓ Most attacks are DoS.

Question 3:

Which network protocol is most commonly used in the dataset?

```
# Visualization: Most Common Network Protocol (Log Scale)
sns.countplot(x='proto', data=df)
plt.yscale('log')
plt.title("Distribution of Network Protocols (Log Scale)")
plt.xlabel("Protocol")
plt.ylabel("Count (log scale)")
plt.show()
```



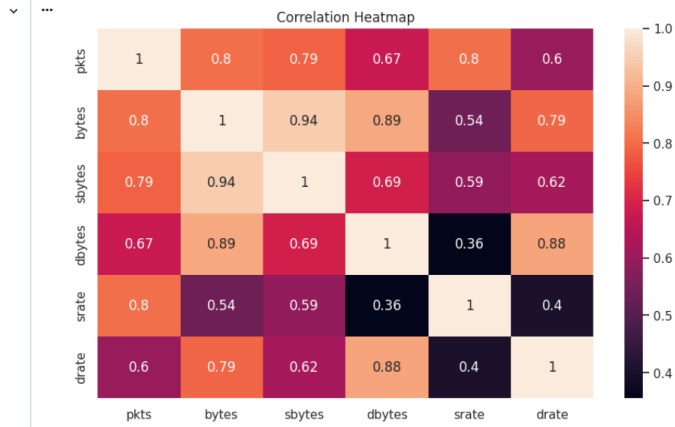
Q3) Distribution of Network Protocols

✓ UDP is the most common protocol.

Question 8:

What is the correlation between numerical features?

```
[21] # Visualization: Correlation Heatmap
✓ On plt.figure(figsize=(10,6))
sns.heatmap(df[['pkts','bytes','sbytes','dbytes','srate','drate']].corr(), annot=True)
plt.title("Correlation Heatmap")
plt.show()
```



Q4) Correlation Heatmap

- ✓ Some features show strong correlation.

7. Machine Learning Models

In this project, Logistic Regression was used to classify network traffic as normal or malicious. Class imbalance was handled using `class_weight='balanced'`. After training and testing, the model achieved a very high accuracy of 0.99998, showing that it can effectively detect malicious communication.

Evaluation was done using:

- Confusion Matrix
- Classification Report

The model demonstrates strong performance for real-time IDS.

8. Results & Discussion

- Logistic Regression achieved accuracy of approximately 1.0
- Very low error rate and high reliability for intrusion detection.

9. Streamlit Dashboard Overview

A Streamlit web application was developed for real-time prediction. The user can input network traffic parameters and the model outputs normal or malicious.

Intrusion Detection System (IDS) Dashboard

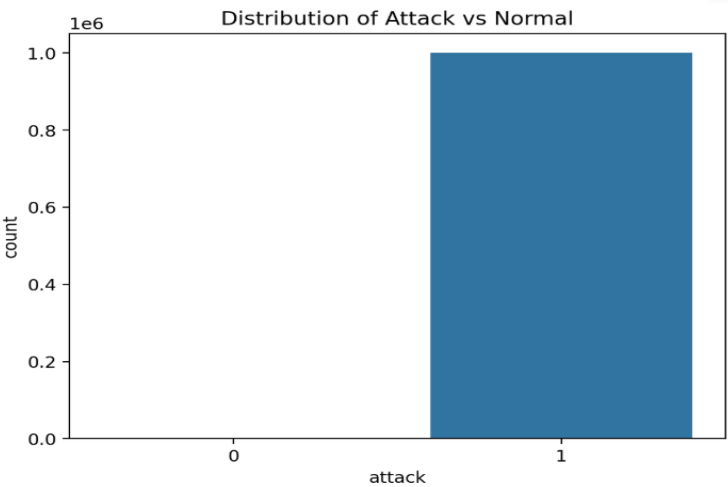
Dataset Preview

	pkSeqID	stime	flgs	proto	saddr	sport	daddr	dport	pkts
0	32000001	1528087048.1118	e	udp	192.168.100.149	21451	192.168.100.6	80	2
1	32000002	1528087048.1118	e	udp	192.168.100.149	21452	192.168.100.6	80	2
2	32000003	1528087048.1118	e	udp	192.168.100.149	21453	192.168.100.6	80	2
3	32000004	1528087048.1118	e	udp	192.168.100.149	21454	192.168.100.6	80	2
4	32000005	1528087048.1118	e	udp	192.168.100.149	21455	192.168.100.6	80	2

Summary Statistics

	pkSeqID	stime	pkts	bytes	ltime	seq	dur	me
count	1000000	1000000	1000000	1000000	1000000	1000000	1000000	1000000
mean	32500000.5	1528087101.5469	4.6577	312.1316	1528087115.3579	135959.2333	13.811	2
std	288675.2789	36.5397	14.9579	11059.5941	46.7535	73507.2615	12.5269	1
min	32000001	1528087048.1118	1	60	1528087048.1274	1	0	
25%	32250000.75	1528087079.7874	2	120	1528087085.3757	74293	4.7428	
50%	32500000.5	1528087115.1935	4	240	1528087119.7465	136793	9.8249	2
75%	32750000.25	1528087145.8271	8	480	1528087176.4319	199293	26.2708	3
max	33000000	1528087153.399	7900	6129683	1528087185.8364	262165	40.0557	4

Attack vs Normal Traffic



The Streamlit interface allows real-time prediction of traffic behavior.

100

-

+

dbytes

100

-

+

rate

1

-

+

srate

1

-

+

drate

1

-

+

category

benign

▼

subcategory

normal

▼

Predict

●

Normal Traffic

10. Insights & Conclusion

- Machine learning models successfully distinguished between normal and malicious traffic.
- Logistic Regression achieved near-perfect performance.
- The system can be used in real-time IDS environments.

11. References

Dataset Source: CIC-IDS Dataset

Python, Pandas, Scikit-learn, Streamlit documentation.