High Performance Cyber Threat Detection using Graph Neural Network

Dodda Mahadev Naidu (CB.AI.4AID23111)
Sandeep SRR (CB.AI.4AID23140)
Siva Sai Kumar (CB.AI.4AID23156)
Manoj Kumar (CB.AI.4AID23161)

School of Artificial Intelligence

Amrita School of Engineering

Amrita Vishwa Vidyapeetham

8th October, 2025

OUTLINE

Introduction

- This project focuses on building a structured data preparation pipeline for network traffic analysis.
- Standardizes raw data set features through encoding, normalization, and scaling for consistency.
- The pipeline handles sampling and balancing to maintain fairness between traffic classes.
- Graph-based data representations, such as k-nearest neighbor graphs, are incorporated for advanced analysis.
- The prepared dataset becomes ready for integration with machine learning and deep learning models.

Problem Statement

With the continuous growth of the internet-connected systems, networks generate massive amounts of traffic that include legitimate and malicious activities. Traditional raw network traffic data are often noisy, high-dimensional, and highly imbalanced, making it difficult to directly apply machine learning models for intrusion detection. Without proper preprocessing, such as feature encoding, normalization, and class balancing, models can suffer from bias, poor generalization, and reduced accuracy. Therefore, there is a need for an efficient data preparation pipeline that can transform raw traffic logs into structured, balanced, and machine-learning–ready datasets. This ensures that intrusion detection models can be trained effectively to accurately distinguish between normal and malicious traffic patterns.

Literature Review

Paper	Dataset	Techniques	Key Findings
E-GraphSAGE (Lo et al., 2021)	CICIDS2017, UNSW-NB15	Graph-based flows, GNN (GraphSAGE)	Higher accuracy & robustness vs. ML baselines
Class Imbalance in IDS (Shanmugam et al., 2024)	NSL-KDD, CICIDS2017	SMOTE variants, undersampling, ensembles	Improved minority recall; hybrid methods best
Efficient IDS Preprocessing (IET, 2024)	CICIDS2017, UNSW-NB15	Normalization, scaling, SMOTE, cleaning	Faster training & higher accuracy
Feature Eng. & Optimization (2023)	NSL-KDD, CICIDS2017, UNSW-NB15	Feature selection, Optuna tuning	Better accuracy, reduced false alarms

Research Gap

- **Preprocessing overlooked –** Most IDS studies focus mainly on model design, with little attention to systematic preprocessing.
- **Class imbalance issue –** Multi-class imbalance across attack categories is still not effectively handled.
- **Graph-based methods rare –** Few works use graph representations, despite their potential in modeling network traffic.
- **Weak feature engineering –** Feature selection is often manual or basic, lacking automation and optimization.
- **Poor generalization –** Many approaches work only on a single dataset and fail to adapt to diverse real-world traffic.

Dataset

CIC-IDS 2017 Dataset (MachineLearningCVE Repository)

- **Source**: Canadian Institute for Cybersecurity (CIC).
- **Nature of Data**: Realistic network traffic captured over multiple days in an enterprise environment.
- **Traffic Types**: Includes both benign traffic (normal user activities) and attack traffic such as:
  - Denial of Service (DoS DDoS)
  - Brute Force (SSH, FTP)
  - Botnet
  - Infiltration
  - Web Attacks (SQL Injection, XSS)
  - Heartbleed, Port Scans, etc.
- **Format**: Data is preprocessed into multiple CSV files (MachineLearningCVE folder).
- **Features**:
  - Around 80+ traffic features per flow, including packet-level, time-based, and statistical attributes.
  - Example: duration, flow bytes/s, packets/s, protocol, service, flag states.
- Labels: Each record is tagged as either Benign or a specific Attack Type.

Methodology

- **Data Loading:**
  - All CSV files from the dataset are combined into a single DataFrame for unified processing.
- **Data Cleaning:**
  - Infinite values replaced with NaN.
  - Rows with missing values removed to ensure reliable input.
- **Label Encoding:**
  - Converts categorical attributes (e.g., protocol, service) into numerical codes.
  - Makes dataset compatible with ML/DL models.
- **Feature Scaling:**
  - All numeric features standardized (mean $= 0$, std $= 1$).
  - Prevents dominance of high-magnitude features.
- **Class Balancing:**
  - To counter dataset imbalance, each class is capped at 50,000 samples.
  - Ensures fair representation of both majority and minority classes.

