머신러닝 기반 손상된 디지털 파일 내부 은닉 악성 스크립트 판별 시스템 설계 및 구현

이형우[1*], 나상원[2]
[1]한신대학교 컴퓨터공학부 교수, [2]한신대학교 컴퓨터공학부 학사과정

Design and Implementation of a ML-based Detection System for Malicious Script Hidden Corrupted Digital Files

Hyung-Woo Lee[1*], Sangwon Na[2]
[1]Professor, Divison of Computer Engineering, Hanshin University
[2]Student, Divison of Computer Engineering, Hanshin University

요 약 최근 MS Office 파일 내에 악성 스크립트 등이 은닉된 멀웨어 파일이 발견되고 있다. 이에 본 논문에서는 머신러닝 기법을 적용하여 악성 디지털 파일을 자동으로 검출할 수 있는 시스템을 설계 및 구현하였다. MS Office 파일 내 OLE VBA 매크로 기능을 악용하여 악성 스크립트를 검출하거나, OOXML 구조 분석을 통해 CDH/LFH/ECDH 내부 필드 값에 악성 스크립트를 탐지하고, OOXML 구조에서 참조되지 않는 비정상적인 CDH/LFH 정보를 추가한 경우 이를 검출할 수 있는 메커니즘을 제시하였다. 그리고 VirusTotal 악성 스크립트 판별 기능을 이용하여 MS Office 파일에 대한 악의적 손상 여부 자동 판별하는 기능을 이용하여 머신러닝 기반 통합 소프트웨어를 설계 및 구현하였다. 실험 결과 파일 손상 여부를 자동 판별할 수 있으며 최적의 머신러닝 모델을 이용하여 임의의 MS Office 파일에 대해 향상된 검출 성능을 제공하는 것을 확인하였다.

주제어 : MS Office 파일, 악성 스크립트, 머신러닝, 자동 검출 시스템, 소프트웨어 통합 구현

Abstract Malware files containing concealed malicious scripts have recently been identified within MS Office documents frequently. In response, this paper describes the design and implementation of a system that automatically detects malicious digital files using machine learning techniques. The system is proficient in identifying malicious scripts within MS Office files that exploit the OLE VBA macro functionality, detecting malicious scripts embedded within the CDH/LFH/ECDR internal field values through OOXML structure analysis, and recognizing abnormal CDH/LFH information introduced within the OOXML structure, which is not conventionally referenced. Furthermore, this paper presents a mechanism for utilizing the VirusTotal malicious script detection feature to autonomously determine instances of malicious tampering within MS Office files. This leads to the design and implementation of a machine learning-based integrated software. Experimental results confirm the software's capacity to autonomously assess MS Office file's integrity and provide enhanced detection performance for arbitrary MS Office files when employing the optimal machine learning model.

Key Words : MS Office File, Malicious Script, Machine Learning, Auto-Detection System, SW Implementation.

이 성과는 2023년도 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구(No 2021R1F1A1046954)이며, 이 논문의 일부는 한신대학교 학술연구비 지원에 의하여 연구되었음.
*교신저자 : 이형우(hwlee@hs.ac.kr)
접수일 2023년 9월 17일 수정일 2023년 11월 20일 심사완료일 2023년 11월 24일

