

# AI-BASED NETWORK INTRUSION DETECTION SYSTEM

## Project Documentation

### INTRODUCTION

With the rapid growth of internet technologies, computer networks have become an essential part of modern communication, business operations, and data sharing. However, this growth has also led to a significant increase in cyber threats such as Distributed Denial of Service (DDoS) attacks, port scanning, infiltration attacks, and unauthorized access. These attacks can compromise sensitive information, disrupt network services, and cause financial losses to organizations.

Traditional security mechanisms like firewalls and signature-based intrusion detection systems are no longer sufficient to protect modern networks. These systems rely on predefined rules and known attack patterns, which makes them ineffective against new or evolving cyber threats. As a result, there is a growing need for intelligent and adaptive security systems.

This project focuses on developing an AI-Based Network Intrusion Detection System (NIDS) that uses machine learning techniques to analyze network traffic and identify malicious activities. By learning patterns from real-world network data, the system can automatically detect abnormal behavior and improve network security.

### PROBLEM STATEMENT

Modern computer networks generate massive volumes of traffic every second. Manually monitoring this traffic for suspicious behavior is impractical and error-prone. Traditional intrusion detection systems are limited in their ability to detect new or unknown attacks due to their dependency on static rules and signatures.

The key challenges addressed in this project are:

- Detecting malicious network traffic in large datasets
- Differentiating between normal and attack traffic accurately
- Providing understandable explanations for detection results
- Creating an interactive and user-friendly monitoring system

Therefore, the problem is to design and implement an intelligent network intrusion detection system that can efficiently analyze network traffic, detect cyberattacks, and assist users in understanding the results.

## PROJECT OVERVIEW

The AI-Based Network Intrusion Detection System is developed using machine learning techniques and real-world network traffic data. The project uses the **CIC-IDS2017 dataset**, which contains labeled network flows representing both benign and various attack scenarios such as DDoS, port scanning, and brute-force attacks.

A **Random Forest classifier** is used to train the intrusion detection model. Random Forest is chosen due to its robustness, high accuracy, and ability to handle large datasets. The model is trained on selected flow-level statistical features extracted from network traffic.

The system also includes a **web-based dashboard built using Streamlit**, allowing users to interact with the model, simulate network traffic, and view intrusion detection results in real time. To improve transparency, the system integrates an **AI explanation module using Groq LLM**, which explains why a particular traffic flow is classified as an attack or normal traffic.

The complete application is deployed on **Hugging Face Spaces**, making it accessible through a web browser without requiring local installation.

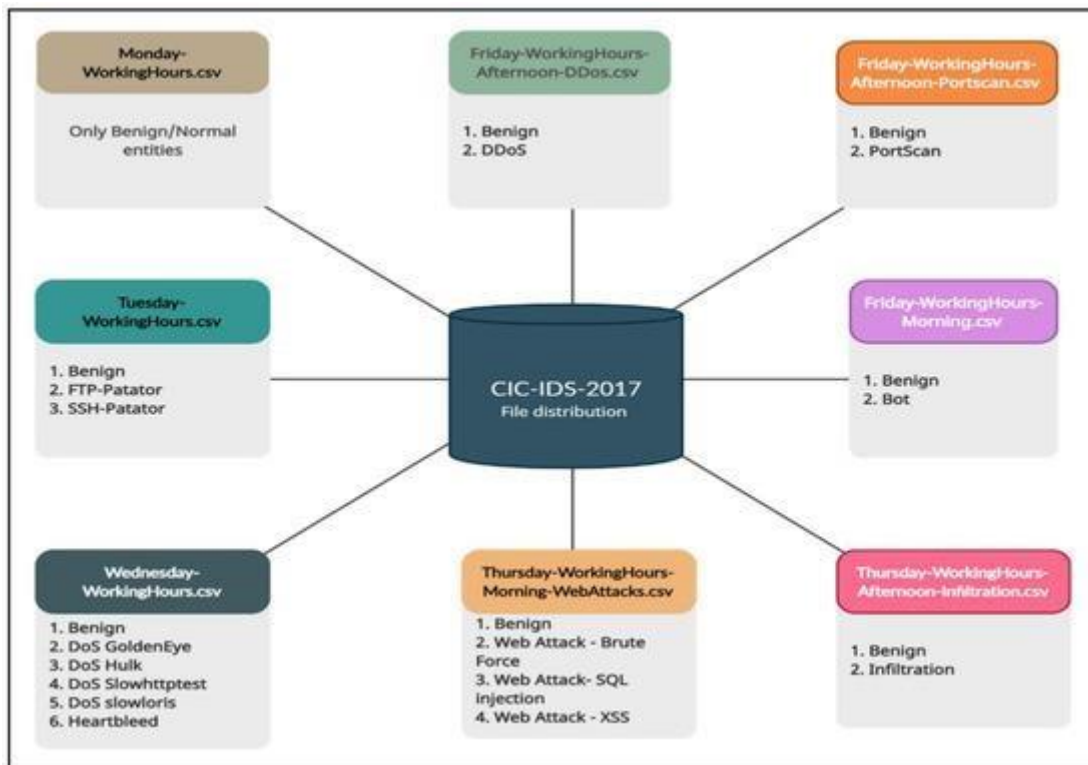
## OBJECTIVES OF THE PROJECT

The main objectives of this project are:

