“基于大模型的攻击结果识别”比赛技术说明

# 样例数据

* **少量被打**标签的样例数据（30条）
  + 文件名：labeled\_demo\_data\_aitmg\_202504.csv
  + 数据字段：uuid, req, rsp, label
  + 具体路径：/home/jovyan/demo\_data/labeled\_demo\_data\_aitmg\_202504.csv
* **大量未打**标签的样例数据（16822条）
  + 文件名：unlabeled\_demo\_data\_aitmg\_202504.csv
  + 数据字段：uuid, req, rsp
  + 具体路径：/home/jovyan/demo\_data/unlabeled\_demo\_data\_aitmg\_202504.csv

# 开发环境

* 参赛选手需基于指定的**免费开源conda环境及Python语言**进行开发，若有新增环境依赖，需在conda环境中进行安装，最后与执行脚本一并提交。
* 通过shell命令将**基础conda环境包**和**demo数据集**拉到本地个人目录。

# 模型接口及要求

* **Qwen2.5-7B-Instruct模型**
  + 说明：约70亿参数量，基于Transformer架构，使用18T tokens的大规模语料进行预训练，显著提升了通用知识（MMLU 74.2）、代码编程（HumanEval 57.9）和数学推理（MATH 49.8）等能力，优于同等参数量其他开源大模型（如：Llama3-8B、Mistral-7B等）。该模型支持128K长文本处理，可生成8K tokens内容。
  + model\_name：**qwen7b**
* **Qwen3-8B模型**
  + 说明：约80亿参数量，最新Qwen系列模型，使用36T tokens的大规模语料进行预训练，在代码、数学、通用能力等基准测试中较同等参数量其他开源模型大幅提升。该模型支持128K长文本处理，提供快慢思考模式自由切换功能。
  + model\_name：**qwen3-8b**
* **Deepseek-R1-Distill-Qwen-7B模型**
  + 说明：约70亿参数量，基于Qwen2.5-Math-7B模型，通过知识蒸馏技术将DeepSeek-R1(671B)的推理能力迁移至轻量化版本，该模型专注于数学、代码和逻辑推理任务，其中AIME2024达到55.5，MATH-500达到92.8。
  + model\_name：**deepseek-r1-distill-qwen7b**
* **SecGPT-7B模型（深信服自研）**
  + 说明：约70亿参数量，使用200B 通用和安全混合语料进行**安全领域二次预训练**，并经过**多阶段指令微调和强化学习的**后训练得到的通专安全大模型。在通用能力和推理能力对齐同等参数量开源模型的同时，安全检测能力大幅提升。
  + model\_name：**secgpt7b**
* 请求样例

# 设置 SGLANG 服务地址和模型信息

SGLANG\_ENDPOINT = "https://aip.sangfor.com.cn:12588/v1" # SGLANG 服务地址

MODEL\_NAME = "model\_name" # 部署的模型名称，"qwen7b","deepseek-r1-distill-qwen7b","secgpt7b","qwen3-8b"

# 创建 OpenAI 兼容客户端

from openai import OpenAI

client = OpenAI(

base\_url=SGLANG\_ENDPOINT,

api\_key="sk-FRdCi5yEKxUyDcRp2fBd67921aCa4bA391506cE7C3Dd7863", # SGLANG api\_key

timeout=120000

)

# 连接参数设置（保持原有配置）

Temperature = 0

# 示例调用方式

def generate(prompt):

completion = client.chat.completions.create(

model=MODEL\_NAME,

messages=[{"role": "user", "content": prompt}],

# extra\_body={"chat\_template\_kwargs": {"enable\_thinking": False}, "top\_k": 20}, ## 当MODEL\_NAME为"qwen3-8b"时可以设置

temperature=Temperature

)

return completion.choices[0].message.content

# 输出结果要求

* 程序最终输出需保存为"**{选手工号}\_{提交时间}.csv**"文件**（包含header）**，**仅包含 uuid 和 predict 两列**，其中 uuid 为输入文件的 uuid ，predict 仅由 **SUCCESS、FAILURE、UNKNOWN** 组成。
* **{选手工号}**为**5位数字**，如：12345；**{提交时间}**格式为***yyyyMMddHHmm***，如：202505011430**（12位数字）**

