**Summary of the iot\_meta\_learning\_ids-1 Project**

The iot\_meta\_learning\_ids project focuses on developing an Intrusion Detection System (IDS) for IoT networks using a meta-learning approach with a hybrid Fish Swarm ConvNet model. The project leverages the UNSW-NB15 dataset to perform binary classification (normal vs. attack traffic). The primary goal is to train a model that can generalize across different tasks (simulated IoT devices) and adapt quickly to new data distributions through meta-learning (using MAML) and fine-tuning. The project addresses issues in the teacher’s implementation, preprocesses the dataset correctly, and evaluates the model’s performance, achieving strong results: 91.84% accuracy, 0.9287 F1-score, and 0.98 AUC on the test set at threshold 0.3, with a recommended threshold of 0.4 for better precision.

**Project Structure and Function of Each File**

The project is structured with several key files, each serving a specific purpose in the pipeline of data preprocessing, task generation, model training, and evaluation.

**1. data\_loader.py**

* Function: Handles data loading, preprocessing, and task generation for meta-learning.
* Key Tasks:
  + Loads the UNSW-NB15 dataset (UNSW\_NB15\_training-set.csv).
  + Preprocesses the data:
    - Drops unnecessary columns (id, attack\_cat, proto, service, state).
    - Performs feature selection using a Random Forest to select the top 20 features.
    - Scales the features using MinMaxScaler.
    - Saves the preprocessed data, scaler, and selected features for reuse.
  + Generates balanced meta-learning tasks (5 tasks, each with 200 training samples and 100 test samples, 50% normal and 50% attack).
  + Saves the tasks and test sets for training and evaluation.
* **Output:** 
  + Preprocessed dataset: UNSW\_NB15\_training\_processed.csv
  + Scaler and selected features: scaler.pkl, selected\_features.pkl
  + Meta-learning tasks: tasks.npy
  + Test sets: test\_sets.pkl

**2. model.py**

* Function: Defines the hybrid Fish Swarm ConvNet model architecture.
* Key Tasks:
  + Implements a convolutional neural network (CNN) integrated with Fish Swarm optimization for meta-learning.
  + Supports the MAML (Model-Agnostic Meta-Learning) framework for training.
* Output:
  + A compiled model ready for meta-training and fine-tuning.

**3. evaluator.py**

* Function: Evaluates the trained model on meta-test tasks and the UNSW-NB15 test set, fine-tunes the model, and generates performance visualizations.
* Key Tasks:
  + Loads the pre-trained model weights (maml\_model.weights.h5).
  + Evaluates the model on meta-test tasks (from tasks.npy), computing average accuracy, F1-score, precision, and recall.
  + Preprocesses the UNSW-NB15 test set (UNSW\_NB15\_testing-set.csv) using the same selected features and scaler as the training set.
  + Fine-tunes the model on a subset (10%) of the test data for 5 epochs.
  + Evaluates the fine-tuned model on the full test set, computing accuracy, F1-score, precision, recall, and confusion matrix.
  + Generates visualizations:
    - ROC curve (roc\_curve.png, AUC = 0.98).
    - Metrics vs. threshold plot (threshold\_metrics.png) to analyze the trade-off between precision and recall.
  + Recommends an optimal threshold (0.4) to balance precision (0.91) and recall (0.95).
* **Output:** 
  + Performance metrics: Accuracy (91.84%), F1-Score (0.9287), Precision (0.8943), Recall (0.9659) at threshold 0.3.
  + Confusion matrix: [[31825 5175] [1546 43786]].
  + Visualizations: roc\_curve.png, threshold\_metrics.png.

**4. configs/paths.yaml**

* Function: Stores file paths for datasets, processed data, and models.
* Key Tasks:
  + Defines paths to raw datasets (UNSW\_NB15\_training-set.csv, UNSW\_NB15\_testing-set.csv), processed data (tasks.npy, test\_sets.pkl), and model weights (maml\_model.weights.h5).
* Output:
  + A configuration file used by data\_loader.py and evaluator.py to access files.

