Configuration Manual

MSc Research Project

Cyber Security

Lihao Sun

Student ID: 23330449

School of Computing

National College of Ireland

Supervisor: Raza Ul Mustafa

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Lihao Sun
Student ID:	23330449
Programm:	MSc Cyber Security **Yea:** 2024
Module:	MSc Research Project
Lecturer:	Raza Ul Mustafa
Submission Due Date:	11-Dec-2025
Project Title:	Evaluating Reasoning vs. No-Reasoning Models on Insider-Threat Datasets
Word Count:	..... **Page Count:** .....

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: Lihao Sun

Date: 08-Dec-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

Lihao Sun
Student ID: 23330449

1 Introduction

This Configuration manual provides clear setup instructions to reproduce the experiments from the thesis. It includes the required software, step by step of preparing the CERT r6.2 dataset, and all run-time settings for both the non-reasoning models (baseline encoder) and the LoRA-tuned reasoning model.

2 Experiment Overview

Experiments include:

1. Non-reasoning baseline models: TF-IDF + Logistic Regression, DistilBERT-base, DistilRoBERTa-base, Longformer-base-4096
2. Non-reasoning LLM: LLaMA-3.1 8B Instruct + LoRA (INT4)
3. Reasoning LLM: DeepSeek-R1-Distill-Qwen-7B + LoRA (INT4)

In this manual, we use DistilBERT-base and DeepSeek-R1-Distill-Qwen-7B + LoRA (INT4) as examples.

3 My Lab Hardware:

- CPU: Intel Core i9-12900KF
- RAM: 64 GB
- Storage: ≥ 1 TB NVMe SSD (≥ 2 GB/s read)
- GPUs:
    - RTX 5080 (16 GB VRAM) for baseline encoders
    - RTX 6000 Pro (96 GB VRAM) for 7B/8B LoRA runs
- OS: Ubuntu/Pop!_OS 24.04 (or equivalent Linux)
- CUDA: 12.x runtime compatible with the listed GPUs

4 Software Environment:

```
# For  ML/NLP
transformers>=4.44.2
datasets>=3.0.1
accelerate>=0.34.0
safetensors>=0.4.3
huggingface_hub>=0.24.5
sentencepiece>=0.2.0

# For  Training utilities
tqdm>=4.66.4
```

```
scikit-learn>=1.4.0

# Data
pandas>=2.2.2
pyarrow>=16.1.0
fastparquet>=2024.5.0

# LoRA / 4-bit
peft>=0.12.0
bitsandbytes>=0.43.1; platform_system == "Linux" and platform_machine == "x86_64"

# Visual
matplotlib>=3.8.0
```

5 Repository and Layout:

A demonstration of this project is available in the GitHub repository:
https://github.com/sunlihao3000/thesis-2026

Repository layout:

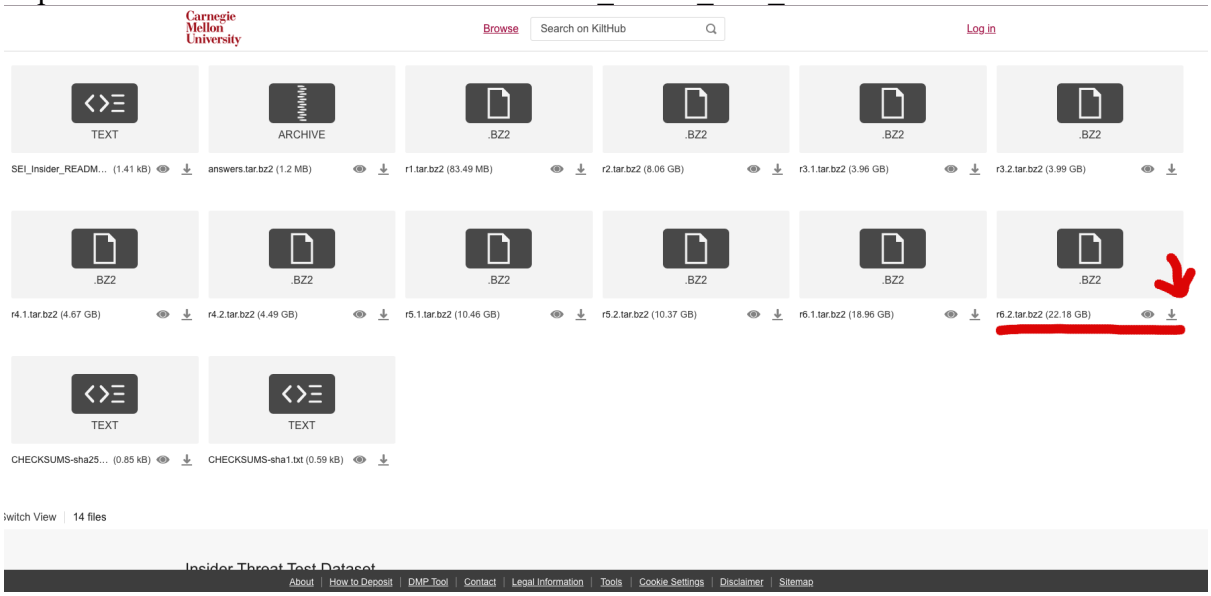
```
repo_root/
        data/
        raw/cert_r62/
        processed/cert_trainsets/
        trainsets/
        runs/
        scripts/
                calibrate_temperature.py
                cert_answers_to_windows.py
                cert_month_slice.py
                cert_to_4w_sessions.py
                eval_cert_deepseek_qwen7b_lora.py
                eval_distilbert_full.py
                label_cert_sessions.py
                metrics_from_logits.py
                train_cert_deepseek_r1_qwen7b.py
                train_distilbert_full.py
        configs/
        DistilBERT_CERT_4W_demo.ipynb
        ReadMe.md
        cert_demo_train_500.parquet
        cert_demo_val_500.parquet
        requirements.txt
# Note: Some empty folders must be created manually.
```