1. 서론

최근 디지털 파일 내부 포맷 구조의 취약점 등을 악용하여 공격자가 생성한 악성 스크립트 등이 정상적인 형태의 디지털 파일 내에 은닉되어 있을 경우 기존의 백신 또는 디지털 포렌식 도구에서 검출되지 못하는 경우가 발생하고 있다. 특히 전세계적으로 많은 사용자가 이용하고 있는 MS Office 파일인 경우 OOXML 저장 포맷[1]의 구조적인 취약점을 이용하여 악성 스크립트 등을 삽입 또는 은닉하여 유포되고 있다[2,3]. 예를들어 OOXML 포맷을 사용하고 있는 MS-Word 파일인 경우 ZIP 압축 파일 형태로 저장되며, XML 기반으로 MS-Word 문서 내 포함된 컴포넌트를 지정하고 이를 참조하여 저장하고 있다. 하지만 OOXML 구조 기반 ZIP/XML의 보안 취약점을 악용하여 MS-Word 파일 내에 악성 스크립트 등을 은닉할 수 있으며, 악의적인 형태로 손상된 디지털 파일을 MS Office 도구를 이용하여 실행하더라도 해당 편집기 SW에서는 오류를 발견되지 못한다는 문제점이 발견되고 있다. 따라서, MS-Word 등과 같은 MS-Office 계열의 각종 디지털 파일에 대한 내부 구조 분석 및 악성 스크립트 검사 과정을 통해 인터넷/이메일 등을 통해 송수신되는 MS-Office 계열의 디지털 파일에 대한 위변조 및 파일 손상 여부를 확인하고 비정상적인 디지털 파일 내부 구조를 식별하며 내부에 은닉된 악성 파일을 탐지할 수 있는 기술이 개발되어여 한다.

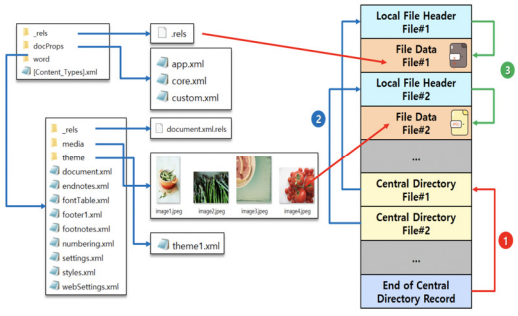
이에 본 연구에서는 머신러닝 기술[4,5,6]을 적용하여 적법한 형태로 위장한 디지털 파일 내부의 비정상적인 구조 여부 및 악성 스크립트 은닉 여부를 판별할 수 있는 메커니즘을 제시하고 이를 토대로 검출 시스템을 설계 및 구현하였다. 제안한 시스템을 이용할 경우 디지털 포렌식 관련 기술에 적용 가능하며 임의의 디지털 파일에 대한 비정상적인 위변조 여부와 악성 스크립트에 대한 은닉 여부를 효율적으로 판별할 수 있다.

2. 디지털 파일 위변조 사례

2.1 OOXML 기반 MS-Office 계열 파일 구조 분석

MS Office 2007 이후에 적용된 방식은 OOXML 기반 파일 포맷[1]을 적용하고 있다. 따라서 MS-Word 2007 이후 버전으로 생성된 파일은 ZIP 압축 방식이 적용되어 있어 그림 1과 같이 MS Office 계열 파일의 내부 구조를 확인할 수 있다. _rels 폴더와 docProps, word 폴더가 생성되며 [Content_Types].xml 파일이 포함되어 있는 것을 확인할 수 있으며 docProps 폴더내 app.xml, core.xml, custom.xml 파일을 통해 각 문서간 메타데이터, 속성 정보를 포함하고 있다. 다시 word 폴더 내에는 여러 개의 세부 폴더가 생성되며 MS-Word 문서내 텍스트 정보는 document.xml 문서 파일 안에 저장/표기되어 있고, 문서에 포함된 이미지 파일 등은 media 폴더 내에 저장되어 있는 구조이다.

