National College of Ireland

Configuration Manual

MSc Research Project
MSc in Data analytics

Prahaladh Lagadapati
Student ID:x23295881

School of Computing
National College of Ireland

Supervisor: Mr. Hicham Rifai

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: Prahaladh Lagadapati

Student ID: X23295881

Programme: MSc in Data analytics **Year:** 2024-2025

Module: MSc (Research) Practicum/Internship

Lecturer: Mr. Hicham Rifai

SubmissionDue Date: 15th September 2025

Project Title: Analyzing Machine Learning Model Synergies for Next- Generation IoT Malware Detection: A BERT and GPT Integration Approach

Word Count: 1159 **Page Count:** 8

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Prahaladh Lagadapati

Date: 15th September 2025

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Configuration Manual

Prahaladh Lagadapati
X23295881

IOT MALWARE DETECTION SYSTEM - CONFIGURATION & SETUP MANUAL

1. Project Overview

This system implements a tuned multi-model approach for IoT malware detection, comparing an enhanced XGBoost baseline, optimized ModernBERT, and a hybrid model combining ModernBERT with OpenAI embeddings. The pipeline processes ~80K network traffic samples, engineers advanced features, and provides comprehensive performance analysis with visualizations and detailed metrics for production-ready malware classification.

2. Google Colab Setup Guide

Step 1: Initial Setup

1. **Upload Your Notebook File**
    - Go to colab.research.google.com
    - Click "File" → "Upload notebook"
    - Select your file: **iot_malware_detection_tuned.ipynb**
    - Wait for upload to complete

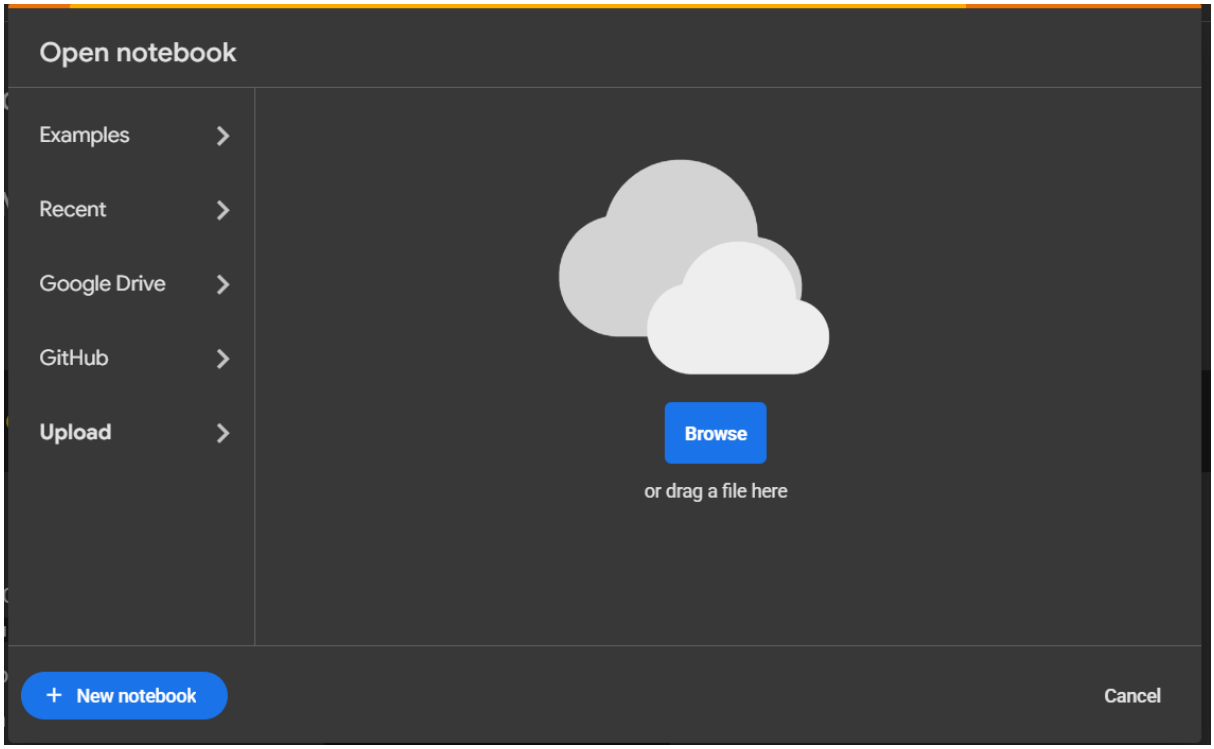


Figure 1 Upload Environment of GOogle colab

Alternative Methods:

