



# 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.

The screenshot shows a Microsoft Excel spreadsheet with the following details:

- File Name:** Friday-WorkingHours-Afternoon-DDos.pcap.JSX.csv
- Columns:** The first 33 columns represent network flow features, starting with Flow ID and ending with Flow IAT Jitter.
- Data:** The data consists of approximately 1000 rows of network traffic data, with each row containing values for the 33 columns.
- Toolbars and Menus:** Standard Excel toolbars for Home, Insert, Draw, Page Layout, Formulas, Data, Review, View, Help, and Tell me what you want to do.
- Status Bar:** Shows system information like battery level (21°C, Sunny), search bar, taskbar icons, language (ENG IN), signal strength, and date/time (05-02-2026).

# 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.

The screenshot shows a Jupyter Notebook interface with a Python script named `app.py`. The code is as follows:

```
E:\Dharani>ALNIDS_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 # --- APP LOGIC ---
40 df = load_data(DATA_FILE)
41
42 if df is None:
43     st.error(f"Error: File '{DATA_FILE}' not found. Please upload it to the Files tab.")
44     st.stop()
45
46 st.sidebar.success(f"Dataset Loaded: {len(df)} rows")
47
48 if st.sidebar.button("Train Model Now"):
49     with st.spinner("Training model..."):
50         clf, accuracy, feature_names, X_test, y_test = train_model(df)
51         if clf:
52             st.session_state['model'] = clf
53             st.session_state['feature_names'] = feature_names
54
55 st.sidebar.button("Explain Model")
56
57 st.sidebar.button("Deploy Model")
58
59 st.sidebar.button("Logout")
```

The screenshot shows a Jupyter Notebook interface with a Python script named `app.py`. The code is as follows:

```
E:\Dharani>ALNIDS_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, None, 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 st.sidebar.button("Explain Model")
78
79 st.sidebar.button("Deploy Model")
80
81 st.sidebar.button("Logout")
```

The image shows a dual-monitor setup. The left monitor displays a Python code editor with a file named 'app.py'. The code implements a threat analysis dashboard using Streamlit. It includes logic for capturing random packets from test data, performing AI detection, and generating explanations using a GPT model like Groq. The right monitor also displays a Python code editor, showing a different part of the application or perhaps a related project.

```
File Edit Selection View Go Run Terminal Help ← → Q Search E > Dharni > AI_NID_project > app.py ... 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) 147 148 Nifty midcap -0.69% 0 Amazon Q File Edit Selection View Go Run Terminal Help ← → Q Search E > Dharni > AI_NID_project > app.py ... 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) 147 148 Nifty midcap -0.69% 0 Amazon Q File Edit Selection View Go Run Terminal Help ← → Q Search E > Dharni > AI_NID_project > app.py ... 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) 147 148 Nifty midcap -0.69% 0 Amazon Q
```

## 4 Model Evaluation

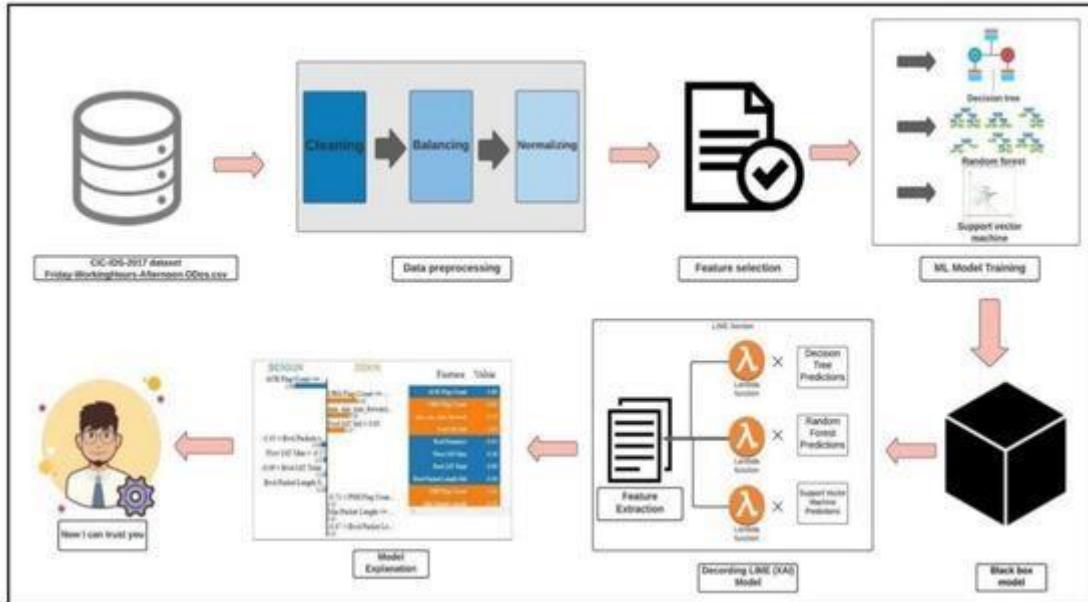
The trained model is evaluated using accuracy as the primary metric. The model demonstrates effective performance in distinguishing between benign and malicious traffic.