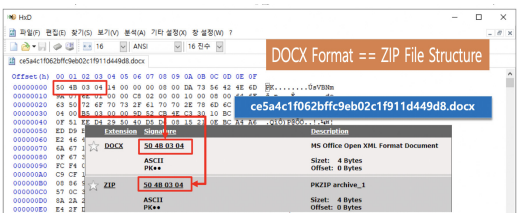
ZIP 파일 포맷 형태로 저장하면서 MS Word 내부에는 여러 개의 Central Directory(CD) 헤더 파일이 존재하며 각각의 CD 파일에는 Local File(LF) 헤더를 가리키고 있고, LF 헤더 정보에 기재된 오프셋 위치에는 해당 파일이 저장되어 있는 구조이다. 이를 다시 구조적으로 해석해 보면, MS-Word 파일 내에 포함된 각각의 파일이 저장된 정보를 표기하는 헤더(Local File Header : LFH)가 있으며, 다시 LFH 정보에 대한 정보를 포함하고 있는 헤더(Central Directory Header : CDH)가 있고, CDH 헤더들의 맨 마지막 부분을 나타내는 해더(End of Central Directory Record : ECDR)가 붙어 있는 구조이다. 그러므로 ECDR 헤더는 파일의 맨 마지막에 위치하고 있으며, 위에서 언급 했듯이 처음 시작하는 CD 해더 관련 상세 정보(시작 offset, 전체 Central Directory 사이즈 등)를 포함하고 있다. 따라서 ZIP 파일 구조를 분석하려면 제일 먼저 ECDR 헤더를 찾은 후 역순으로 이동하면서 CDH와 각각의 LFH 헤더를 찾아가면서 분석 과정을 진행하여야 한다[8].



[Fig. 1] OOXML based MS Office File Structure

예를들어 MS Word 편집기를 통해 생성된 DOCX 파일은 ZIP 파일 포맷 구조와 동일한 형태로 저장되어 있으며, 파일의 처음 2 바이트에는 File Magic Number 정보가 설정되어 있으며, ECDR, CDH 그리고 LFH의 고유 Signature 값은 각각 0x06054B50, 0x02014B50, 0x04034B50이며 이와 같은 정보를 이용하여 OOXML

기반 MS Office 파일에 대한 내부 구조를 분석할 수 있다.

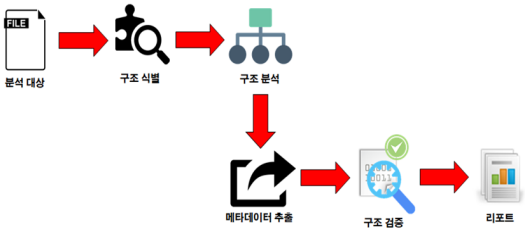


[Fig. 2] File Magic Number (Signature)

2.2 OOXML 기반 MS Office 파일 내부 구조 위변조

분석 결과 MS-Word 파일을 대상으로 위변조 과정이 실제로 적용 가능하다는 것을 확인할 수 있었다. LFH와 CDH 헤더의 내 Extra Field에 임의의 악성 스크립트를 은닉/저장할 수 있었으며, 다수의 CDH 내에 악성 스크립트를 분할하여 저장하는 것 역시 가능하였다. 또한 CDH와 LFH의 개수를 조정하거나 특정 CDH를 삭제하는 방식, 또는 CDH에 의해 참조되지 않는 LFH를 악의적으로 추가하는 방식으로도 악성 스크립트를 디지털 파일 내에 은닉 시킬 수 있었다[9].

따라서 이를 상세 기법 중심으로 정리해 보면, (1) OOXML 기반 MS Office 파일에 대한 직접적인 수정 또는 변경 과정 없이도 OOXML 기반 내부 CDH 헤더와 LFH 헤더 내에 Extra Field 부분을 이용하여 임의의 악성 스크립트를 은닉/저장하는 것이 가능하며, (2) CDH와 LFH의 개수를 조정하거나 특정 CDH를 삭제하거나 또는 CDH에 의해 참조되지 않는 LFH를 악의적으로 추가하는 것도 가능하고, (3) 이와 같은 방식으로 OOXML 기반 MS Office 파일에 대해 악의적인 위변조 과정을 수행하더라도 현행 MS Office 소프트웨어에서는 별도의 에러 표출과정 없이 정상적으로 파일이 오픈된다는 문제점이 발생한다.

