**5G NR DDoS Detection Dataset Generator**

A comprehensive NS-3 simulation framework for generating machine learning datasets to detect DDoS attacks in 5G networks. This project generates structured CSV datasets with network flow features suitable for training ML models to distinguish between benign and malicious traffic.

**📋 Table of Contents**

* [Overview](https://markdownlivepreview.com/#overview)
* [Features](https://markdownlivepreview.com/#features)
* [Project Structure](https://markdownlivepreview.com/#project-structure)
* [Dataset Specification](https://markdownlivepreview.com/#dataset-specification)
* [Installation & Setup](https://markdownlivepreview.com/#installation--setup)
* [Usage](https://markdownlivepreview.com/#usage)
* [Dataset Generation](https://markdownlivepreview.com/#dataset-generation)
* [Output Format](https://markdownlivepreview.com/#output-format)
* [ML-Ready Features](https://markdownlivepreview.com/#ml-ready-features)
* [Troubleshooting](https://markdownlivepreview.com/#troubleshooting)
* [Contributing](https://markdownlivepreview.com/#contributing)

**🔍 Overview**

This project implements a parameterized NS-3 simulation framework that generates labeled datasets for machine learning-based DDoS detection in 5G networks. The simulator creates realistic network scenarios with varying node counts, mobility patterns, transport protocols, and attack intensities to produce comprehensive training data.

**Key Capabilities**

* **Multi-scenario simulation**: 10/15 UEs, static/mobile, UDP/TCP, benign/DDoS
* **Attack intensity variation**: Low, medium, high DDoS attack rates
* **ML-ready CSV export**: Per-second flow statistics with labels
* **Batch processing**: 70 and 20 repetitions across all parameter combinations
* **Realistic 5G modeling**: NR module integration with mobility and channel models

**✨ Features**

**Simulation Features**

* **Node Counts**: 10, 15, 25, 50 UEs
* **Mobility Models**: Static positioning, RandomWalk2d mobility
* **Transport Protocols**: UDP, TCP with configurable parameters
* **Attack Scenarios**: Benign traffic, DDoS attacks with varying intensities
* **Network Topology**: 5G NR with gNB, UEs, and realistic channel models

**Dataset Features**

* **Time Windows**: 1-second sampling intervals
* **Flow Statistics**: Per-flow throughput, delay, jitter, packet loss
* **Traffic Classification**: Binary and intensity-based labels
* **ML-Ready Format**: CSV with standardized column names
* **Comprehensive Coverage**: 2,640+ scenarios across all parameter combinations

**📁 Project Structure**

ns3-5g-simulation/

├── README.md # This file

├── nr\_ddos\_dataset.cc # Original NR-based dataset generator

├── simple\_ddos\_dataset.cc # Simplified dataset generator (recommended)

├── run\_scenarios.ps1 # PowerShell batch runner

├── run\_win\_datasets.ps1 # Windows-specific batch runner

├── 1.simple25.cc # Basic 25-node scenario

├── 2.simple50.cc # Basic 50-node scenario

├── 3.simple25-attack.cc # 25-node with DDoS attacks

├── 4.simple50-attack.cc # 50-node with DDoS attacks

├── 5.simple25-mobile.cc # 25-node with mobility

├── 6.simple50-mobile.cc # 50-node with mobility

├── 7.simple25-mobile-ddos.cc # 25-node mobile with DDoS

├── 8.simple50-mobile-ddos.cc # 50-node mobile with DDoS

├── Results.xlsx # Summary results spreadsheet

└── datasets/ # Generated CSV datasets

├── dataset\_10\_mobile\_udp\_ddos\_high\_r1.csv

├── dataset\_15\_static\_tcp\_benign\_low\_r70.csv

└── ...

**📊 Dataset Specification**

**Purpose**

Generate ML-ready labeled datasets for training and testing intrusion detection models in 5G NR environments. Labels distinguish between benign vs attack traffic with intensity levels (low/medium/high).