## SYSTEM ARCHITECTURE

The system follows a modular architecture consisting of:

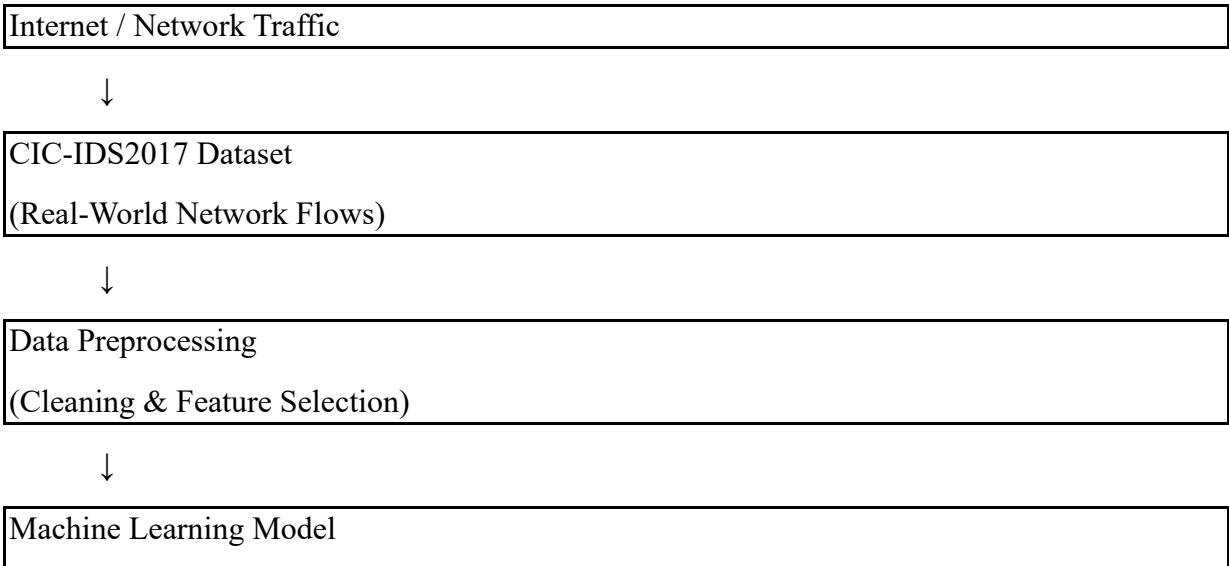
1. Data Input Module
  2. Data Preprocessing Module

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



(Random Forest Classifier)



Traffic Classification

(BENIGN / ATTACK)



AI Explanation Module

(Groq LLM)



Web-Based Dashboard

(Streamlit Application)