- To analyze real-world network traffic using machine learning
- To detect malicious activities in network traffic
- To classify traffic as benign or attack traffic
- To provide AI-based explanations for detection results
- To build an interactive web-based intrusion detection dashboard
- To deploy the system on a cloud platform for easy access

## DATASET DESCRIPTION

The **CIC-IDS2017 dataset**, developed by the Canadian Institute for Cybersecurity, is used in this project. This dataset contains realistic network traffic collected from a simulated enterprise environment.



### Key Features of the Dataset:

- Real-world traffic patterns
- Multiple attack types (DDoS, Port Scan, Infiltration, Web Attacks, etc.)
- Flow-level statistical features
- Labeled data for supervised learning

The dataset is preprocessed to remove missing values, infinite values, and irrelevant features. Only important features related to traffic behavior are selected for training the machine learning model.

Friday-WorkingHours-Afternoon-DDos.pcap\_ISCX.csv - Excel

Vishnu prya Vishnu prya

FileHomeInsertDrawPage LayoutFormulasDataReviewViewHelp

Tell me what you want to do

Share

Cut

Copy

Paste

Format Painter

Clipboard

Calibri

11

Wrap Text

General

Conditional Formatting

Format as Table

Cell Styles

Insert

Delete

Format

AutoSum

Fill

Clear

Sort & Filter

Find & Select

Add-ins

Add-ins

A1

Flow ID

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA
Flow ID	Source IP	Source Port	Destination IP	Destination Port	Protocol	Timestamp	Flow Duration	Total Fwd	Total Bwd	Total Len	Total Len	Fwd Packets	Fwd Packets	Fwd Packets	Bwd Packets	Bwd Packets	Bwd Packets	Bwd Packets	Bwd Packets	Bwd Packets	Flow Bytes	Flow Packets	Flow IAT	Flow IAT	Flow IAT	Flow IAT	Flow IAT
1	192.168.1.104.16.20	443	192.168.1.1	55055	6	#####	3	2	0	12	0	6	6	6	0	0	0	0	0	0	4000000	666666.7	3	0	3	3	3
2	192.168.1.104.16.28	80	192.168.1.1	55054	6	#####	109	1	1	6	6	6	6	6	0	6	6	6	0	0	110091.7	18348.62	109	0	109	109	109
3	192.168.1.104.16.28	80	192.168.1.1	55055	6	#####	52	1	1	6	6	6	6	6	0	6	6	6	0	0	230769.2	38461.54	52	0	52	52	52
4	192.168.1.104.17.24	443	192.168.1.1	46236	6	#####	34	1	1	6	6	6	6	6	0	6	6	6	0	0	352941.2	58823.53	34	0	34	34	34
5	192.168.1.104.18.19	443	192.168.1.1	54863	6	#####	3	2	0	12	0	6	6	6	0	0	0	0	0	0	4000000	666666.7	3	0	3	3	3
6	192.168.1.104.20.10	443	192.168.1.1	54871	6	#####	1022	2	0	12	0	6	6	6	0	0	0	0	0	0	11741.68	1956.947	1022	0	1022	1022	1022
7	192.168.1.104.20.10	443	192.168.1.1	54925	6	#####	4	2	0	12	0	6	6	6	0	0	0	0	0	0	3000000	500000	4	0	4	4	4
8	192.168.1.104.20.10	443	192.168.1.1	54925	6	#####	42	1	1	6	6	6	6	6	0	6	6	6	0	0	285714.3	47619.05	42	0	42	42	42
9	192.168.1.104.28.13	443	192.168.1.1	9282	6	#####	4	2	0	12	0	6	6	6	0	0	0	0	0	0	3000000	500000	4	0	4	4	4
10	192.168.1.104.97.12	443	192.168.1.1	55153	6	#####	4	2	0	37	0	31	6	18.5	17.67767	0	0	0	0	0	9250000	500000	4	0	4	4	4