# 提交步骤

## 提交运行应用

### 1.1、准备及打包

* 把所有与运行应用相关的文件一并归档上传，至少包含以下4个文件，**以下文件需放在同一级目录下**：
  + **conda环境包（可选）**：如需新增环境依赖，使用conda-pack对conda环境进行压缩打包，环境包命名格式："**aitmg\_2025\_{选手工号}**"，否则默认使用基础conda环境。

# 安装conda-pack

pip install conda-pack

# 将conda环境my\_env打包

conda pack -n my\_env -o aitmg\_2025\_{选手工号}.tar.gz

* + **run.sh（必需）**：基于指定的conda环境可以通过运行 *`sh run.sh`* 命令直接运行并得到结果，**结果文件*"{选手工号}\_{提交时间}.csv"*保存至 *run.sh*的同一级目录。**

## run.sh

#!/bin/bash

# 用户修改

python main.py --input\_file XXX(如：/home/jovyan/demo\_data/unlabeled\_demo\_data\_aitmg\_202504.csv) --output\_file XXX(如：./{选手工号}\_{提交时间}.csv)

注：

1. **run.sh**保证输出 **{选手工号}\_{提交时间}.csv** 这个文件，其中格式要求如**“输出结果要求”**所示
2. 最终测试集和**unlabeled\_demo\_data\_aitmg\_202504.csv** 格式一致，可以用提供的这个demo测试集作为验证
3. **--input\_file** 指定输入测试集路径，本地调试时指定为unlabeled\_demo\_data\_aitmg\_202504.csv路径即可**，程序提交后后台会统一自动替换为最终测试集文件**
   * **return\_result.sh（必需）**：提供结果返回脚本，通过scp传输到选手指定的个人目录。

## return\_result.sh

#!/bin/bash

# 用户修改

sshpass -p '{密码}' scp -o StrictHostKeyChecking=no -r -p -P {端口} /path/to/final/output {指定用户名}@{指定IP}:/path/to/your/dir

注：

1. **return\_result.sh**作用是将最终测试集的预测结果返回给选手本人，需提供期望返回的内网IP和路径
2. **/path/to/final/output** 指定测试结果文件路径，本地调试时指定为run.sh中 --output\_file 路径即可，**程序提交后后台会统一自动替换为最终测试结果文件**
   * **README.md（可选）**：对开发的应用进行简要说明，并写清楚应用的文件架构和执行步骤。

* 将上述文件夹打包，仅可提交.zip或.tar.gz格式的文件，提交文件需统一命名为"**{选手工号}\_{提交时间}\_AI\_TMG\_202505.zip**"或"**{选手工号}\_{提交时间}\_AI\_TMG\_202505.tar.gz**"，且**除环境包外的文件大小不得超过 3 GB**。

注：

1. 提交的**.zip文件**需能在Linux环境中通过以下命令正常解压：

**unzip** {选手工号}\_{提交时间}\_AI\_TMG\_202505.zip

1. 提交的**.tar.gz文件**需能在Linux环境中通过以下命令正常解压：

**tar -xvzf** {选手工号}\_{提交时间}\_AI\_TMG\_202505.tar.gz

* **作品提交之前需在demo数据集上完整跑通并验证输出结果符合规定的格式要求。**

### 1.2、提交应用

通过以下**submit.sh**命令将打包的文件和标识文件一并提交到指定目录。

## submit.sh

#!/bin/bash

# 用户修改

FILE\_PATH=/path/to/your/file

# 传输提交文件

sshpass -p 'sshuser@AIP1206' scp -o StrictHostKeyChecking=no -r -p -P 50075 "$FILE\_PATH" jovyan@10.72.1.16:/home/jovyan/AI\_TMG\_sectest\_202505

filename=$(basename "$FILE\_PATH")

base=${filename%%.\*}

flagname="${base}\_finish.txt"

# 创建标识文件

touch "$PWD/$flagname"

# 传输标识文件

sshpass -p 'sshuser@AIP1206' scp -o StrictHostKeyChecking=no -r -p -P 50075 "$PWD/$flagname" jovyan@10.72.1.16:/home/jovyan/AI\_TMG\_sectest\_202505

注：

1. 在测试集（约3000条）上端到端运行总时间**不得超过 1h**。
2. **（必需）**标识文件命名格式："**{选手工号}\_{提交时间}\_AI\_TMG\_202505\_finish.txt**"，用于判断提交文件是否传输完成。

# 1.3、补充说明

## 外部信息输入

可以使用IPS/WAF规则库、情报库、漏洞库作为输入（如作为RAG进行查询等），但如果模型在里面的作用较小，则评分会低。