- **From Google Drive:** Upload .ipynb to Drive → Right-click → Open with → Google Colaboratory

2. **Connect to T4 GPU** (REQUIRED)
    - Go to: Runtime → Change runtime type
    - Hardware accelerator: GPU
    - GPU type: T4 (preferred) or L4
    - Click "Save"
    - Verify GPU: Run !nvidia-smi in a cell

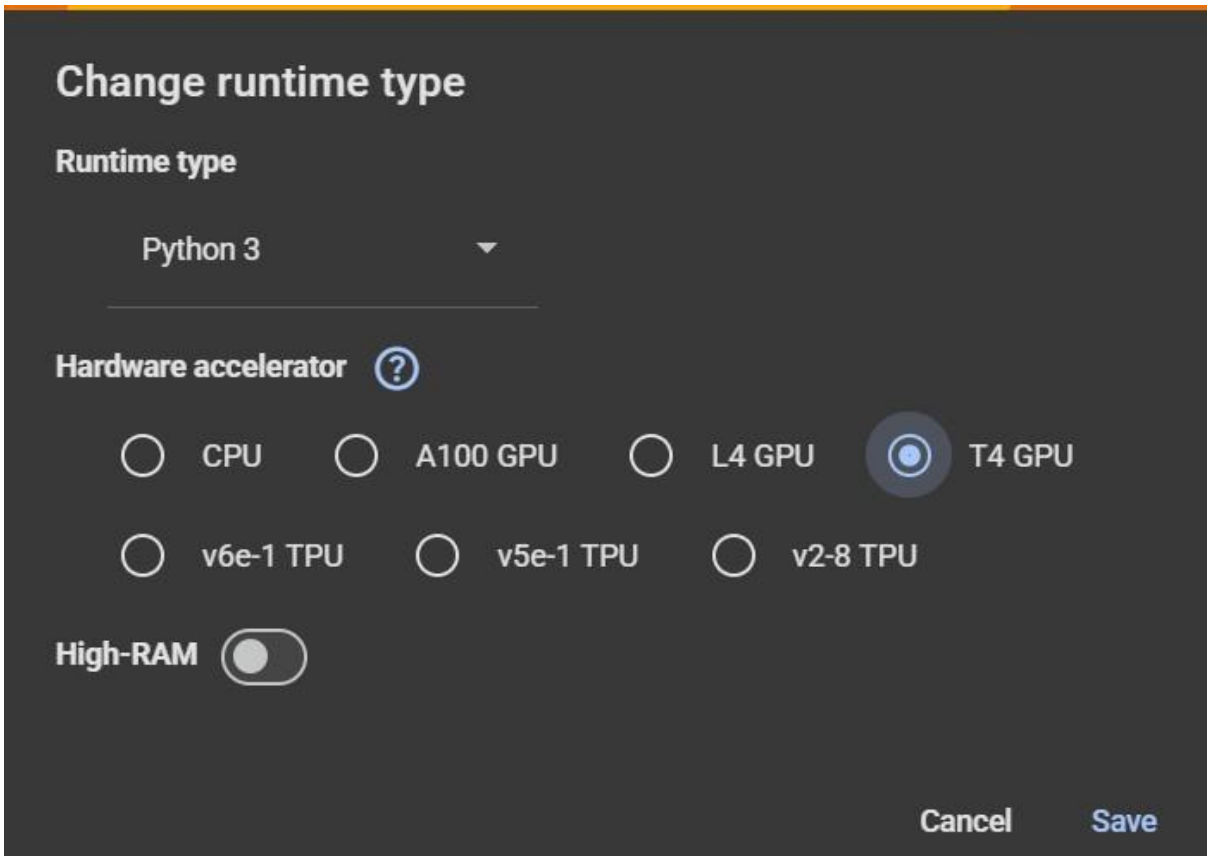


Figure 2 run time selection

Step 2: Dependencies & Environment
Create Setup Cells in Your Notebook:
Cell 1: Install Required Packages