11	192.168.1.104.97.12	443	192.168.1.1	55143	6	#####	3	2	0	37	0	31	6	18.5	17.67767	0	0	0	0	0	12300000	666666.7	3	0	3	3	3
12	192.168.1.104.97.12	443	192.168.1.1	55144	6	#####	1	2	0	37	0	31	6	18.5	17.67767	0	0	0	0	0	37000000	2000000	1	0	1	1	1
13	192.168.1.104.97.12	443	192.168.1.1	55145	6	#####	4	2	0	37	0	31	6	18.5	17.67767	0	0	0	0	0	9250000	500000	4	0	4	4	4
14	192.168.1.104.97.13	443	192.168.1.1	55254	6	#####	3	3	0	43	0	31	6	14.33333	14.43376	0	0	0	0	0	14300000	1000000	1.5	0.707107	2	1	1
15	192.168.1.104.97.14	80	192.168.1.1	36206	6	#####	54	1	1	0	0	0	0	0	0	0	0	0	0	0	37037.04	54	0	54	54	54	
16	192.168.1.121.29.54	443	192.168.1.1	53524	6	#####	1	2	0	0	0	0	0	0	0	0	0	0	0	0	2000000	1	0	1	1	1	
17	192.168.1.121.29.54	443	192.168.1.1	53524	6	#####	154	1	1	0	0	0	0	0	0	0	0	0	0	0	12987.01	154	0	154	154	154	
18	192.168.1.121.29.54	443	192.168.1.1	53526	6	#####	1	2	0	0	0	0	0	0	0	0	0	0	0	0	2000000	1	0	1	1	1	
19	192.168.1.121.29.54	443	192.168.1.1	53526	6	#####	118	1	1	0	0	0	0	0	0	0	0	0	0	0	16949.15	118	0	118	118	118	
20	192.168.1.121.29.54	443	192.168.1.1	53527	6	#####	239	1	1	0	0	0	0	0	0	0	0	0	0	0	8368.201	239	0	239	239	239	
21	192.168.1.121.29.54	443	192.168.1.1	53528	6	#####	1	3	0	0	0	0	0	0	0	0	0	0	0	0	3000000	0.5	0.707107	1	0	1	
22	192.168.1.121.29.54	443	192.168.1.1	53527	6	#####	1	2	0	0	0	0	0	0	0	0	0	0	0	0	2000000	1	0	1	1	1	
23	192.168.1.121.29.54	443	192.168.1.1	55035	6	#####	4	2	0	248	0	217	31	124	131.5219	0	0	0	0	0	62000000	500000	4	0	4	4	4
24	144.76.12.144.76.12	443	192.168.1.1	55275	6	#####	5	3	0	254	0	217	6	84.66667	115.2837	0	0	0	0	0	50800000	600000	2.5	2.12132	4	1	1
25	145.243.2.145.243.2	443	192.168.1.1	55277	6	#####	4	2	0	12	0	6	6	6	0	0	0	0	0	0	3000000	500000	4	0	4	4	4
26	151.101.0.151.101.0	443	192.168.1.1	8850	6	#####	4	3	0	43	0	31	6	14.33333	14.43376	0	0	0	0	0	108000000	750000	2	1.414214	3	1	1
27	151.101.0.151.101.0	80	192.168.1.1	43248	6	#####	54	1	1	0	0	0	0	0	0	0	0	0	0	0	37037.04	54	0	54	54	54	
28	151.101.0.151.101.0	443	192.168.1.1	8678	6	#####	42	3	0	43	0	31	6	14.33333	14.43376	0	0	0	0	0	1023810	71438.57	21	25.45584	39	3	3
29	151.101.1.151.101.1	443	192.168.1.1	55063	6	#####	4	2	0	37	0	31	6	18.5	17.67767	0	0	0	0	0	9250000	500000	4	0	4	4	4
30	151.101.1.151.101.1	443	192.168.1.1	55203	6	#####	3	2	0	37	0	31	6	18.5	17.67767	0	0	0	0	0	12300000	666666.7	3	0	3	3	3
31	151.101.1.151.101.1	443	192.168.1.1	55140	6	#####	3	2	0	37	0	31	6	18.5	17.67767	0	0	0	0	0	12300000	666666.7	3	0	3	3	3
32	151.101.1.151.101.1	443	192.168.1.1	55180	6	#####	737	2	1	37	6	31	6	18.5	17.67767	6	6	6	6	0	58344.64	4070.556	368.5	310.4199	588	149	149

Friday-WorkingHours-Afternoon-D

Ready

Accessibility: Unavailable

# METHODOLOGY

## 1 Data Preprocessing

- Removal of missing and infinite values
- Cleaning and formatting feature names
- Selection of important flow-level features

## 2 Feature Selection

Key features such as:

- Flow Duration
- Total Forward Packets
- Packet Length Statistics
- Flow Inter-Arrival Time
- Packet Rate

are used to train the model.

