AI 기반 시큐어 코딩 점검 도구 개발에 관한 연구

김동연, 김세진, 이도경, 이채윤, 임승연[1], 서혁준[2]
[1]동덕여자대학교 컴퓨터학과 학부생
[2]LG CNS
redlhuads069@naver.com, smilestar73@naver.com, sunny008822@naver.com,
codbs43@naver.com, lnsndus@gmail.com, suh_hyuck_jun@hotmail.com

A Study on Tools for Development of AI-based Secure Coding Inspection

Dong-Yeon Kim, Se-jin Kim, Do-Kyung Lee, Chae-Yoon Lee, Seung-Yeon Lim[1], Hyuk-Joon Seo[2]
[1]Dept. of Computer Science, Dongduk University
[2]LG CNS

요 약

시큐어 코딩은 해킹 등 사이버 공격의 원인인 보안 취약점을 제거해 안전한 소프트웨어를 개발하는 SW 개발 기법을 의미한다. 개발자의 실수나 논리적 오류로 인해 발생할 수 있는 문제점을 사전에 차단하여 대응하고자 하는 것이다. 그러나 현재 시큐어 코딩에는 오탐과 미탐의 문제가 발생한다는 단점이 있다. 따라서 본 논문에서는 오탐과 미탐이 발생하는 단점을 해결하고자 머신러닝 알고리즘을 활용하여 AI 기반으로 개발자의 실수나 논리적 오류를 탐지하는 시큐어 코딩 도구를 만들고자 한다. 다양한 모델을 사용하여 보안 취약점을 모아놓은 Juliet Test Suite를 전처리하여 학습시켰고, 정확도를 높이기 위한 과정 중에 있다. 향후 연구를 통해 정확도를 높여 정확한 시큐어 코딩 점검 도구를 개발할 수 있을 것이다.

1. 서론

미국은 2002년 연방 정보보안 관리법을 제정해 시큐어 코딩을 의무화했고, 마이크로소프트는 윈도 비스타(Windows Vista) 개발 때 이를 도입했다. 우리나라는 2009~2011년 소프트웨어 보안 약점 진단 시범사업을 했으며, 2012년 12월부터 '소프트웨어 개발 보안 의무제'가 시행됐다.[1] 갈수록 시큐어 코딩의 중요성이 커지고 있지만 시큐어 코딩에는 오탐과 미탐의 문제가 발생한다는 단점이 있다. 따라서 본 연구에서는 이 문제를 해결하기 위해 머신러닝 알고리즘을 이용하여 AI를 기반으로 코드의 보안 약점을 탐지하는 도구를 만들어 보고자 한다.

2. 에이전트 개발도구의 요구사항

1. 전처리 과정

머신러닝 학습을 위하여 Juliet Test Suite에서 취약한 코드와 취약하지 않은 코드 데이터를 가져와 주석을 제거하고 텍스트화하였다.

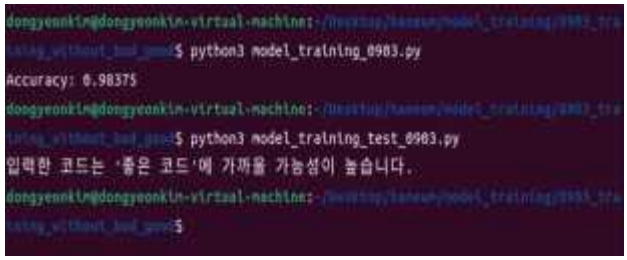
Juliet Test Suite의 Java 코드를 살펴보면, 취약한 코드(Bad)와 취약하지 않은 코드(Good)를 동일한 클래스 내의 메소드로 나타내고 있다. 따라서 Bad(취약한 코드) 메소드와 Good(취약하지 않은 코드) 메소드를 서로 분리하였다.

메소드 분리 후 취약한 코드와 취약하지 않은 코드를 구분하기 위하여 이진 분류 모델인 Logistic Regression 모델을 사용하여 학습하였다.

2. 학습 과정 -> 진행 중 (과정 정리)

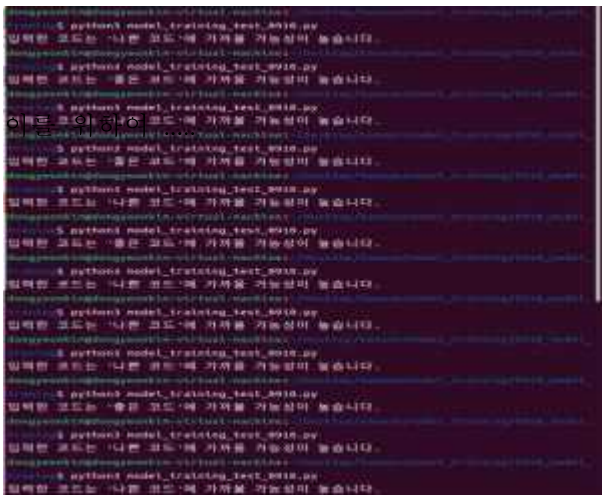
학습하는 과정에서 모델이 데이터 안에 존재하는 'good'와 'bad' 키워드를 사용하여 데이터를 분류한다는 사실을 발견하였다. 따라서 데이터의 'good'와 'bad' 키워드를 'test' 또는 'data'로 통일하였다.

