##### Project Document

문제점 개요서

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

14 조

202002514 안상준

202202602 손예진

202202487 박혜연

지도교수: 조은선 교수님 (서명)

Document Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Rev# | Date | Affected Section | Author |
| 1 | 2025/03/19 | 문제점 개요서 작성 | 박혜연 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

[1. Survey Paper – Limitations Focus 5](#_Toc192581443)

[2. Limitations and Research Gaps 6](#_Toc192581444)

List of Figure

그림 목차 항목을 찾을 수 없습니다.

# Survey Paper – Limitations Focus

|  |
| --- |
| **작성 요령(제출 시 삭제할 것)** |
| **◦ 최신 연구 중심으로**   * 연구 개요서에서는 기존 연구의 배경을 설명하는 것이 주요 목표였지만, 이 문서는 연구의 한계점을 구체적으로 분석하는 것이 핵심임 * 최근 연구 중심으로 정리해야 함(최소 3~5년 이내) * 해결되지 않은 문제와 한계를 구체적으로 분석한 뒤 한계를 도출하는 것이 중요함. (예: 연구의 실용성(실제 적용 가능성), 데이터셋, 실험 방법의 제약 등)   **◦ 저명한 학회 및 저널 인용 필수**   * 저명한 학회 및 저널에서 발표된 논문을 인용하는 것이 중요함 (CVPR, ICML, ACL, EMNLP, ICLR, IEEE Transactions, ACM CCS) * 될 수 있으면 국내보다는 국제 학술지, 컨퍼런스 발표된 영어 논문을 인용하는 것을 권장함 * 목표로 하고 있는 컨퍼런스나 저널이 있다면, 해당 분야의 논문을 중심으로 읽고 연구 방향을 설정해보는 것을 추천함 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 번호 | 연구 제목(저자) | 저널/컨퍼런스(연도) | 주요 내용 요약 | 한계점 |
| 1 | Can LLMsObfuscate Code? A Systematic Analysis of Large Language Models into Assembly Code Obfuscation  (Seyedreza Mohseni, Seyedali Mohammadi, Deepa Tilwani, Yash Saxena, Gerald Ketu Ndawula, Sriram Vema, Edward Raff, Manas Gaur) | AAAI(2025) | 연구의 핵심 기여, 사용된 방법론, 주요 실험 내용 정리  LLM의 prompting과 in-context-learning을 위한 어셈블리 코드 난독화 데이터셋 MAD를 제공하였다. 이는 LLM의 코드 난독화 성능을 테스트하기 위해 만들어진 첫 어셈블리 코드 난독화 데이터셋이다. 이 데이터셋을 GPT, CodeLLAMA, CodeGemma 등의 모델로 pre-training 하거나 prompting 함으로써 데이터셋의 학습가능성과 신뢰성을 입증하였다.  Control Flow Change, Dead Code Insertion, Register Substitution 으로 세가지 난독화 기법에 대해 테스트를 진행하고, 결과를 Delta Entropy와 Cosine Similarity를 사용해 평가하였고, 사람이 평가하는 부분에서는 20년 이상 경력의 전문가들을 통해 진행되었다. 결과적으로, GPT 계열이 다른 Coder 모델보다도 난독화를 잘 수행했으며, 이 논문에 사용된 모델 중 가장 좋은 성능을 보였다. | 연구에서 해결하지 못한 문제, 실험 조건의 제약, 데이터셋의 한계 등  다양한 종류의 LLM을 사용했지만, fine-tuning으로 학습한 모델이 한가지 있었고, 나머지는 in-context-learning만을 사용했다. 이는 정확한 비교가 될 수 없을 것이라고 생각한다. 뿐만 아니라 모델의 예측 결과를 평가할 Delta Entropy와 Cosine Similarity 값의 범위를 실험을 통해 선정하였지만, 이는 난독화 성능에 대한 객관적인 지표가 될 수 없다. 사람이 직접 평가한 부분에 대해서도 주관이 개입되지 않았다고는 보기 어렵다. 이 논문에서 제시한 방향과 데이터셋 구축 방안은 좋은 참고가 되었지만, 난독화 성능에 대한 객관적인 지표는 확인할 수 없었다. |
| 2 | PalmTree: Learning an Assembly Language Model for Instruction Embedding (Xuezixiang Li, Yu Qu, Heng Yin) | CCS (2021) | PalmTree는 bert 기반의 모델이다. 일부를 마스킹 하는 Masked Language Model, 명령어 간의 문맥을 고려하는 Context Window Prediction, 데이터 흐름을 통해 명령어 간 데이터 흐름 관계를 학습하는 Def-Use Prediction을 사용해 사전 학습을 진행했다. 내부 평가로는 이상치 탐지, 기본 블록 유사성 검색을, 외부 평가로는 바이너리 코드 유사성 탐지, 함수 원형 추론, 메모리 영역 분석을 활용했다. 이는 기존 임베딩 방법에 비해 높은 정확도를 보였다. 따라서 PalmTree는 기존 임베딩 방법의 한계를 극복하고, 바이너리 분석에 유용한 명령어 임베딩 방법을 제안했다. 뿐만 아니라 소스 코드와 사전 학습 모델을 공개하여 다른 연구에 사용될 수 있도록 했다. | 이 논문에서 제시한 임베딩 방안은 딥러닝 모델이 바이너리 코드를 더 잘 학습할 수 있게 해준다. 그러나 Bert 기반의 transformer 네트워크를 사용하였기 때문에, 기존의 다른 모델보다 계산량이 많아 속도가 느리다는 문제가 있다. |
|  |  |  |  |  |