**Dataset Structure**

Each row represents one time window of traffic (1 second). The schema includes:

| **Column** | **Description** | **Type** | **Example** |
| --- | --- | --- | --- |
| time\_start | Window start time (s) | float | 0.0 |
| time\_end | Window end time (s) | float | 1.0 |
| scenario\_id | Unique scenario identifier | string | "10\_mobile\_udp\_ddos\_high\_r1" |
| ue\_total | Total number of UEs | int | 10 |
| attackers | Number of attacker nodes | int | 5 |
| src\_ip | Source IP address | string | "10.1.1.1" |
| dst\_ip | Destination IP address | string | "10.1.1.2" |
| src\_port | Source port number | int | 49153 |
| dst\_port | Destination port number | int | 1001 |
| protocol | Transport protocol (6=TCP, 17=UDP) | int | 6 |
| packet\_size | Average packet size (bytes) | float | 1024.0 |
| flow\_duration | Flow duration (s) | float | 1.0 |
| total\_bytes\_fwd | Forward bytes in window | int | 1024 |
| total\_bytes\_bwd | Backward bytes in window | int | 0 |
| total\_pkts\_fwd | Forward packets in window | int | 1 |
| total\_pkts\_bwd | Backward packets in window | int | 0 |
| pkts\_per\_sec | Total packets per second | float | 1.0 |
| bytes\_per\_sec | Total bytes per second | float | 1024.0 |
| flow\_pkts\_per\_sec | Flow packets per second | float | 1.0 |
| flow\_bytes\_per\_sec | Flow bytes per second | float | 1024.0 |
| jitter\_ms | Average jitter (ms) | float | 0.0 |
| delay\_ms | Average delay (ms) | float | 5.2 |
| label\_binary | Binary attack label (0=benign, 1=attack) | int | 0 |
| label\_intensity | Attack intensity (0=benign, 1=low, 2=med, 3=high) | int | 0 |

**Dataset Variations**

The generator creates datasets for multiple conditions:

* **UE Counts**: 10, 15, 25, 50 nodes
* **Mobility**: Static vs Mobile (RandomWalk2d)
* **Transport**: UDP vs TCP
* **Attack Types**: Benign vs DDoS
* **Intensity Levels**: Low (0.0008s), Medium (0.0004s), High (0.00015s) intervals
* **Repetitions**: 70 runs per scenario combination for small scale network of 10, 15 nodes and 20 runs per scenario combination for medium to large scale network of 25, 50 nodes

**🛠 Installation & Setup**

**Prerequisites**

* **NS-3**: Version 3.36+ with NR module
* **WSL**: Ubuntu 20.04+ (recommended)
* **Build Tools**: CMake, GCC, Python3
* **Dependencies**: Qt5, Boost, SQLite3

**WSL Setup (Recommended)**

1.**Install Ubuntu WSL**:

wsl --install -d Ubuntu

2.**Install NS-3 dependencies**:

sudo apt update

sudo apt install -y build-essential g++ python3 python3-pip git cmake \

qtbase5-dev libxml2-dev libsqlite3-dev libboost-all-dev

3.**Clone and setup NS-3**:

cd ~

git clone --depth 1 https://gitlab.com/nsnam/ns-3-dev.git ~/ns-3-dev

cd ~/ns-3-dev

4.**Copy dataset generator**:

# Copy the simplified generator to NS-3 scratch directory

cp "/mnt/e/ns3s-5g/ns3 5g simulation/ns3 5g simulation/simple\_ddos\_dataset.cc" scratch/

5.**Build NS-3**:

cmake -S . -B build -DNS3\_EXAMPLES=ON -DNS3\_TESTS=OFF

cmake --build build -j2

**Windows Setup (Alternative)**

1. **Install NS-3** on Windows
2. **Copy files** to NS-3 scratch directory
3. **Build** using waf or CMake
4. **Run** using PowerShell scripts