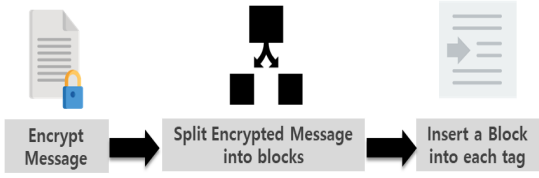


[Fig. 3] Malware Analysis and Detection Process

그러므로 그림 4와 같이 OOXML 기반 MS Office 디지털 파일 내부 구조를 식별 및 분석한 후 관련 메타데이터 정보를 추출하여 구조 검증 과정을 수행하여 디지털 파일의 손상 여부를 자동 분석할 수 있는 메커니즘을 개발하여 악성 스크립트 등에 대한 은닉 여부를 자동 판별할 필요가 있다.

2.3 MS Office 파일 내부 XML 정보 위변조

MS-Word 파일을 압축 해재하면 OOXML 형태의 파일 문서가 생성된다. 이때 내부 문서 구조를 분석해 보면 문서내 포함된 각종 엘리먼트 등 간에 충돌을 피하기 위해 고유 식별자가 부여된 네임스페이스가 있다. 따라서, OOXML 문서 내에 포함된 네임스페이스는 서로 다른 여러 개의 네임스페이스로 구성되며, 특정 문서 유형 내에서 특정 요소에 대한 네임스페이스 접두사를 사용한다. 이와같은 구조적 특성을 이용하여 임의의 악성 정보를 MS-Word 파일 내에 은닉할 수 있다. 아래와 같이 메시지와 암호화 키를 준비한 후, 메시지를 암호화하여 암호문을 얻고 나서 암호문을 블록개수로 분할하여 XML 문서 내 태그에 속성값으로 추가/은닉할 수 있다[10].



[Fig. 4] Hidden Text Analysis Process

Zero Dimension Image, Property Coding 기반 Character Scale, Revision Identifier 등을 이용하여 디지털 파일 내에 악성 스크립트를 은닉할 수 있다. Zero Dimension Image 방식을 이용한 경우 해당 이미지 파일을 검색한 후 이미지의 파일 헤더 형태와 데이터 구조를 분석할 필요가 있다.

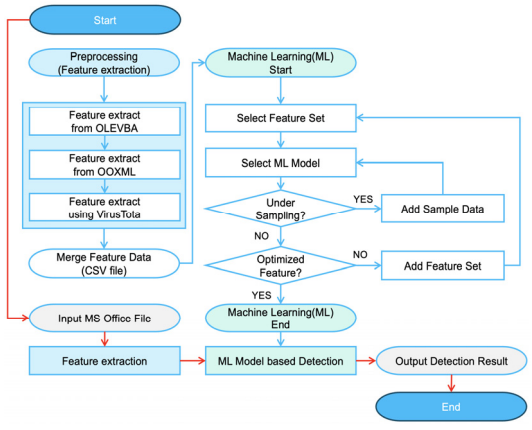
이밖에도 OLE VBA 객체 내에 악성 스크립트를 은닉할 수 있기 때문에 이에 대한 검출 메커니즘 개발이 필요하며[3], VirusTotal과 같은 시스템을 통해 대단위 샘플 데이터 셋을 기반으로 랜섬웨어와 같은 악성 스크립트 포함 유무를 검사할 필요가 있다.

3. 머신러닝 기반 손상된 디지털 파일 내부 은닉 악성 스크립트 판별 시스템

3.1 머신러닝 기반 악성 스크립트 판별 메커니즘

이에 본 연구에서 제시하는 머신러닝 기반 자동 판별 메커니즘을 제시하고자 한다. 임의의 MS Office 파일을 대상으로 정상/비정상 여부를 판별하기 위해서는 해당 디지털 파일의 내부 구조를 분석하여 특징 정보를 추출하고 이를 토대로 분석하는 과정을 수행할 수 있다. 즉, 특징 공학(feature engineering) 기법을 적용하여 MS Office 파일 내부 속성, 메타 정보와 특징 정보를 추출하고, 이를 토대로 최적의 머신러닝 모델[11,12,13,14]을 적용하여 자동화된 판단 시스템을 설계 및 구축[15]할 수 있다. 머신러닝 학습 과정을 토대로 만일 최적의 머신러닝 모델을 찾을 수 있다면 임의의 MS Office 파일에 대해 정상 파일과 악성 스크립트가 은닉된 손상 파일 여부를 정확하게 판단할 수 있다.