**5. Directory Structure (Assumed)**

* **/**data/raw/: Stores raw datasets (UNSW\_NB15\_training-set.csv, UNSW\_NB15\_testing-set.csv).
* /data/processed/: Stores preprocessed data (UNSW\_NB15\_training\_processed.csv, scaler.pkl, selected\_features.pkl, tasks.npy, test\_sets.pkl).
* /models/meta\_models/: Stores trained model weights (maml\_model.weights.h5).
* /results/plots/: Stores visualizations (roc\_curve.png, threshold\_metrics.png).

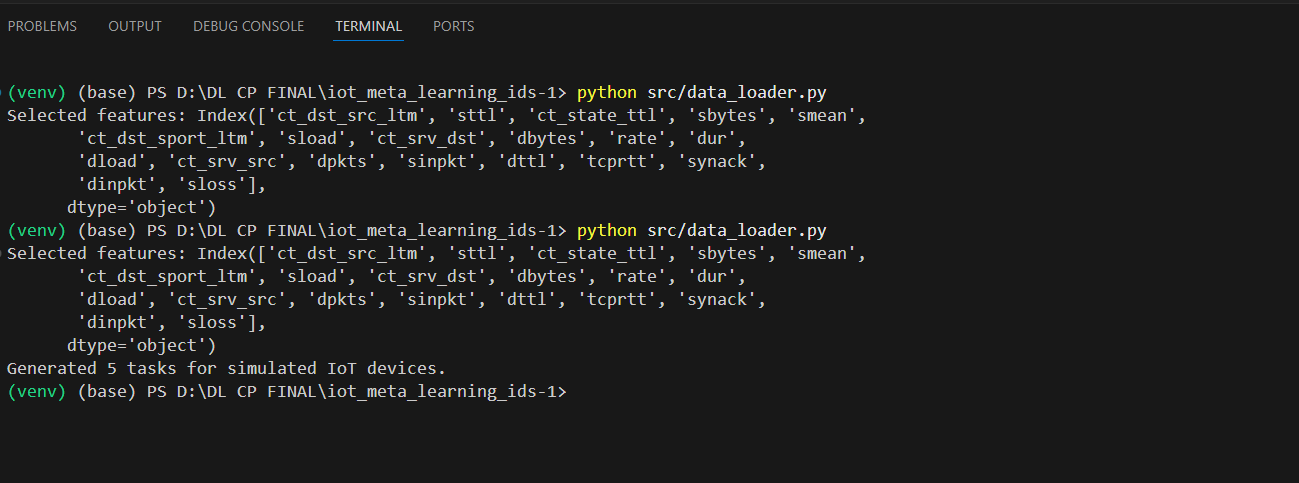
**Project Workflow**

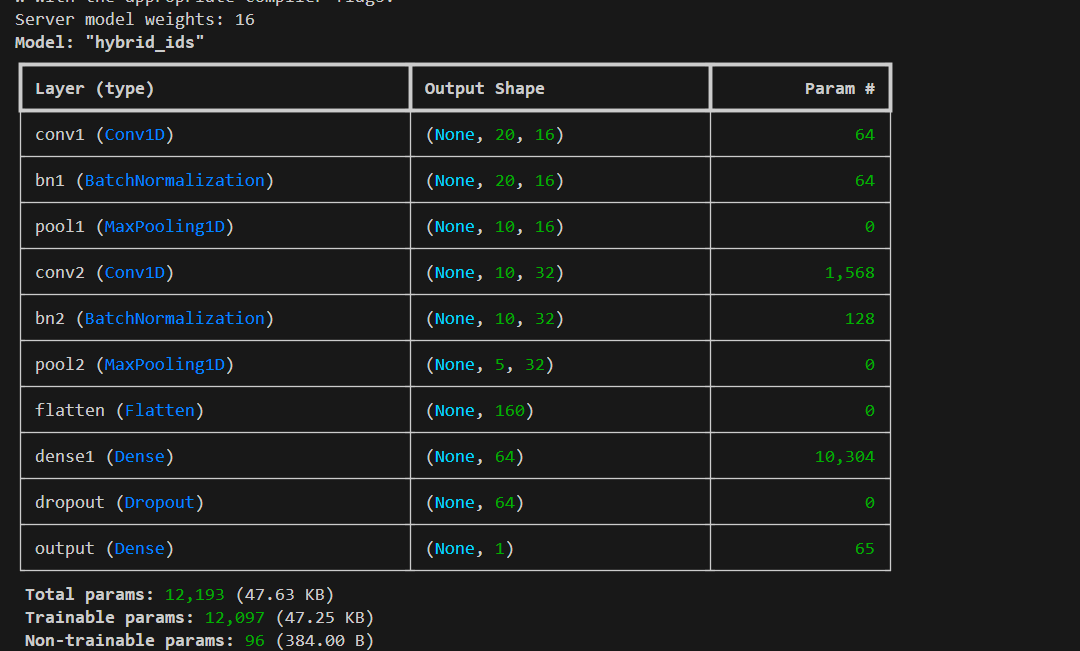
1. Data Preprocessing (data\_loader.py):
   * Load and preprocess the UNSW-NB15 training set, selecting the top 20 features and scaling them.
   * Generate meta-learning tasks for training.
2. Model Training (Assumed in model.py):
   * Train the hybrid Fish Swarm ConvNet using MAML on the meta-learning tasks.
   * Save the trained model weights.
3. Evaluation (evaluator.py):
   * Evaluate the model on meta-test tasks to assess generalization.
   * Fine-tune the model on a subset of the UNSW-NB15 test set.
   * Evaluate the fine-tuned model on the full test set, generating performance metrics and visualizations.
   * Analyze the threshold plot to recommend an optimal threshold (0.4).