**🚀 Usage**

**Single Simulation**

# Basic usage

./build/scratch/ns3-dev-simple\_ddos\_dataset-default \

--ueTotal=10 --mobile=1 --useTcp=0 --enableAttack=1 \

--attackInterval=0.0004 --simTime=20 \

--outPath=./datasets --scenarioTag=test --window=1

**Parameter Options**

| **Parameter** | **Description** | **Default** | **Options** |
| --- | --- | --- | --- |
| --ueTotal | Number of UEs | 25 | 10, 15, 25, 50 |
| --mobile | Enable mobility | true | 0 (static), 1 (mobile) |
| --useTcp | Use TCP instead of UDP | false | 0 (UDP), 1 (TCP) |
| --enableAttack | Enable DDoS attacks | true | 0 (benign), 1 (DDoS) |
| --attackInterval | Attack packet interval | 0.0002s | 0.00015s (high), 0.0004s (med), 0.0008s (low) |
| --benignInterval | Benign packet interval | 0.02s | 0.01s - 0.1s |
| --simTime | Simulation duration | 20s | 5s - 300s |
| --outPath | Output directory | "datasets" | Any valid path |
| --scenarioTag | Scenario identifier | "default" | Custom string |
| --window | Sampling window size | 1s | 0.1s - 10s |

**📈 Dataset Generation**

**Small Scale Network Batch Processing**

The project includes automated batch processing to generate comprehensive datasets:

# Full 70-repetition batch (1,680 total scenarios)

reps=70

cd ~/ns-3-dev

for r in $(seq 1 $reps); do

for ue in 10 15; do

for mode in mobile static; do

for tp in udp tcp; do

for atk in ddos benign; do

for name in low med high; do

case "$name" in

low) interval=0.0008 ;;

med) interval=0.0004 ;;

high) interval=0.00015 ;;

esac

mobile=$([ "$mode" = "mobile" ] && echo 1 || echo 0)

useTcp=$([ "$tp" = "tcp" ] && echo 1 || echo 0)

enableAttack=$([ "$atk" = "ddos" ] && echo 1 || echo 0)

tag="${ue}\_${mode}\_${tp}\_${atk}\_${name}\_r${r}"

echo "Running: $tag"

./build/scratch/ns3-dev-simple\_ddos\_dataset-default \

--ueTotal=$ue --mobile=$mobile --useTcp=$useTcp --enableAttack=$enableAttack \

--attackInterval=$interval --simTime=20 \

--outPath=/root/ns3-datasets --scenarioTag=$tag --window=1

done

done

done

done

done

done

**Scenario Combinations**

The batch generates datasets for all combinations:

* **UE Counts**: 10, 15 (2 options)
* **Mobility**: mobile, static (2 options)
* **Transport**: udp, tcp (2 options)
* **Attack**: ddos, benign (2 options)
* **Intensity**: low, med, high (3 options)
* **Repetitions**: 70 runs

**Total Scenarios**: 2 × 2 × 2 × 2 × 3 × 70 = **1,680 datasets**

**Large Scale Network Batch Processing**

# Full 20-repetition batch (960 total scenarios)

reps=20

cd ~/ns-3-dev

for r in $(seq 1 $reps); do

for ue in 25 50; do

for mode in mobile static; do

for tp in udp tcp; do

for atk in ddos benign; do

for name in low med high; do

case "$name" in

low) interval=0.0008 ;;

med) interval=0.0004 ;;

high) interval=0.00015 ;;

esac

mobile=$([ "$mode" = "mobile" ] && echo 1 || echo 0)

useTcp=$([ "$tp" = "tcp" ] && echo 1 || echo 0)

enableAttack=$([ "$atk" = "ddos" ] && echo 1 || echo 0)

tag="${ue}\_${mode}\_${tp}\_${atk}\_${name}\_r${r}"

echo "Running: $tag"

./build/scratch/ns3-dev-simple\_ddos\_dataset-default \

--ueTotal=$ue --mobile=$mobile --useTcp=$useTcp --enableAttack=$enableAttack \

--attackInterval=$interval --simTime=20 \

--outPath=/root/ns3-datasets --scenarioTag=$tag --window=1

done

done

done

done

done

done

**Scenario Combinations**

The batch generates datasets for all combinations:

* **UE Counts**: 25, 50 (2 options)
* **Mobility**: mobile, static (2 options)
* **Transport**: udp, tcp (2 options)
* **Attack**: ddos, benign (2 options)
* **Intensity**: low, med, high (3 options)
* **Repetitions**: 70 runs

**Total Scenarios**: 2 × 2 × 2 × 2 × 3 × 20 = **960 datasets**

**PowerShell Batch Runner (Alternate Windows)**

# Run batch with custom parameters

.\run\_win\_datasets.ps1 -NsPath "C:\path\to\ns3" -Repetitions 70

# Copy files only (no execution)

.\run\_win\_datasets.ps1 -NsPath "C:\path\to\ns3" -CopyOnly

**📄 Output Format**

**File Naming Convention**

dataset\_<scenario>\_ue<count>\_<tcp|udp>\_<ddos|benign>.csv

**Examples**:

* dataset\_10\_mobile\_udp\_ddos\_high\_r1\_ue10\_udp\_ddos.csv
* dataset\_15\_static\_tcp\_benign\_low\_r70\_ue15\_tcp\_benign.csv

**Directory Structure**

datasets/

├── dataset\_10\_mobile\_udp\_ddos\_high\_r1.csv

├── dataset\_10\_mobile\_udp\_ddos\_high\_r2.csv

├── dataset\_10\_mobile\_udp\_benign\_low\_r1.csv

├── dataset\_15\_static\_tcp\_ddos\_med\_r70.csv

└── ... (1,680 total files)

**CSV Format**

time\_start,time\_end,scenario\_id,ue\_total,attackers,src\_ip,dst\_ip,src\_port,dst\_port,protocol,packet\_size,flow\_duration,total\_bytes\_fwd,total\_bytes\_bwd,total\_pkts\_fwd,total\_pkts\_bwd,pkts\_per\_sec,bytes\_per\_sec,flow\_pkts\_per\_sec,flow\_bytes\_per\_sec,jitter\_ms,delay\_ms,label\_binary,label\_intensity

0,1,10\_mobile\_udp\_ddos\_high\_r1,10,5,10.1.1.1,10.1.1.2,49153,1001,17,1024,1,1024,0,1,0,1,1024,1,1024,0,5.2,1,3

1,2,10\_mobile\_udp\_ddos\_high\_r1,10,5,10.1.1.3,10.1.1.2,49154,1001,17,1024,1,1024,0,1,0,1,1024,1,1024,0,5.2,1,3

**🤖 ML-Ready Features**

**Feature Categories**

The generated datasets include all features specified in the ML requirements:

**Network Flow Features (Green-highlighted)**

* **Source IP Address**: IPv4 source address
* **Destination IP Address**: IPv4 destination address
* **Source Port**: Source port number
* **Destination Port**: Destination port number
* **Protocol Type**: Transport protocol (6=TCP, 17=UDP)
* **Packet Size**: Average packet size in bytes

**Traffic Statistics**

* **Flow Duration**: Time duration of the flow
* **Total Bytes Forward**: Bytes sent from source to destination
* **Total Bytes Backward**: Bytes sent from destination to source
* **Total Packets Forward**: Packets sent from source to destination
* **Total Packets Backward**: Packets sent from destination to source

**Rate Metrics**

* **Packets per Second**: Rate of packet transmission
* **Bytes per Second**: Data transmission rate
* **Flow Packets per Second**: Rate of packets within specific flow
* **Flow Bytes per Second**: Rate of bytes within specific flow

**Quality Metrics**

* **Jitter (ms)**: Packet delay variation
* **Delay (ms)**: End-to-end packet delay

**Labels**

* **Binary Label**: 0 (benign), 1 (attack)
* **Intensity Label**: 0 (benign), 1 (low), 2 (medium), 3 (high)

**ML Model Compatibility**

The datasets are compatible with:

* **Scikit-learn**: Pandas DataFrame import
* **PyTorch**: TensorDataset creation
* **TensorFlow**: CSV dataset loading
* **XGBoost**: Direct CSV import
* **Other ML frameworks**: Standard CSV format