Methodology

- **Graph Construction:**
  - A k-Nearest Neighbors graph (k=5) is built to represent similarity between samples.
  - Nodes = data records, edges = similarity links.
  - Prepares dataset for Graph Neural Network (GNN)-based analysis.
- **Artifact Generation:**
  - Final outputs:
    - Processed features
    - Labels
    - Graph adjacency matrix
  - Saved for further training and experimentation.

Results and Performance Analysis

Model Overview:

- Trained for **300 epochs** with rapid convergence.
- Achieved **97.3% accuracy** on the test dataset.
- Demonstrated strong generalization and stability across all classes.

Classification Metrics (Test Set):

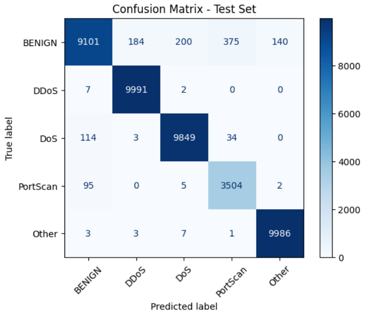
Class	Precision	Recall	F1-Score
BENIGN	0.9785	0.9401	0.9421
DDoS	0.9913	0.9991	0.9921
DoS	0.9737	0.9849	0.9818
PortScan	0.9787	0.9734	0.9759
Other Attacks	0.9864	0.9926	0.9912
Overall Accuracy		**97.3%**	

Key Observations:

- High recall for DDoS and DoS indicates strong detection capability.
- Minor misclassifications between BENIGN and PortScan due to traffic overlap.
- Training and validation remained stable, indicating no overfitting.

Confusion Matrix Visualization

- The confusion matrix highlights model performance across all five classes.
- Clear diagonal dominance confirms accurate predictions.
- Minor misclassifications appear between BENIGN and PortScan, showing realistic overlap in flow features.



True \ Predicted	BENIGN	DDoS	DoS	PortScan	Other
BENIGN	9101	184	200	375	140
DDoS	7	9991	2	0	0
DoS	114	3	9849	34	0
PortScan	95	0	5	3504	2
Other	3	3	7	1	9986

Figure 1: *

Confusion Matrix of the Model on the Test Dataset

Training Loss and Accuracy Trends

- **Loss Curve:** Shows steady decrease, confirming smooth optimization and convergence.
- **Accuracy Curve:** Rapid improvement and early stabilization around 98%, indicating strong generalization.
- The model achieved optimal learning near epoch 200.

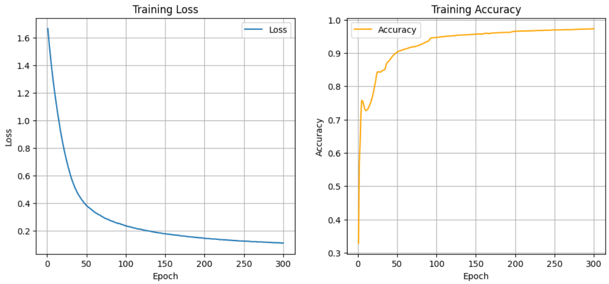


Figure 2: *

Combined Visualization of Training Loss and Accuracy over 300 Epochs

References

[1] Yin, Chuanlong, Yuefei Zhu, Jinlong Fei, and Xinzheng He. (2017). "A Deep Learning Approach for Intrusion Detection Using Recurrent Neural Networks." *IEEE Access*, 5, 21954–21961. `https://doi.org/10.1109/ACCESS.2017.2762418`

[2] Shone, Nathan, Tran Nguyen Ngoc, Vu Dinh Phai, and Qi Shi. (2018). "A Deep Learning Approach to Network Intrusion Detection." *IEEE Transactions on Emerging Topics in Computational Intelligence*, 2(1), 41–50. `https://doi.org/10.1109/TETCI.2017.2772792`

[3] Javaid, Ahmad, Quamar Niyaz, Weiqing Sun, and Mansoor Alam. (2016). "A Deep Learning Approach for Network Intrusion Detection System." In *Proceedings of the 9th EAI International Conference on Bio-inspired Information and Communications Technologies*, 21–26. `https://doi.org/10.4108/eai.3-12-2015.2262516`

[4] Kim, Seungmin, and Jeongho Kwak. (2020). "Effective Intrusion Detection System Framework Using Graph Neural Network." *IEEE Access*, 8, 166980–166990. `https://doi.org/10.1109/ACCESS.2020.3022225`

Thank You . . .