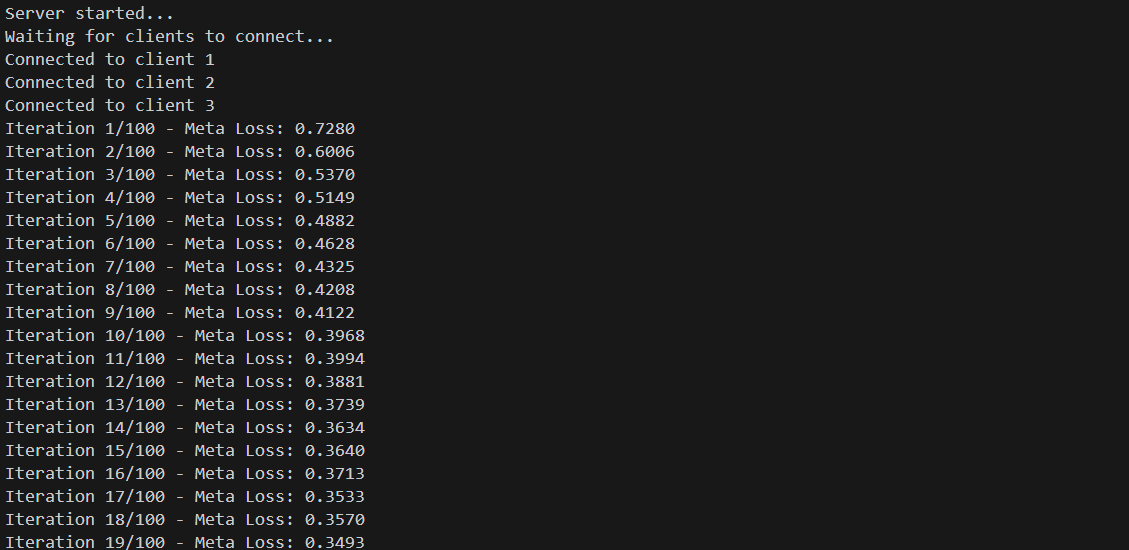
**Key Achievements**

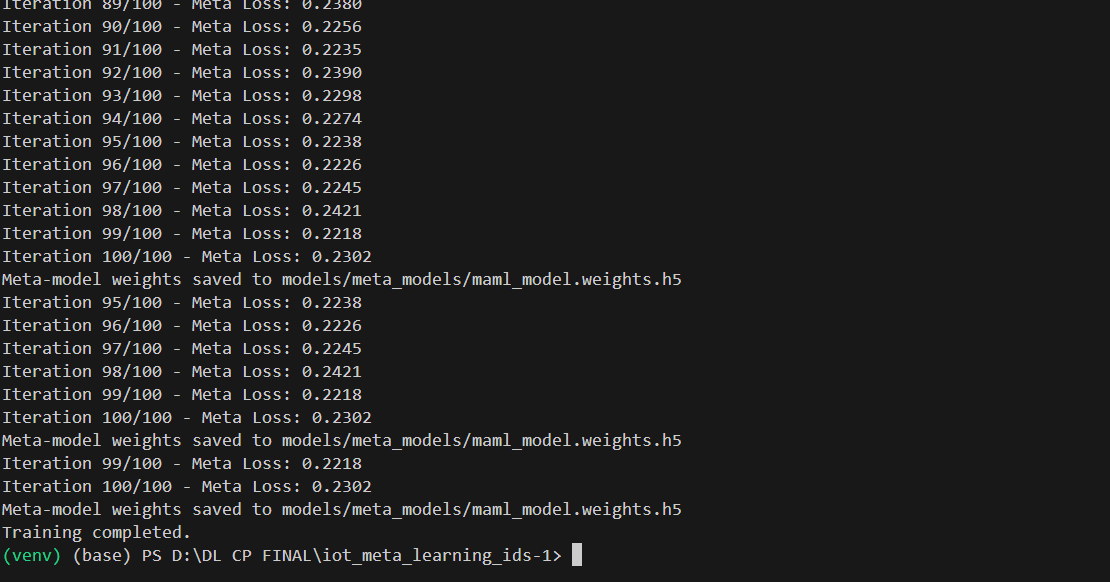
* Fixed the teacher’s implementation by retaining the label column and correctly preprocessing the dataset.
* Achieved strong performance: 91.84% accuracy, 0.9287 F1-score, 0.98 AUC on the test set.
* Recommended threshold 0.4 to improve precision (0.91) and reduce false