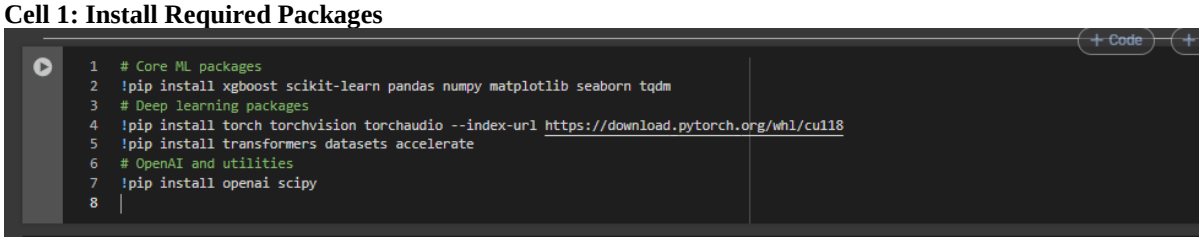


Figure 3 Basic Installations

Cell 2: Configuration & API Keys

```
import os
# CRITICAL: Replace with your OpenAI API key
OPENAI_API_KEY = "sk-proj-YOUR_KEY_HERE"  # <- CHANGE THIS
os.environ['OPENAI_API_KEY'] = OPENAI_API_KEY

```

Figure 4 API key setup

 # File paths
DATASET_PATH = "/content/cic_iot_research_dataset_final_clean.csv"
OUTPUT_DIR = "/content/iot_malware_tuned_full"

Cell 3: Upload Dataset

- Use Files sidebar (🗀) → Upload
- Select: **cic_iot_research_dataset_final_clean.csv**
- Or drag & drop the CSV file
- Verify upload:

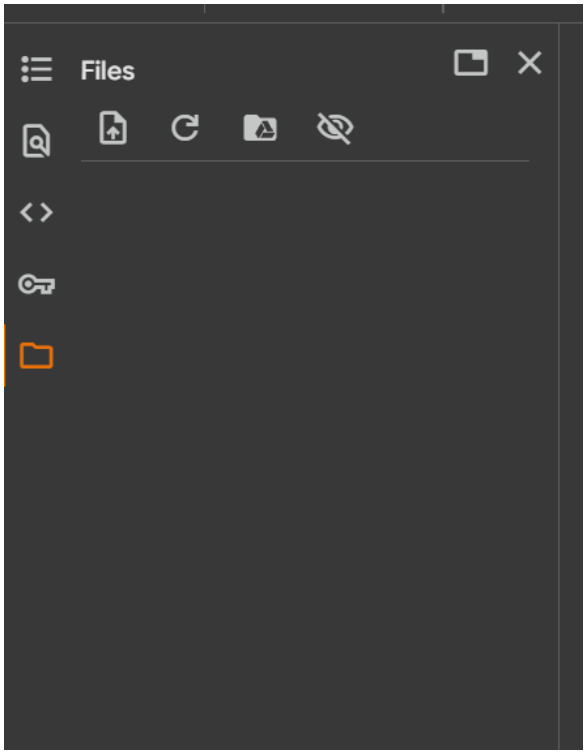


Figure 5 file upload icon

Step 3: Critical Configuration Points
In your main code, update these lines:
Dataset Path
DATASET_PATH = "/content/cic_iot_research_dataset_final_clean.csv" *# <- VERIFY PATH*
Output Directory
OUTPUT_DIR = "/content/iot_malware_tuned_full" *# <- CHANGE IF NEEDED*
OpenAI API Key
OPENAI_API_KEY = "sk-proj-YOUR_ACTUAL_KEY_HERE" *# <- MUST CHANGE*

```
67  from openai import OpenAI
68  import xgboost as xgb
69  from scipy.stats import skew, kurtosis
70
71  # Suppress warnings
72  warnings.filterwarnings('ignore')
73
74  # ============================
75  # TUNED CONFIGURATION
76  # ============================
77  class TunedConfig:
78      """Optimized configuration for full dataset"""
79
80      # Paths
81      DATASET_PATH = "/content/cic_iot_research_dataset_final_clean.csv"
82      OUTPUT_DIR = "/content/iot_malware_tuned_full"
83
84      # API Key
```