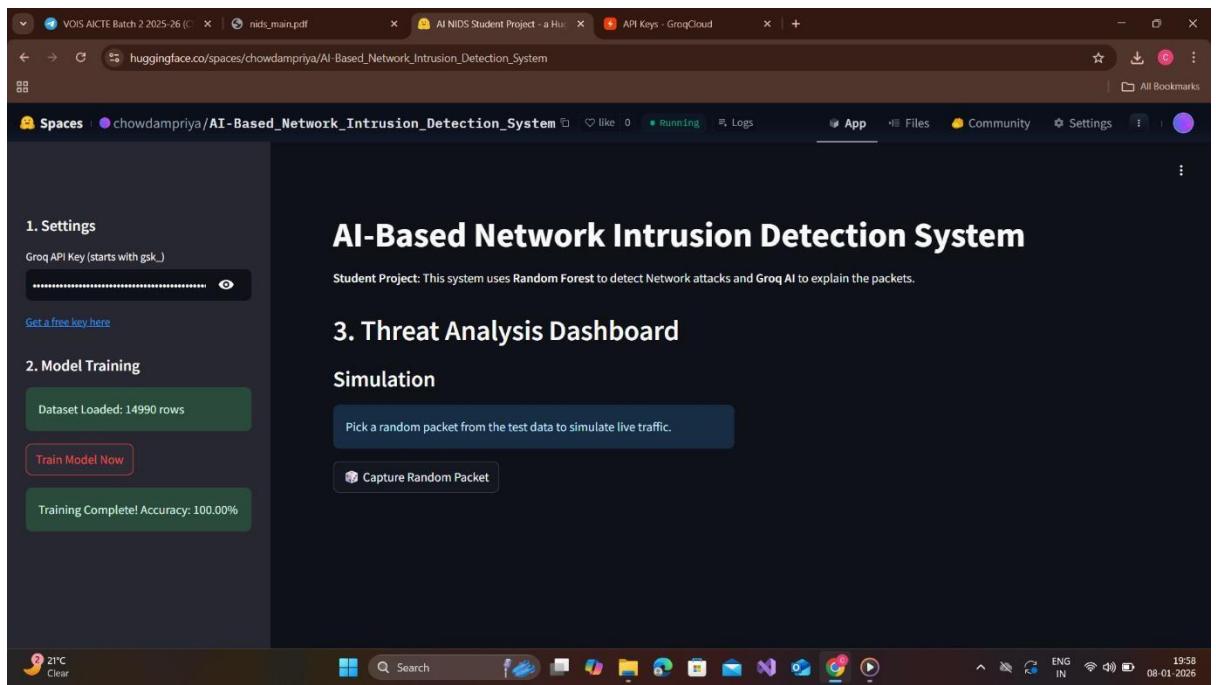


Deployment Platform

(Hugging Face Spaces)

## 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



**AI-Based Network Intrusion Detection System**

Student Project: This system uses Random Forest to detect Network attacks and Groq AI to explain the packets.

**3. Threat Analysis Dashboard**

**Simulation**

Pick a random packet from the test data to simulate live traffic.

**AI Detection Result**

STATUS: SAFE (BENIGN)

Ground Truth Label: BENIGN

**Ask AI Analyst (Groq)**

Generate Explanation

**Packet Header Info:**

	3046
Flow Duration	20,270,436
Total Fwd Packets	23
Total Backward Packets	18
Total Length of Fwd Packets	1,115
Fwd Packet Length Max	709
Flow IAT Mean	506,760.9
Flow IAT Std	2,207,564.438
Flow Packets/s	2.0227

The screenshot shows a web browser window with several tabs open. The active tab displays network flow statistics in a table:

	3046
Flow Duration	20,270,436
Total Fwd Packets	23
Total Backward Packets	18
Total Length of Fwd Packets	1,115
Fwd Packet Length Max	709
Flow IAT Mean	506,760.9
Flow IAT Std	2,207,564.438
Flow Packets/s	2.0227

To the right of the table is a sidebar titled "Ask AI Analyst (Groq)" with a "Generate Explanation" button. The sidebar contains text and a numbered list:

As a cybersecurity analyst, I'll break it down:

- Why these values indicate BEIGN:**
  - Long Flow Duration (20,704,400 seconds) suggests a stable, ongoing connection, which is typical of normal traffic.
  - Low Flow Packets/s (2.02 packets per second) indicates a slow, steady flow of data, which is unlikely to be a malicious attack.
  - Fwd Packet Length Max (709 bytes) is relatively small, which is common for normal network traffic.
- Why it looks normal:**
  - The packet flow is steady and slow, which is typical of normal network activity, like a user browsing the internet or downloading a file.
  - The packet lengths are consistent and not unusually large, which suggests that the data being transferred is not malicious.
- In simple terms:**
  - This network packet looks like normal, everyday internet activity, like someone browsing a website or downloading a file. The slow and steady flow of data, combined with the small packet sizes, suggests that it's not a malicious attack,

The browser interface includes a toolbar at the top and a taskbar at the bottom showing various icons and system status.

## 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.