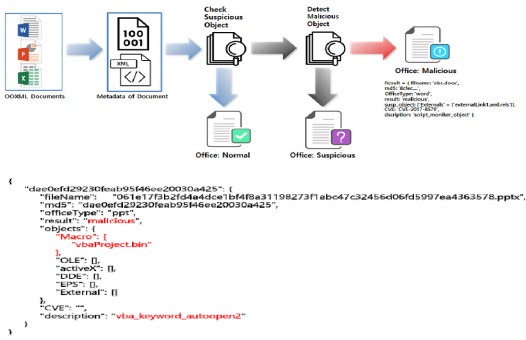
머신러닝에 사용될 특징 정보를 추출하기 위해서 본 연구에서는 (1) OLE VBA 기반 매크로 포함 여부 판별, (2) OOXML 구조 분석 기반 위변조 여부 판별, (3) VirusTotal[16] 기반 악성 스크립트 포함 여부 등 세 가지 판별 메커니즘 기반 특징 추출 방법을 사용하였다. 그리고 정상/비정상 여부를 자동 판별하기에 가장 적합한 머신러닝 모델을 학습토록 하였으며 결과적으로 최적의 성능을 제시하는 머신러닝 모델을 선택할 수 있었다.



[Fig. 5] Proposed ML based Detection Algorithm

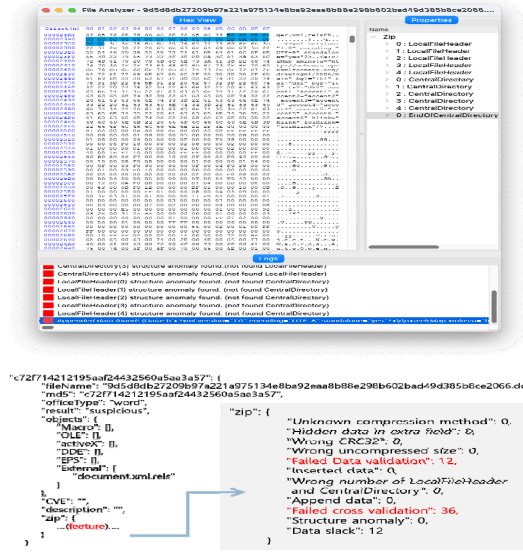
3.2 MS Office 계열 파일 내부 특징 정보 추출

위에서 제시한 알고리즘에서 OLE VBA 매크로 내에 악성 스크립트 포함 여부를 분석하고 이를 특징 정보로 추출하는 과정은 다음 그림과 같다. OOXML 구조 기반 디지털 파일에 대해 OLE VBA 중심 메타 데이터를 분석하고 내부에 의심이 되는 객체를 대상으로 특징 정보를 JSON 파일로 생성할 수 있다.



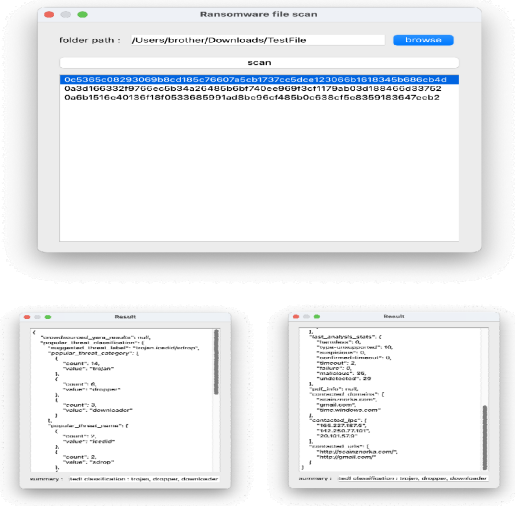
[Fig. 6] OLEVBA Analysis and Feature Extraction