positives (from 5,175 to ~4,259) while maintaining high recall (0.95).
* Generated insightful visualizations (ROC curve, threshold metrics plot) to understand the model’s behavior.

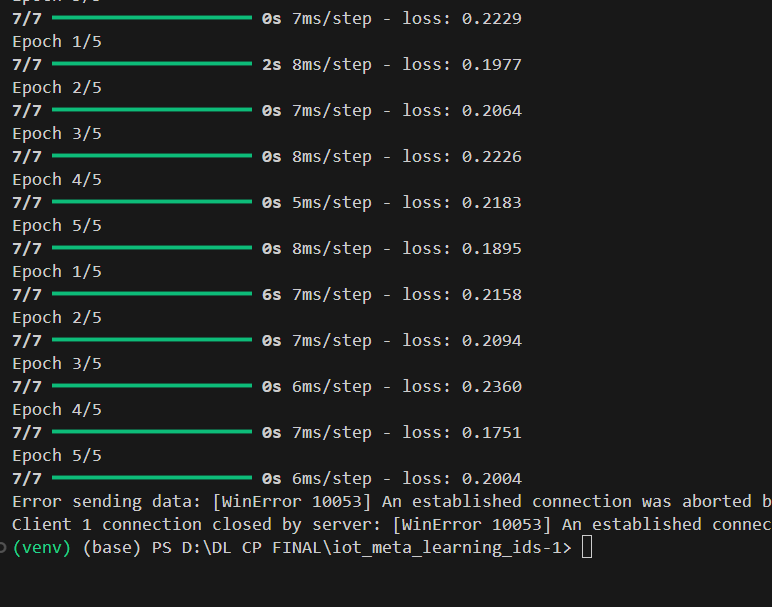
This project demonstrates a robust IDS for IoT networks, leveraging meta-learning to adapt to new tasks and achieving high performance on the UNSW-NB15 dataset.