# Limitations and Research Gaps

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 번호 | 기존 연구 | 한계점 | 연구 필요성 | 본 연구의 기여 |
| 1 | 논문 A(저자, 연도)  Can LLMsObfuscate Code? A Systematic Analysis of Large Language Models into Assembly Code Obfuscation  (Seyedreza Mohseni, Seyedali Mohammadi, Deepa Tilwani, Yash Saxena, Gerald Ketu Ndawula, Sriram Vema, Edward Raff, Manas Gaur), AAAI 2025 | 기존 연구의 주요 한계점을 분석 (제한점, 문제점, 실험 조건 등)  다양한 종류의 LLM을 사용했지만, fine-tuning으로 학습한 모델이 한가지 있었고, 나머지는 in-context-learning만을 사용했다. 이는 정확한 비교가 될 수 없을 것이라고 생각한다. 뿐만 아니라 모델의 예측 결과를 평가할 Delta Entropy와 Cosine Similarity 값의 범위를 실험을 통해 선정하였지만, 이는 난독화 성능에 대한 객관적인 지표가 될 수 없다. 사람이 직접 평가한 부분에 대해서도 주관이 개입되지 않았다고는 보기 어렵다. 이 논문에서 제시한 방향과 데이터셋 구축 방안은 좋은 참고가 되었지만, 난독화 성능에 대한 객관적인 지표는 확인할 수 없었다. | 해결되지 않은 문제를 보완하기 위해 필요한 연구 방향 제시  우선, 다양한 LLM을 사용하여 실험해보되, fine-tuning 방법에서는 하나의 모델만을 다르게 실험하기보다는 최소 2개 이상의 모델을 같은 조건으로 학습시켜 좀 더 정확한 비교가 될 수 있도록 할 것이다. 추가적으로, 모델의 역난독화 성능을 평가하기 위한 지표에 대해서도 고민이 필요할 것 같다. | 본 연구가 어떻게 기존 연구의 한계를 해결하는지 구체적으로 설명  연구 기여를 정량적/정성적으로 설명 (예: 성능 개선, 새로운 접근법 제안 등)  아직 LLM을 가상화 난독화에 사용한 연구가 없기 때문에 이는 보안 분야에서 새로운 접근이라고 할 수 있다. 또한, LLM에 바이너리 코드를 fine-tuning 하는 것과 in-context-learning 만을 하는 것의 차이를 비교해 바이너리 코드 분석에 어떤 방법이 적합한지 분석할 수 있을 것이라고 생각한다. |
| 2 | PalmTree: Learning an Assembly Language Model for Instruction Embedding (Xuezixiang Li, Yu Qu, Heng Yin), CCS 2021 | 이 논문에서 제시한 임베딩 방안은 딥러닝 모델이 바이너리 코드를 더 잘 학습할 수 있게 해준다. 그러나 Bert 기반의 transformer 네트워크를 사용하였기 때문에, 기존의 다른 모델보다 계산량이 많아 속도가 느리다는 문제가 있다. | 모델의 레이어 수를 줄이거나, 더 효율적인 transformer 아키텍쳐를 사용해 성능은 유지하며 계산 비용만을 줄일 수 있을 것이다. | 현재는 바이너리 코드를 벡터화 하는 방법과 바이너리 코드 분석에 LLM을 사용하는 것에 대한 연구가 따로 진행되고 있다. 그러나, 이 연구에서는 두가지를 함께 진행할 예정이다. 바이너리 코드의 특성에 맞게 벡터화 한 후 LLM을 학습시킨다면 기존보다 훨씬 좋은 성능이 나올 것이라고 생각한다. |
|  |  |  |  |  |