Figure 6 path configuration locations

Sample Sizes for Quick Testing (Lines ~87-89):
 # For full run (25-35 minutes):
BENIGN_SAMPLES = 40000
SAMPLES_PER_MALWARE_CLASS = 10000
NUM_EPOCHS = 3
 # For quick test (5-10 minutes):
BENIGN_SAMPLES = 2000
SAMPLES_PER_MALWARE_CLASS = 500
NUM_EPOCHS = 1

```
84      # API Key
85      OPENAI_API_KEY = "sk-proj-CabPwNnctFxcoBFYzYfVayq1ZlTiqaqpn4
86
87      # Model Configuration
88      MODERNBERT_MODEL = "answerdotai/ModernBERT-base"
89      OPENAI_MODEL = "text-embedding-3-large"
90      EMBEDDING_DIM = 3072
91
92      # FULL DATASET MODE - TUNED
93      TRAINING_MODE = "full_tuned"
94      BENIGN_SAMPLES = 40000
95      SAMPLES_PER_MALWARE_CLASS = 10000
96      NUM_EPOCHS = 3
97
```

Figure 7 dataset sample selection

Memory Settings
BATCH_SIZE = 16 *# Reduce to 8 or 4 if GPU memory errors*

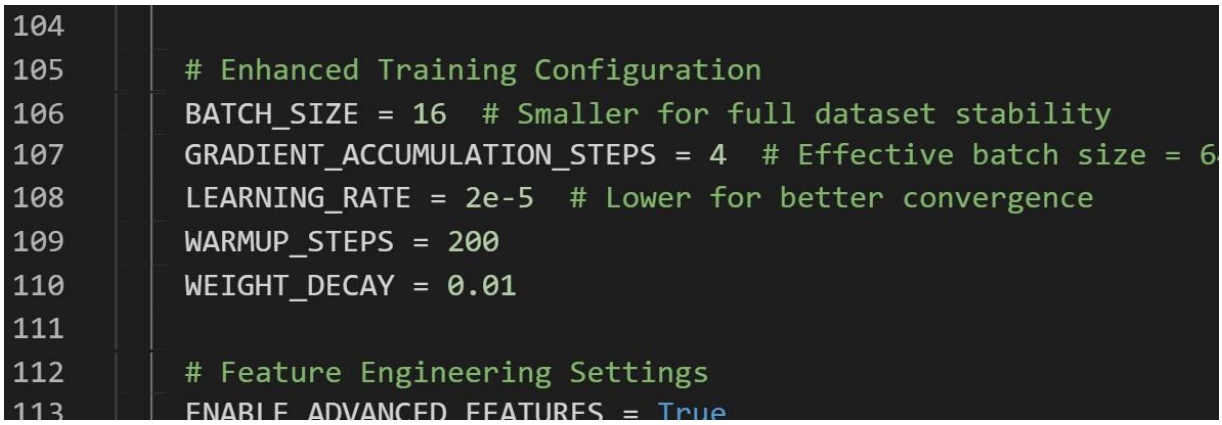


Figure 8 Batch size selection

Step 5: Execution Order
Run cells in this sequence:

1. **Setup Cells** (2 minutes)
2. Cell 1: Dependencies installation
3. Cell 2: Configuration & API keys
4. **Data Upload** (1 minute)
5. Upload CSV file via sidebar
6. **Main Execution**

 # Run the main cell containing your full code
 # Or if split into multiple cells, run sequentially

7. **Expected Runtime Breakdown:**
    - Data loading & feature engineering: 3-5 min
    - Enhanced baseline (XGBoost): 2-3 min
    - Optimized ModernBERT: 15-20 min
    - OpenAI embeddings generation: 5-8 min
    - Hybrid model training: 2-3 min
    - Evaluation & plots: 1-2 min