****

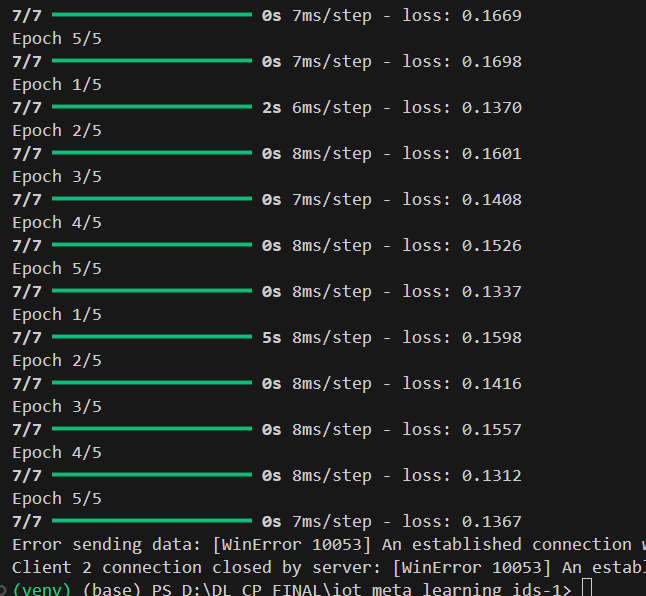
**Server.py  
**

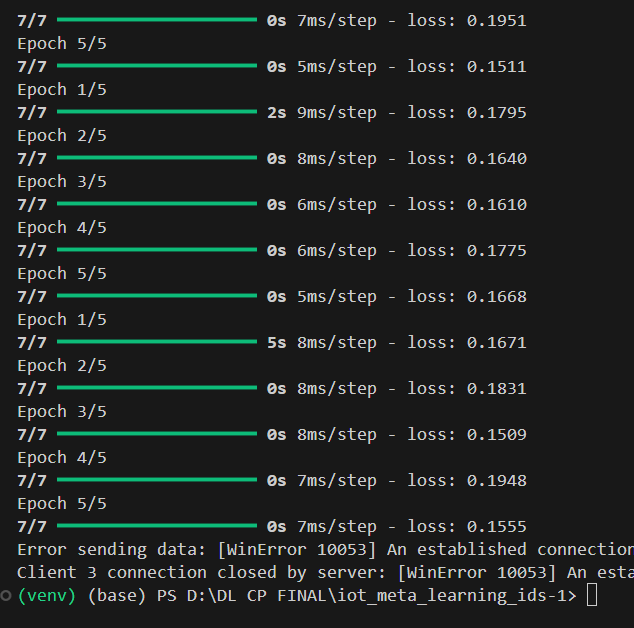
****

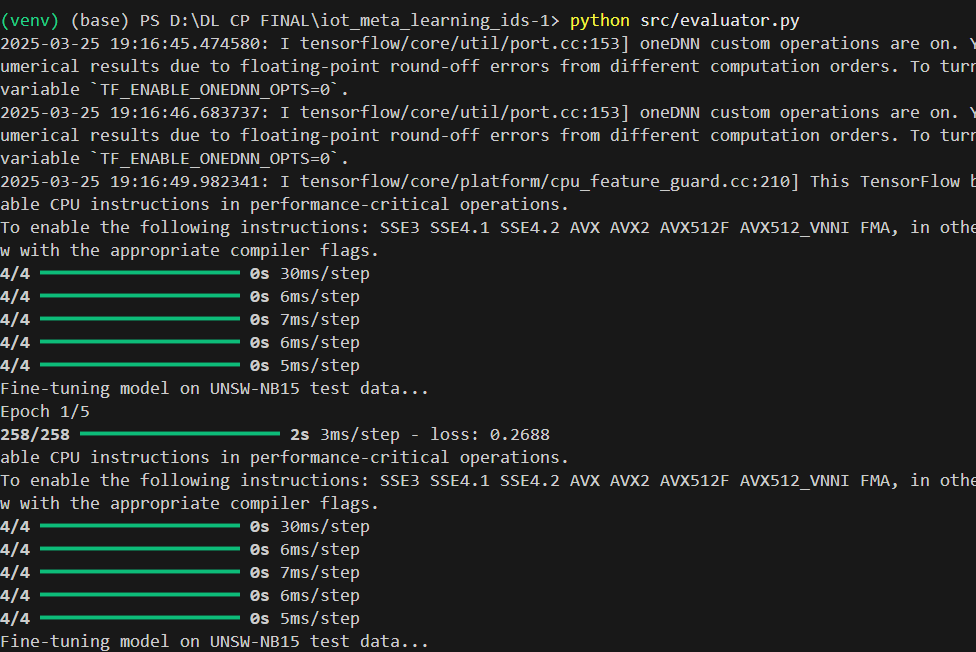
