##### Project Document

Test Plan / Test Cases Design Document

|  |  |
| --- | --- |
| Project Name | 바이너리 프로그램에서 제어 구조를 식별하는 도구 개발 |

14 조

202002514 안상준

202202602 손예진

202202487 박혜연

지도교수: 조은선 교수님 (서명)

Table of Contents

[1. Introduction 3](#_Toc197957189)

[1.1. 연구 질문/ 가설 3](#_Toc197957190)

[2. Test Plan 4](#_Toc197957191)

[3. Test Cases 5](#_Toc197957192)

[4. AI 도구 활용 정보 5](#_Toc197957193)

# Introduction

## 연구 질문/ 가설

본 연구는 다음과 같은 연구 질문에 답하고자 한다:

* **RQ1.**  
  LLM 기반 제어구조 식별 도구를 활용한 가상화 난독화 코드 분석이 기존의 방식에 비해 제어구조 식별 정확도에 어떤 영향을 미치는가?
* **RQ2.**  
  LLM 기반 제어구조 식별 도구는 난독화 난이도나 코드 복잡도에 따라 제어구조 식별 성능에 차이를 보이는가?

본 연구는 다음과 같은 가설을 설정할 수 있다:

* **H1.**  
  LLM 기반 제어구조 식별 도구를 활용한 가상화 난독화 코드 분석이 기존의 방식보다 제어구조 식별 정확도를 유의미하게 향상시킬 것이다.
* **H2.**  
  LLM 기반 제어구조 식별 도구는 다양한 난독화 난이도 및 코드 복잡도 조건에서도 안정적인 제어구조 식별에 성과 향상을 보일 것이다.

# Test Plan

|  |
| --- |
| **1. 배경과 목적** |
| 1.1 배경 |
| 이번 연구의 목표인 LLM 기반 제어구조 식별 도구는 난독화 여부와 상관없이 입력으로 주어진 바이너리 코드에 대해 원본 코드에 switch문이 포함되어 있는지를 판별한다. Switch문 포함 여부를 예측하는 분류 문제라고 할 수 있다. 따라서, 제어구조 식별 측면에서 기존 LLM에 비해 성능이 좋은 것을 입증하기 위해 정량적 분석이 필요하다. |
| **2. 테스트 상세** |
| 2.1 독립/ 종속 변수 정의 |
| 코드에 난독화를 적용하였을 경우 Flatten, Virtualize 등의 기법 중 어떤 것을 적용할 것인지에 대해서 변경할 수 있다. 이를 기반으로 원본 코드에 switch문이 포함되어있는지 분류하고 정확도(Accuracy)와 정밀도(precision), 재현율(recall), F1 Score 를 측정하여 모델의 성능을 평가한다. |
| 2.2 실험 대상/ 환경 |
| 바이너리 코드로 fine-tuning 한 모델을 사용해 정확도를 측정하고, GPT, Claude 등의 모델과 비교할 계획이다. GPT와 Claude에 대해서는 프롬프트를 동일하게 입력하여 비교의 공정성을 유지해야한다. |
| **3. 테스트 관리** |
| 3.1 실험 절차 요약 |
| - LLM 기반 제어구조 식별 도구와 기존 LLM에 대한 실험  - 각 난독화 기법이 적용된 바이너리 코드를 입력  - 모델이 원본 코드에 switch문이 포함되어 있는지 예측 |
| 3.2 측정 지표 및 도구 |
| Accuracy, precision, recall, F1 Score |

# Test Cases

|  |
| --- |
| **1. 테스트 케이스** |
| 1.1 테스트 케이스 명세 |
| |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | Id | 대상(모델/조건) | 실험 조건 | 테스트 데이터 | 평가지표 | 예상 결과 | | TC-1 | BERT model | Fine-tuning | 난독화 되지 않은 코드4,000개, 각 난독화 기법에 대한 코드 3,000개  → 총 10,000개 | Accuracy, F1 score | Accuracy 70% | | TC-2 | Chat GPT | 프롬프트 방식 적용 | 동일 | 동일 | Accuracy 50% | | TC-3 | Claude | 프롬프트 방식 적용 | 동일 | 동일 | Accuracy 50% | |
| 1.2 검증 기준(metirc) |
| (Accuracy) = (정확하게 예측한 데이터 수) / (전체 데이터 수)  (Precison) = (TP) / (TP + FP)  (Recall) = (TP) / (TP + FN)  (F1 Score) = (2 × precision × recall) / (precision + recall) |

# AI 도구 활용 정보

|  |  |
| --- | --- |
| *사용 도구* | GPT-4o-mini |
| *사용 목적* | 문장 흐름 정리 |
| *프롬프트* | * 보완할 부분 있으면 수정해서 알려줘 |
| *반영 위치* | 1. 테스트 계획 (p.4) |
| *수작업*  *수정* | 부분적 반영 |