6 Dataset Configuration (CERT r6.2):

Due to the dataset's size, we could not upload it to the GitHub repository.

To reproduce the experiments, download the CERT r6.2 dataset from the official source [1]:

https://kilthub.cmu.edu/articles/dataset/Insider_Threat_Test_Dataset/12841247



Place the raw CERT logs under the following directories (create them if they do not exist):

data/raw/<here>/
data/trainsets/<here>/

Required files for full training typically include:

- logon.csv
- http.csv
- file.csv
- email.csv
- device.csv
- insiders.csv (ground truth)

7 Dataset Sessionisation & 4W Tokenisation:

Step 1 — Slice raw CERT logs by month (Python)

Code example:

```
python scripts/cert_month_slice.py \
  --in data/raw/2010/ \
  --out data/monthly/2010-06 \
  --month 2010-06
```

Step 2 — Convert monthly Parquet files to 4W sessions

Code example:

```
python scripts/cert_to_4w_sessions.py \
  --in-dir data/monthly/2010-06 \
```

```
  --out data/trainsets/cert_2010-06_4w.parquet \
  --win-min 60 --stride-min 10
```

After 4W sessionisation, the data looks like:

user : U217
host : PC-01
start: 2010-06-17T08:00:00Z
text : WHEN:2010-06-17T08:00Z WHERE:user=U217 host=PC-01 …

Step 3 — Generate window-level ground truth labels

Code example:

```
python scripts/cert_answers_to_windows.py \
  --answers data/raw/insiders.csv \
  --out data/labels/cert_windows.parquet
```

Step 4 — Add ground truth labels into each 4W session

```
python scripts/label_cert_sessions.py \
  --jsonl data/pilot/cert_r62/2010-06_sessions.jsonl \
  --labels data/labels/cert_r62_windows.csv \
  --win-mins 60 \
  --outdir data/pilot/cert_r62/sessions_labeled_2010-06
```

At this point, preprocessing is complete and the 4W sessions are ready for training.

8 Model Training:

Example: Train a non-reasoning baseline (DistilBERT)

```
python scripts/train_distilbert_full.py \
  --base distilbert-base-uncased \
  --train data/trainsets/cert_train_balanced.parquet \
  --out runs/distilbert_full_3ep_lr2e5 \
  --epochs 3 \
  --batch 16 \
  --lr 2e-5 \
  --max_len 512 \
  --seed 42
```

Example: Train the reasoning model (DeepSeek R1 7B)

```
python scripts/train_cert_deepseek_r1_qwen7b.py \
  --outdir runs/deepseek_r1_7b \
  --epochs 1
  --bsz 1
  --grad-accum 2
  --lr 2e-4 \
```

```
  --max-steps 200  # example 200 max steps, the larger the steps are better but require much more time
```

9 Evaluation:

Full validation + logit export for DistilBERT

```
python scripts/eval_distilbert_full.py \
  --model runs/distilbert_full \
  --data  data/trainsets/cert_val_full.parquet \
  --out-pt runs/distilbert_full/val_logits.pt
```

200k validation (DeepSeek LoRA checkpoint)
This script is a stand-alone evaluator for an already trained LoRA checkpoint.

```
python scripts/eval_cert_deepseek_qwen7b_lora.py \
  --ckpt runs/deepseek_qwen7b_cert/checkpoint-200 \
  --base deepseek-ai/DeepSeek-R1-Distill-Qwen-7B \
  --val_path data/trainsets/cert_val_full.parquet \
  --val_limit 200000 \
  --batch_size 4 \ #on my nivida rtx 6000 pro I am able to manage the size 16
  --max_length 512 \
  --dtype bfloat16 \
  --device_map auto \
  --out runs/deepseek_qwen7b_cert/eval_val200k.json
```

10 Metrics and Temperature Calibration:

```
python scripts/metrics_from_logits.py \
  --pt runs/distilbert_full/val_logits.pt \
  --topk 50 100 1000
```

```
python scripts/calibrate_temperature.py \
  --pt runs/distilbert_full/val_logits.pt
```

References

[1]“Insider Threat Test Dataset,” kilthub.cmu.edu, Sep. 2020, doi: https://doi.org/10.1184/R1/12841247.v1.