Air University, Multan Campus

Department of Computer Science

Course Title: Artificial Intelligence (CYS-340)

Project Title:

"AI-Based Intrusion Detection System"

An Advanced Deep Learning Approach for Intelligent Network Security Monitoring

Submitted By:

Name	Reg ID
Danish Manzoor	223731
Abdul Wahab	223617
Malik Arbab Tahir	223613
Muhammad Ibrahim	223623

BS CYBER SECURITY F-22 (2022-2026)

Supervised By:

Mr. Babar Ahmad
Lecturer, Department of Computer Science

Submission Date: June 2025

Contents

Title: AI-Based Multiclass Intrusion Detection System Using Convolutional Neural Networks (CNN) on the NSL-KDD Dataset

---

1. Introduction

In today's hyper-connected digital world, cybersecurity threats are rapidly evolving. Organizations face a wide range of network-based attacks, from simple probes to complex intrusion attempts. Traditional intrusion detection systems (IDS) rely on rule-based methods that are limited in scalability and adaptability, often failing to detect novel or zero-day attacks. As a result, there is a growing need for intelligent, data-driven systems capable of learning from historical data and generalizing to unseen patterns.

Artificial Intelligence (AI), particularly Deep Learning (DL), has shown great promise in addressing these challenges. Deep learning models, such as Convolutional Neural Networks (CNNs), can automatically learn intricate patterns in high-dimensional data, making them well-suited for complex tasks like network intrusion detection. This project presents an AI-based multiclass intrusion detection system using a CNN trained and evaluated on the NSL-KDD dataset. Traditional IDS implementations rely on signature-based or rule-based detection, which are often ineffective against novel or zero-day attacks. These systems also struggle with scalability, dynamic environments, and evolving attack patterns. Therefore, a shift toward intelligent, data-driven approaches has become essential.

Artificial Intelligence (AI) and, more specifically, Deep Learning (DL) offer advanced methods for detecting and classifying network intrusions. Convolutional Neural Networks (CNNs), known for their pattern recognition capabilities, are particularly well-suited for analyzing structured data such as network traffic. This project presents a CNN-based multiclass classification model trained on the NSL-KDD dataset to detect five categories of network activity: Normal, DoS, Probe, R2L, and U2R.

---

2. Dataset Overview

The NSL-KDD dataset is a refined version of the KDD'99 dataset, which addresses several key issues such as redundant records, class imbalance, and bias toward frequent records. It includes a comprehensive set of features and a well-balanced distribution of attack and normal data.

- *Classes:*

- *Normal: Legitimate, benign traffic.*
- *DoS (Denial of Service): Attacks aimed at making a system or service unavailable.*
- *Probe: Scanning and surveillance activities.*
- *R2L (Remote to Local): Unauthorized access from a remote machine.*
- *U2R (User to Root): Unauthorized root access from a user-level account.*

Each record in the dataset contains 41 features representing connection attributes and an associated class label.

The NSL-KDD dataset is a well-known benchmark for evaluating intrusion detection systems. It was designed to overcome the shortcomings of the KDD'99 dataset, which included issues like redundant records and skewed class distributions that could lead to biased models.

Each record in the NSL-KDD dataset represents a single connection and consists of 41 features grouped into categories:

- ***Basic features**: e.g., duration, protocol_type*
- ***Content features**: e.g., number of failed logins, hot indicators*
- ***Traffic features**: e.g., same_srv_rate, dst_host_count*

Target Classes:

- ***Normal**: Legitimate, benign network traffic.*
- ***DoS (Denial of Service)**: Attacks designed to overwhelm resources (e.g., neptune, smurf).*
- ***Probe**: Surveillance or information gathering (e.g., satan, nmap).*
- ***R2L (Remote to Local)**: Attempts to gain access from a remote system (e.g., guess_passwd).*
- ***U2R (User to Root)**: Exploits to gain root access (e.g., buffer_overflow).*

The NSL-KDD dataset has two main components:

- ***KDDTrain+**: 125,973 records*
- ***KDDTest+**: 22,544 records*

---

3. Data Preprocessing

To prepare the dataset for training a deep learning model, the following steps were applied:

3.1 Feature Engineering

- *Dropping Irrelevant Columns: The difficulty column was removed as it does not contribute to model performance.*

- *Label Mapping: Attack labels were mapped to one of the five broader categories: Normal, DoS, Probe, R2L, U2R.*

3.2 Normalization

- *Numerical features were standardized using StandardScaler from Scikit-learn. This transforms features to have zero mean and unit variance, which accelerates and stabilizes CNN training.*

3.3 Encoding Categorical Variables

- *Categorical features such as protocol_type, service, and flag were one-hot encoded to convert them into binary vectors.*

3.4 Label Encoding

- *The target labels were encoded to integer values (0 to 4), followed by one-hot encoding to make them compatible with the softmax output layer of the CNN.*

3.5 Data Reshaping

- *The input data was reshaped to 3D tensors of shape (samples, features, 1) to match the input requirement of a 1D CNN.*

---

4. Model Architecture: Convolutional Neural Network (CNN)

CNNs, although widely used in image processing, are also effective for time-series and tabular data. They can extract spatial and temporal patterns by applying convolution operations across input features.