## 3 Model Training

A **Random Forest classifier** is trained using the preprocessed dataset. The data is split into training and testing sets to evaluate model performance.

```
File Edit Selection View Go Run Terminal Help ← → Search
app.py x
E:\Dharani> ALNID_project> app.py
1 import streamlit as st
2 import pandas as pd
3 import numpy as np
4 from sklearn.ensemble import RandomForestClassifier
5 from sklearn.model_selection import train_test_split
6 from sklearn.metrics import accuracy_score
7 from groq import Groq
8 import os
9
10 # --- PAGE SETUP ---
11 st.set_page_config(page_title="AI-NIDS Student Project", layout="wide")
12
13 st.title("AI-Based Network Intrusion Detection System")
14 st.markdown("""
15 **Student Project**: This system uses **Random Forest** to detect Network attacks and **Groq AI** to explain the packets.
16 """)
17
18 # --- CONFIGURATION ---
19 DATA_FILE = "Friday-Workinghours-Afternoon-DDos.pcap_ISCX.csv"
20
21 # --- SIDEBAR: SETTINGS ---
22 st.sidebar.header("1. Settings")
23 groq_api_key = st.sidebar.text_input("Groq API Key (starts with gsk_)", type="password")
24 st.sidebar.caption("[get a free key here](https://console.groq.com/keys)")
25
26 st.sidebar.header("2. Model Training")
27
28 @st.cache_data
29 def load_data(filepath):
30     try:
31         df = pd.read_csv(filepath, nrows=15000)
32         df.columns = df.columns.str.strip()
33         df.replace([np.inf, -np.inf], np.nan, inplace=True)
34         df.dropna(inplace=True)
35         return df
36     except FileNotFoundError:
37         return None
38
39 Amazon Q Activating Extensions... Ln 1, Col 1 Spaces 4 UTF-8 LF Python Signed out 3.14.0 Go Live Prettier
21°C Sunny
```

```
File Edit Selection View Go Run Terminal Help ← → Search
app.py 1 x
E:\Dharani> ALNID_project> app.py
39 def train_model(df):
40     features = ['Flow Duration', 'Total Fwd Packets', 'Total Backward Packets',
41               'Total Length of Fwd Packets', 'Fwd Packet Length Max',
42               'Flow IAT Mean', 'Flow IAT Std', 'Flow Packets/s']
43     target = 'Label'
44
45     missing_cols = [c for c in features if c not in df.columns]
46     if missing_cols:
47         st.error(f"Missing columns in CSV: {missing_cols}")
48         return None, 0, [], None, None
49
50     X = df[features]
51     y = df[target]
52
53     X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
54
55     clf = RandomForestClassifier(n_estimators=10, max_depth=10, random_state=42)
56     clf.fit(X_train, y_train)
57
58     score = accuracy_score(y_test, clf.predict(X_test))
59     return clf, score, features, X_test, y_test
60
61 # --- APP LOGIC ---
62 df = load_data(DATA_FILE)
63
64 if df is None:
65     st.error(f"Error: File '{DATA_FILE}' not found. Please upload it to the Files tab.")
66     st.stop()
67
68 st.sidebar.success(f"Dataset loaded: {len(df)} rows")
69
70 if st.sidebar.button("Train Model Now"):
71     with st.spinner("Training model..."):
72         clf, accuracy, feature_names, X_test, y_test = train_model(df)
73         if clf:
74             st.session_state['model'] = clf
75             st.session_state['feature_names'] = feature_names
76
77 Nifty midcap -0.69% Amazon Q Ln 1, Col 1 Spaces 4 UTF-8 LF Python Signed out 3.14.0 Go Live Prettier
10:51 05-02-2026
```

```
75 st.session_state['features'] = feature_names
76 st.session_state['X_test'] = X_test
77 st.session_state['y_test'] = y_test
78 st.sidebar.success(f"Training Complete! Accuracy: {accuracy:.2%}")
79
80 st.header("Threat Analysis Dashboard")
81
82 if 'model' in st.session_state:
83     col1, col2 = st.columns(2)
84
85     with col1:
86         st.subheader("Simulation")
87         st.info("Pick a random packet from the test data to simulate live traffic.")
88
89         if st.button("Capture Random Packet"):
90             random_idx = np.random.randint(0, len(st.session_state['X_test']))
91             packet_data = st.session_state['X_test'].iloc[random_idx]
92             actual_label = st.session_state['y_test'].iloc[random_idx]
93
94             st.session_state['current_packet'] = packet_data
95             st.session_state['actual_label'] = actual_label
96
97     if 'current_packet' in st.session_state:
98         packet = st.session_state['current_packet']
99
100     with col1:
101         st.write("***Packet Header Info***")
102         st.dataframe(packet, use_container_width=True)
103
104     with col2:
105         st.subheader("AI Detection Result")
106         prediction = st.session_state['model'].predict([packet])[0]
107
108         if prediction == "BENIGN":
109             st.success(f"STATUS: **SAFE (BENIGN)**")
110         else:
111             st.error(f"STATUS: **ATTACK DETECTED ((prediction))**")
112
113     st.caption(f"Ground Truth Label: {st.session_state['actual_label']}")
114
115     st.markdown("----")
116     st.subheader("Ask AI Analyst (Groq)")
117
118     if st.button("Generate Explanation"):
119         if not groq_api_key:
120             st.warning("Please enter your Groq API Key in the sidebar first.")
121         else:
122             try:
123                 client = Groq(api_key=groq_api_key)
124
125                 prompt = f"""
126                 You are a cybersecurity analyst.
127                 A network packet was detected as: {prediction}.
128
129                 Packet Technical Details:
130                 {packet.to_string()}
131
132                 Please explain:
133                 1. Why these specific values (like flow Duration or Packet Length) might indicate {prediction}.
134                 2. If it is BENIGN, explain why it looks normal.
135                 3. Keep the answer short and simple for a student.
136                 """
137
138                 with st.spinner("Groq is analyzing the packet..."):
139                     completion = client.chat.completions.create(
140                         model="llama-3.3-70b-versatile", # <--- UPDATED MODEL NAME
141                         messages=[
142                             {"role": "user", "content": prompt}
143                         ],
144                         temperature=0.6,
145                     )
146                     st.info(completion.choices[0].message.content)
```