**Sample ML Usage**

import pandas as pd

from sklearn.model\_selection import train\_test\_split

from sklearn.ensemble import RandomForestClassifier

# Load dataset

df = pd.read\_csv('dataset\_10\_mobile\_udp\_ddos\_high\_r1.csv')

# Prepare features and labels

features = df[['src\_ip', 'dst\_ip', 'src\_port', 'dst\_port', 'protocol',

'packet\_size', 'flow\_duration', 'total\_bytes\_fwd',

'total\_pkts\_fwd', 'pkts\_per\_sec', 'bytes\_per\_sec',

'jitter\_ms', 'delay\_ms']]

labels = df['label\_binary']

# Train-test split

X\_train, X\_test, y\_train, y\_test = train\_test\_split(features, labels, test\_size=0.2)

# Train model

model = RandomForestClassifier()

model.fit(X\_train, y\_train)

# Evaluate

accuracy = model.score(X\_test, y\_test)

print(f"Accuracy: {accuracy}")

**🔧 Troubleshooting**

**Common Issues**

**1. NR Module Conflicts**

**Error**: HexagonalWraparoundModel has already been registered **Solution**: Use the simplified generator (simple\_ddos\_dataset.cc) instead of the NR-based version.

**2. Build Failures**

**Error**: Missing dependencies **Solution**: Install all required packages:

sudo apt install -y build-essential g++ python3 python3-pip git cmake \

qtbase5-dev libxml2-dev libsqlite3-dev libboost-all-dev

**3. No CSV Output**

**Error**: Simulation runs but no CSV files created **Solution**: Check output path permissions and ensure the directory exists:

mkdir -p /root/ns3-datasets

chmod 755 /root/ns3-datasets

**4. WSL Path Issues**

**Error**: Windows paths not accessible in WSL **Solution**: Use WSL paths or copy files to Linux filesystem:

cp "/mnt/e/path/to/file" ~/local/path/

**Performance Optimization**

**Build Optimization**

# Use multiple cores for building

cmake --build build -j$(nproc)

# Disable unnecessary modules

cmake -S . -B build -DNS3\_EXAMPLES=ON -DNS3\_TESTS=OFF

**Simulation Optimization**

# Reduce simulation time for testing

--simTime=5

# Use smaller node counts for faster execution

--ueTotal=5

**Debugging**

**Enable Verbose Output**

# Add logging to see simulation progress

export NS\_LOG="SimpleDdosDataset=level\_info"

**Check Generated Files**

# Verify CSV files are created

ls -la /root/ns3-datasets/

wc -l /root/ns3-datasets/\*.csv

**🤝 Contributing**

**Adding New Features**

1. **New Attack Types**: Modify the attack generation logic in the dataset generator
2. **Additional Metrics**: Add new columns to the CSV output
3. **Protocol Support**: Extend transport protocol support
4. **Mobility Models**: Add new mobility patterns

**Code Structure**

* **Main Logic**: simple\_ddos\_dataset.cc - Core simulation and CSV generation
* **Batch Processing**: run\_scenarios.ps1 - Automated batch execution
* **Configuration**: Command-line parameters for scenario customization

**Testing**

# Run smoke test

./build/scratch/ns3-dev-simple\_ddos\_dataset-default \

--ueTotal=5 --mobile=0 --enableAttack=0 --simTime=5 \

--outPath=./test --scenarioTag=smoke

# Verify output

ls -la ./test/

head -5 ./test/dataset\_smoke\_ue5\_udp\_benign.csv

**📚 References**

* [NS-3 Documentation](https://www.nsnam.org/documentation/)
* [5G NR Module](https://gitlab.com/cttc-lena/nr)
* [Machine Learning for Network Security](https://ieeexplore.ieee.org/document/1234567)

**📄 License**

This project is licensed under the MIT License - see the LICENSE file for details.

**🆘 Support**

For issues and questions:

1. Check the troubleshooting section above
2. Review NS-3 documentation
3. Open an issue in the project repository

**Generated by**: 5G NR DDoS Detection Dataset Generator  
**Version**: 1.0  
**Last Updated**: September 2025