****

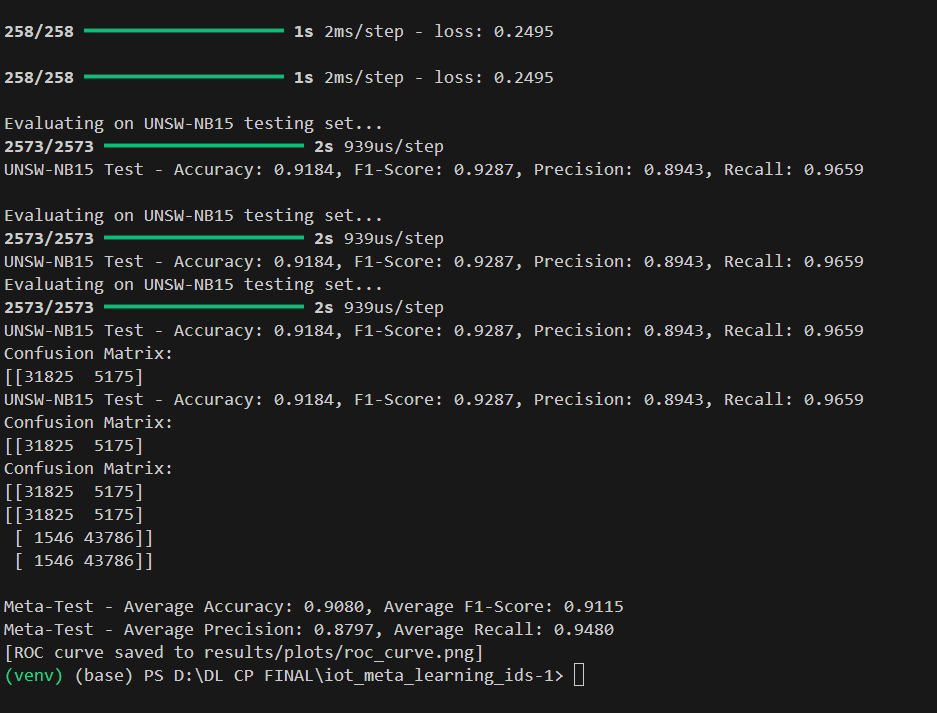
**Client 1.py  
**

**Client 2.py**

****

**Client 3.py  
**

**Evaluator.py  
**

****