## 4 Model Evaluation

The trained model is evaluated using accuracy as the primary metric. The model demonstrates effective performance in distinguishing between benign and malicious traffic.

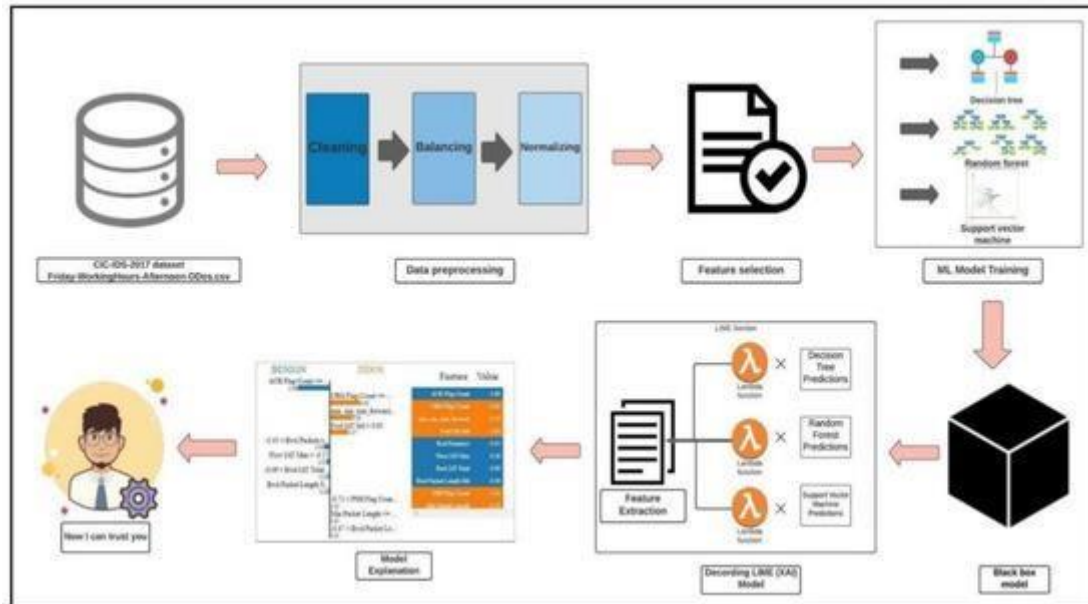
# SYSTEM ARCHITECTURE

The system follows a modular architecture consisting of:

1. Data Input Module
2. Data Preprocessing Module

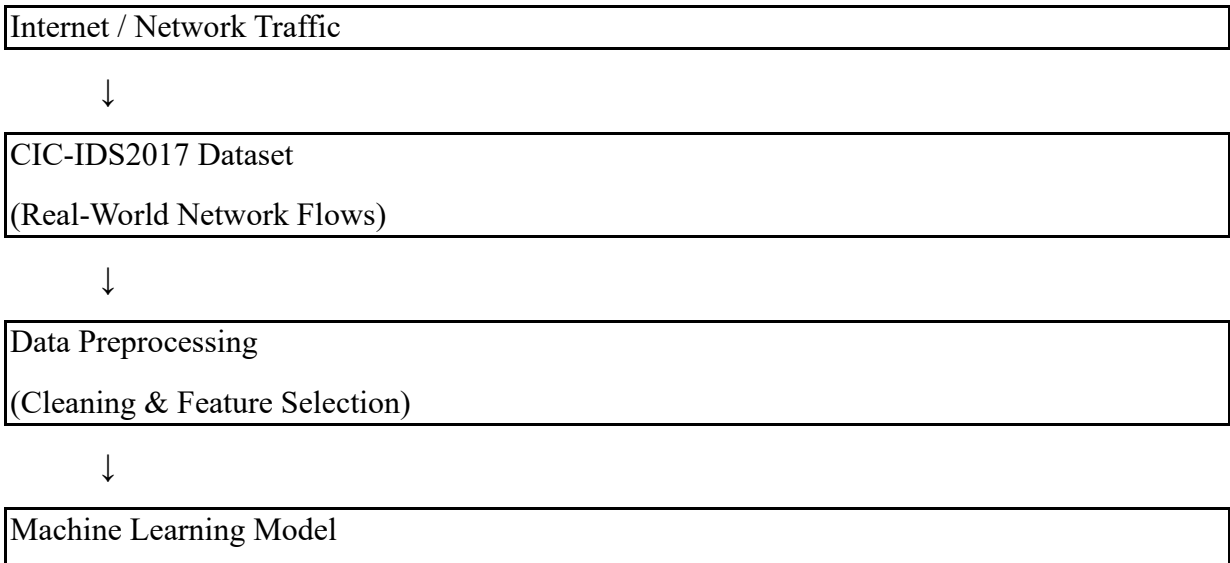


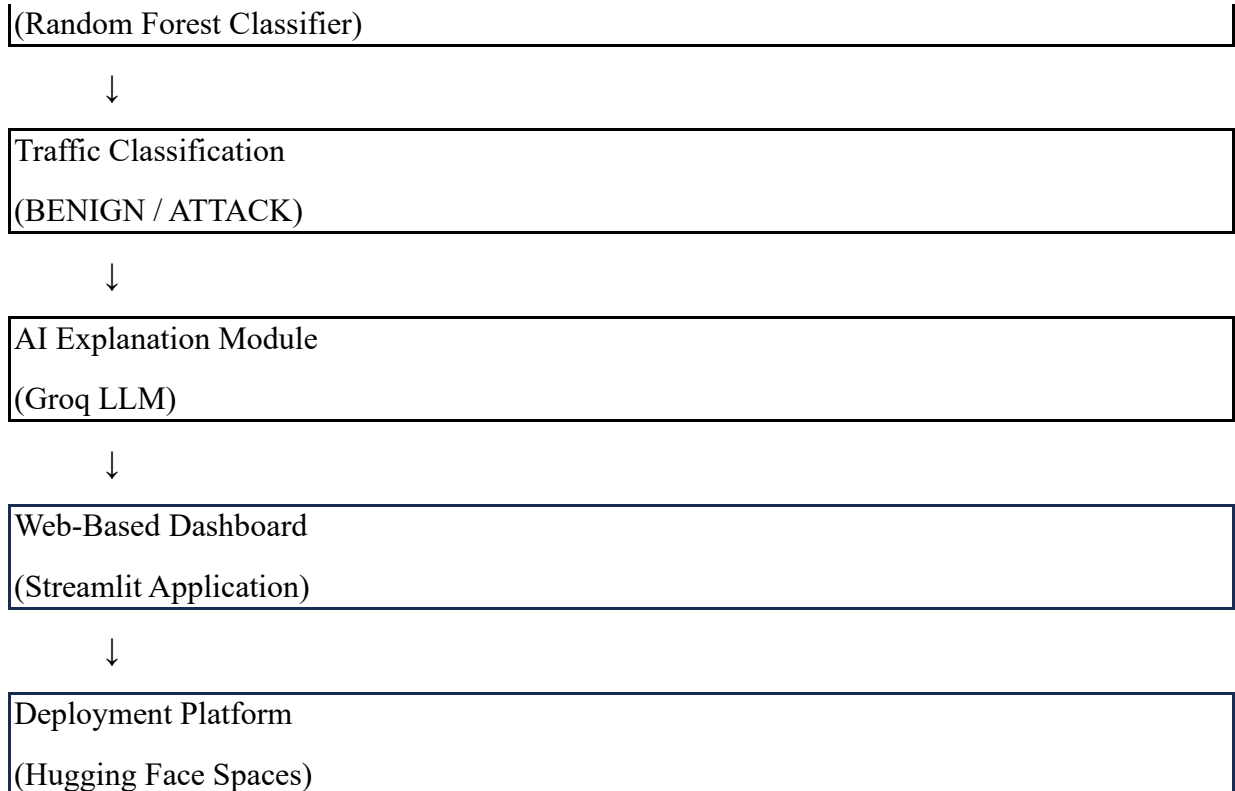
3. Machine Learning Model
4. Intrusion Detection Module
5. AI Explanation Module
6. Web Dashboard Interface
7. Deployment Platform



Each module works independently and communicates with other components to provide a complete intrusion detection solution.