4.1 Layers Used

- *Conv1D (32 filters, kernel size=3): Detects local patterns.*
- *MaxPooling1D (pool size=4): Reduces dimensionality and computation.*
- *Dropout (rate=0.2): Prevents overfitting.*
- *Second Conv1D + MaxPooling + Dropout*
- *Flatten Layer: Converts 3D output to 1D.*
- *Dense Layer (50 units): Learns global patterns.*
- *Output Layer (5 units with softmax activation): Provides a probability distribution across the five classes.*

4.2 Compilation and Training

- *Loss Function: categorical_crossentropy suitable for multi-class classification.*
- *Optimizer: Adam for adaptive learning rates.*
- *Epochs: 100*
- *Batch Size: 5000*
- *Validation Split: 0.2*
- *Hardware: Training executed on GPU (/GPU:0) for performance.*

---

5. Evaluation and Results

5.1 Accuracy

- *Test Accuracy: 74.41%*

5.2 Area Under Curve (AUC)

AUC measures the model's ability to distinguish between classes.

Class	***AUC***
Normal	*0.878*
DoS	*0.767*
Probe	*0.546*
R2L	*0.604*
U2R	*0.788*

5.3 Classification Report

Class	***Precision***	***Recall***	***F1-Score***	***Support***
Normal	*0.89*	*0.80*	*0.85*	*7460*
DoS	*0.81*	*0.55*	*0.65*	*2421*
Probe	*0.96*	*0.09*	*0.17*	*2885*
R2L	*0.88*	*0.21*	*0.34*	*67*
U2R	*0.66*	*0.94*	*0.78*	*9711*

- *Macro Average F1: 0.56*
- *Weighted Average F1: 0.71*

5.4 Confusion Matrix

- *The model shows strong prediction for Normal and U2R classes.*
- *Performance on R2L and Probe is weak due to class imbalance and limited samples.*

---

6. Analysis and Discussion

6.1 Strengths

- *Good Generalization: Model performs reasonably well across multiple classes.*
- *Effective for U2R: Despite its rarity, U2R is predicted with high recall (0.94).*
- *Efficient Architecture: CNN handles high-dimensional input with reduced parameters.*

6.2 Weaknesses

- *Poor Recall for Probe (0.09) and R2L (0.21): Indicates class imbalance and learning difficulty.*
- *Dataset Limitations: NSL-KDD, although improved, may still not reflect modern network behavior.*
- *Training Limitations: Further improvements possible through hyperparameter tuning.*

6.3 Proposed Improvements

- *Class Balancing: Use techniques like SMOTE or class-weighted loss functions.*
- *Ensemble Models: Combine CNN with LSTM or Random Forest.*
- *Hyperparameter Optimization: Use Random Search or Bayesian Optimization.*
- *Feature Engineering: Explore advanced features using domain knowledge.*

---

7. Conclusion

This project demonstrated that a CNN-based approach can effectively identify and classify multiple types of network intrusions. While the model achieved a respectable 74.41% accuracy, the evaluation highlighted challenges related to class imbalance and limited detection for rare attack types. Despite these challenges, the high recall for U2R attacks and strong overall accuracy indicate the potential of CNNs in intrusion detection. Future work will focus on improving recall for underrepresented classes, optimizing model architecture, and validating on real-world datasets.

Future enhancements should focus on balancing data classes, improving recall for minority classes, and testing against newer, real-world intrusion datasets. Overall, this project provides a strong foundation for building intelligent, AI-powered intrusion detection systems.

8. References

1. *Tavallaee, M., Bagheri, E., Lu, W., & Ghorbani, A. A. (2009). A Detailed Analysis of the KDD CUP 99 Data Set. Proceedings of the 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications.*
2. *NSL-KDD Dataset: https://www.unb.ca/cic/datasets/nsl.html*
3. *TensorFlow Documentation: https://www.tensorflow.org*
4. *Scikit-learn Documentation: https://scikit-learn.org*