3. Troubleshooting
Common Errors and Solutions
1. CUDA Out of Memory Error

```
# Solution A: Clear GPU cache
import torch
torch.cuda.empty_cache()
import gc
gc.collect()
# Solution B: Reduce batch size
BATCH_SIZE = 8  # or 4
# Solution C: Restart runtime
# Runtime → Restart runtime → Re-run setup cells
```

2. OpenAI API Error

```
# Check API key validity
print(f"Key starts with: {OPENAI_API_KEY[:10]}...")
# Verify it's set in environment
print(os.environ.get('OPENAI_API_KEY', 'NOT SET'))
# Add retry delay if rate limited
```

```
time.sleep(1)  # Between API calls
```
3. Dataset Not Found
```
# Check exact location and name
!ls -la /content/
!find /content -name "*.csv"
# Verify file size (should be 100-500MB)
!du -h /content/*.csv
```
4. ModernBERT Loading Failure
```
# Clear cache and retry
!rm -rf ~/.cache/huggingface/
!rm -rf /tmp/transformers_cache/
# Restart runtime and retry
# Runtime → Restart runtime
```
5. Memory Management
```
# Monitor GPU usage
!nvidia-smi

# Monitor RAM
!free -h

# Clear large variables
del variable_name
gc.collect()
```
6. Import Errors
```
# Reinstall specific package
!pip uninstall package_name -y
!pip install package_name --upgrade
```
Quick Test Configuration

For 5–10-minute test run, create this cell:
```
# Quick test configuration
class QuickTestConfig:
    DATASET_PATH = "/content/cic_iot_research_dataset_final_clean.csv"
    OUTPUT_DIR = "/content/iot_malware_test"
    OPENAI_API_KEY = "sk-proj-YOUR_KEY"  # <- CHANGE
    # Reduced samples
    BENIGN_SAMPLES = 2000
    SAMPLES_PER_MALWARE_CLASS = 500
    NUM_EPOCHS = 1
    BATCH_SIZE = 32
    # Use same models
    MODERNBERT_MODEL = "answerdotai/ModernBERT-base"
    OPENAI_MODEL = "text-embedding-3-large"
    EMBEDDING_DIM = 3072
config = QuickTestConfig()
```
Performance Tips

- **Best GPU:** T4 (free tier) or A100
- **Best Time:** Run during off-peak hours (late night/early morning)
- **Save Progress:** Export results to Drive after each model
- **Monitor Resources:** Keep GPU usage < 90% for stability

Verification Checklist

Before Starting:

- GPU connected (check with !nvidia-smi)
- Dataset uploaded (check with !ls /content/*.csv)
- API key set (non-empty string)

- Output directory created

During Execution:

- Watch for "✅" symbols in output
- No red error messages
- Progress bars moving
- GPU memory < 15GB

After Completion:

- Results JSON file created
- Comparison plot saved
- All three models show accuracy > 70%
- Total runtime < 40 minutes

Getting Help
If errors occur:

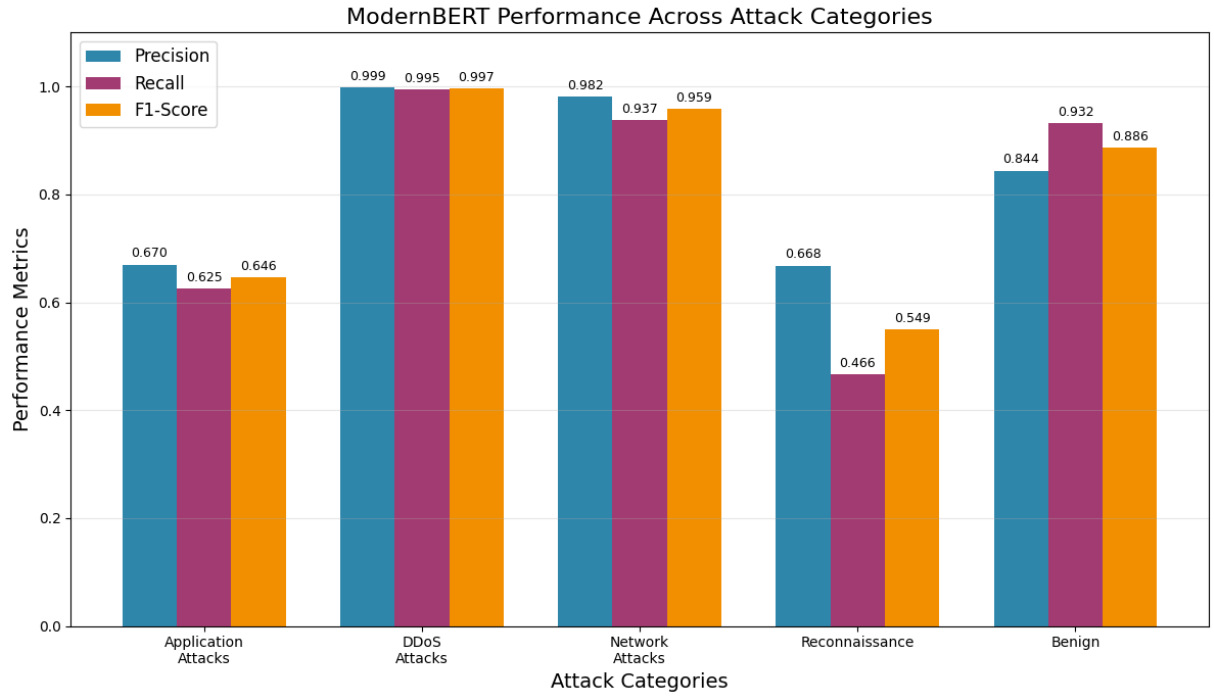
1. Copy the full error message
2. Check the line number mentioned
3. Verify all paths are correct
4. Ensure packages are installed
5. Restart runtime if needed

Resource Links:

- Google Colab Documentation: colab.research.google.com/notebooks/
- GPU Usage Guide: Check Resources monitor in sidebar
- OpenAI API Status: status.openai.com

Outputs

```
✅ GOOD performance - Strong results

================================================================================
🏆 TUNED MODEL COMPARISON - FULL DATASET RESULTS
================================================================================

📊 COMPREHENSIVE PERFORMANCE SUMMARY:
Model                Accuracy   Macro F1   Weighted F1  MCC      ROC AUC
=============================================================================
Enhanced Baseline    0.4908     0.1411     0.3360       -0.0020  0.5013
Optimized ModernBERT 0.8440     0.8075     0.8369       0.7693   0.9556
Optimized Hybrid     0.8417     0.8057     0.8350       0.7659   0.9549

🚀 TUNED IMPROVEMENT ANALYSIS:
  Optimized ModernBERT vs Enhanced Baseline:  +72.0%
  Optimized Hybrid vs Enhanced Baseline:     +71.5%
  Optimized Hybrid vs Optimized ModernBERT:  -0.3%

🏆 CHAMPION: Optimized ModernBERT (Accuracy: 0.8440)

📈 IMPROVEMENTS VS ORIGINAL RESULTS:
  Original Baseline (RF):      49.5% -> Enhanced Baseline (XGB): 49.1%
  Original ModernBERT:         78.8% -> Optimized ModernBERT:    84.4%
  Original Hybrid:             77.6% -> Optimized Hybrid:       84.2%

📊 Creating enhanced comparison visualization...
  ✅ Enhanced plot saved to: tuned_model_comparison.png
```



REFERENCES

Warner, B., Chaffin, A., Clavié, B., Weller, O., Hallström, O., Taghadouini, S., Gallagher, A., Biswas, R., Ladhak, F., Aarsen, T., Cooper, N., Adams, G., Howard, J., & Poli, I. (2024). Smarter, Better, Faster, Longer: A Modern Bidirectional Encoder for Fast, Memory Efficient, and Long Context Finetuning and Inference. *ArXiv, abs/2412.13663*.