## Flowchart

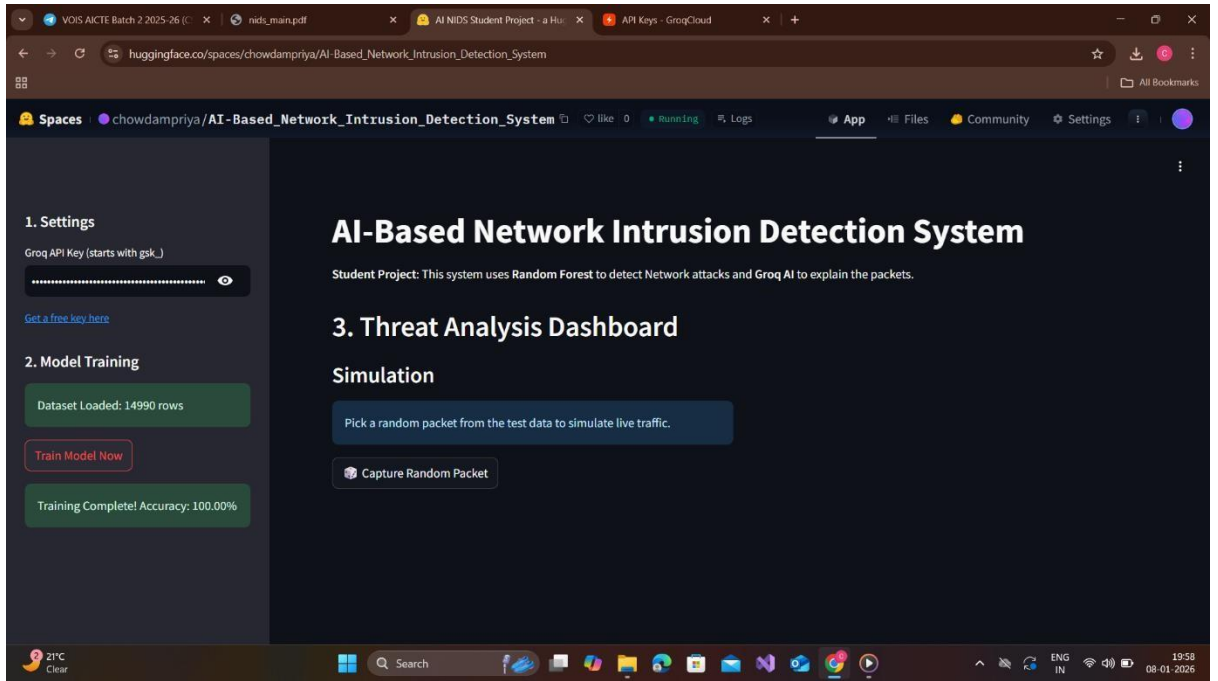


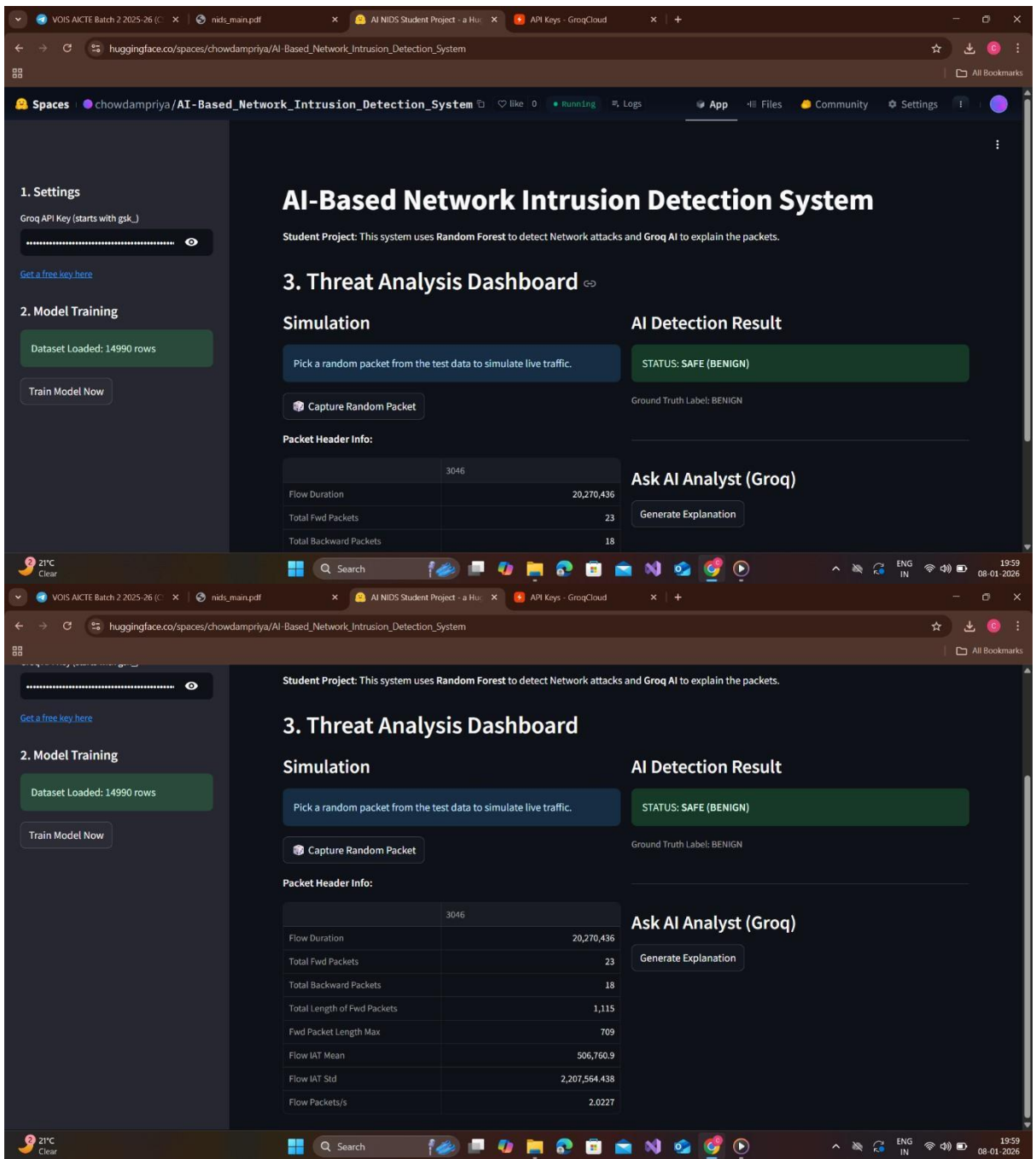


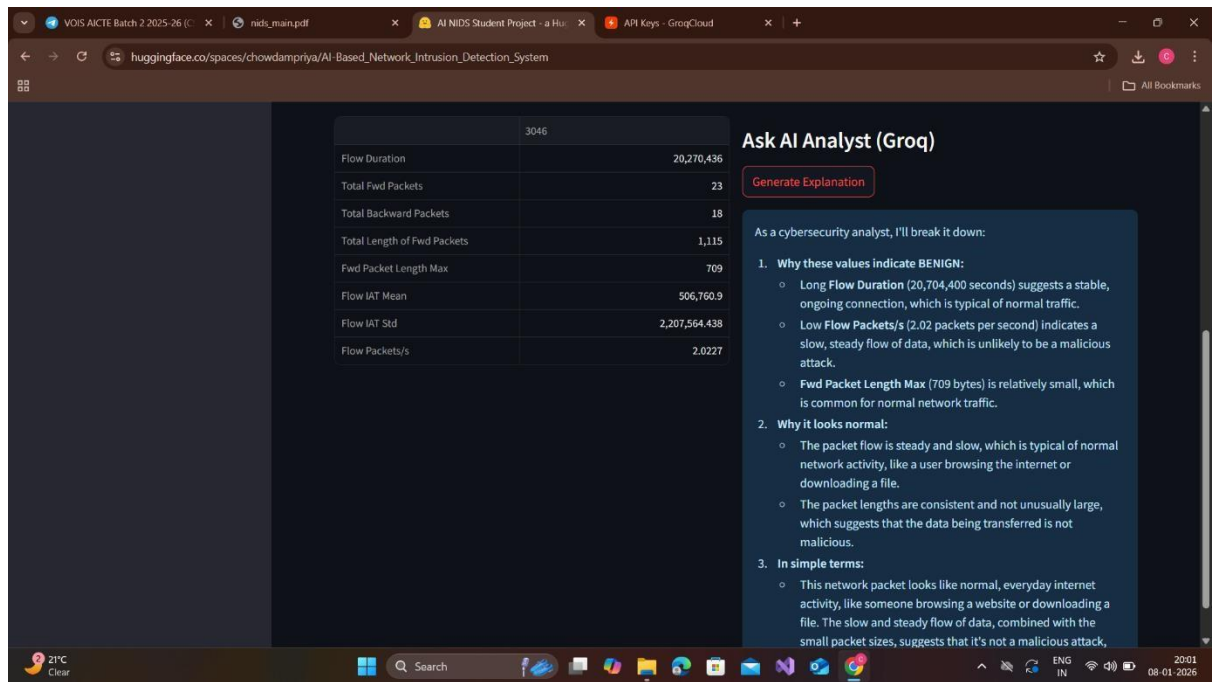
## RESULTS AND OUTCOMES

- Successful detection of network intrusions
- High classification accuracy using Random Forest
- Clear AI-based explanations for detection results
- Interactive and user-friendly web dashboard
- Successful cloud deployment on Hugging Face Spaces









## APPLICATIONS

- Network security monitoring
- Cyberattack detection and prevention
- Academic research and learning
- Enterprise network protection
- Security operations centers (SOC)

## FUTURE ENHANCEMENTS

- Real-time packet capture using Scapy
- Deep learning-based intrusion detection
- Multi-class attack classification
- Automated alert and reporting system
- Integration with enterprise security tools

## CONCLUSION

This project demonstrates the effective use of machine learning and explainable AI in network security. The AI-Based Network Intrusion Detection System provides accurate detection of cyberattacks while offering transparency through AI explanations. The system is

scalable, interactive, and suitable for real-world security applications as well as academic research.