그 결과 생성된 모델을 테스트했을 때, 취약한 코드를 정확하게 구분하지 못하는 문제를 발견하였다.



(그림 2) good, bad 키워드 삭제 후 결과

이 과정에서 학습 데이터의 개수가 모델의 정확도에 영향을 미친다는 것을 인지하였다. 초기 학습 단계에서는 'good' 코드의 개수가 'bad' 코드보다 훨씬 많았고, 그 후 'bad' 코드와 'good' 코드의 개수를 동일하게 조정한 후 모델을 다시 생성하여 테스트하였다.

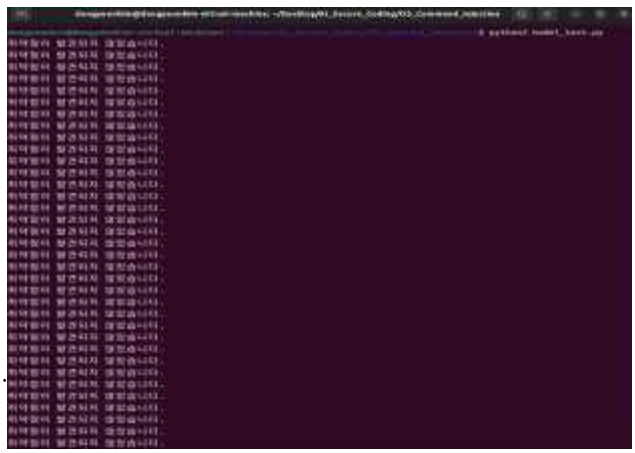


(그림 3) 데이터 개수 통일 후 결과

그 결과 80%의 정확도로 취약한 코드와 취약하지 않은 코드를 구분하는 데 성공하였다. 이와 함께 20%의 오탐률을 나타내었다.

이후 모델의 정확도를 높이기 위하여 학습 데이터의 개수를 2100여 개로 늘려 모델 학습을 진행하였다.

모델에 대한 1052번의 테스트 중 56개의 테스트에서 오류 탐지가 발생하여 총 오탐률을 20%에서 0.0532%로 낮출 수 있었다.



(그림 4) 데이터 양을 늘려서 학습한 결과

3. 결론

본 연구를 통해 머신러닝 모델이 학습 과정에서 전처리 과정과 데이터 개수에 따라 모델의 정확도가 변한다는 것을 알 수 있었다.

이를 바탕으로 더 많은 test case를 학습시켜서 모델의 정확도를 높이고, test case의 개수를 늘리는 방법 외에도 정확도를 높이고 오탐률을 낮출 수 있는 방법을 찾아 연구를 진행하면 AI 기반의 정확한 시큐어 코딩 점검 도구를 개발할 수 있다고 생각한다.

본 연구는 Java 코드만 점검할 수 있다는 한계를 갖고 있다. 하지만 본 연구를 바탕으로 다양한 언어의 코드를 점검할 수 있는 AI 기반 시큐어 코딩 점검 도구를 개발할 수 있을 것으로 기대한다.

- 본 논문은 과학기술정보통신부 정보통신창의인재양성사업의 지원을 통해 수행한 ICT멘토링 프로젝트 결과물 입니다 -

참고문헌

[1] 김영명. 시큐어 코딩, 안전한 소프트웨어 개발 위한 나침반. 보안뉴스. 2022. https://www.boannews.com/media/view.asp?idx=112603&kind=

[2] 양준혁, 모지환, 홍성문, 도경구. code2vec 모델을 활용한 소스 코드 보안 취약점 탐지. 한국소프트웨어감정평가학회논문지, 16(2), 45-52. 2020.

[3] 엄태호, 홍성문, 양준혁, 장효석, 도경구. code2vec을 이용한 유사도 감정 도구의 성능 개선. 한국소프트웨어감정평가학회논문지, 17(1), 31-40. 2021.

[4] 박양환, 최진영. 기계학습을 이용한 소스코드 정적 분석 개선에 관한 연구. 정보보호학회논문지, 30(6), 1131-1139, 2020.

[5] 이영호. 모두의 인공지능 with 파이썬. 서울. 길벗. 2020.

[6] 안상준, 유원준. 딥 러닝을 이용한 자연어 처리 입문. 위키독스. 2022.