또한 머신러닝 과정을 수행하기 위해서 그림 7과 같이 OOXML 기반 MS Office 파일에 대한 특징 정보를 추출하였다. 이와 같은 전처리 과정을 통해 OOXML 구조에 대한 분석 과정을 수행하여 디지털 파일 내부 구조적 문제점을 분석하고 비정상적 CDH, LFH 및 ECDR 구조 여부를 분석하고 이를 특징 정보로 생성하도록 구현하였다. 악성 스크립트가 포함된 MS Word 파일을 분석하였을 경우 그림 7과 같이 OOXML 구조내 비정상적인 CDH, LFH 헤더 정보를 검출 할 수 있었으며, ECDR 헤더 뒷 부분에는 악성 스크립트가 포함되어 있는 것을 확인할 수 있었다. 따라서 분석 결과를 JSON 포맷 또는 CSV 파일 형태로 변환해서 머신러닝 과정에 학습 데이터로 생성할 수 있다.



[Fig. 7] OOXML Analysis and Feature Extraction

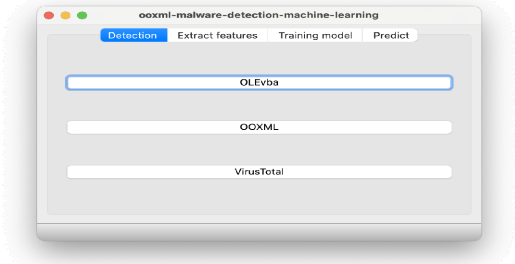
또한 그림 8과 같이 임의의 MS Office 계열 파일이 저장된 폴더를 선택하면 자동적으로 VirusTotal 사이트에 접속하여 해당 파일을 업로드/전송한 후, VirusTotal 사이트로부터 분석 결과를 JSON 파일 형태로 수신할 수 있다. 따라서 수신된 분석 결과를 다시 CSV 파일 형태로 변환하여 머신러닝 학습 과정에 적용할 수 있다.



[Fig. 8] VirusTotal based Analysis and Feature Extraction

3.3 머신러닝 기반 시스템 설계 및 구현

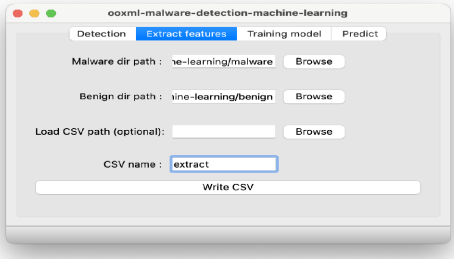
앞서 제시한 내용을 토대로 본 연구에서는 그림 9와 같은 머신러닝 기반 손상된 디지털 파일 자동 판별 시스템을 설계 및 구현하였다. Python 3.11 기반으로 PyQt5 라이브러리와 OLETools.olevba 패키지 내 VBA_Parser, VT 패키지를 이용하여 개발하였다.



[Fig. 9] Malicious Script Detection Function

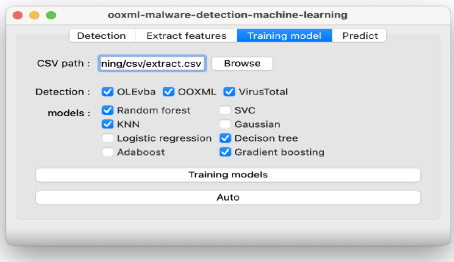
그림과 같이 OLEVBA, OOXML 그리고 VirusTotal 기반 MS Office 계열 파일 내 악성 스크립트 포함 여부를 개별적으로도 검사하고 분석 결과를 JSON 파일로 생성할 수 있도록 구현하였다.

그리고 그림 10과 같이 악성 데이터가 포함된 폴더와 정상 데이터가 포함된 폴더를 지정하여 특징 추출 과정을 수행하고 이를 CSV 파일로 통합하는 기능을 구현하였다.



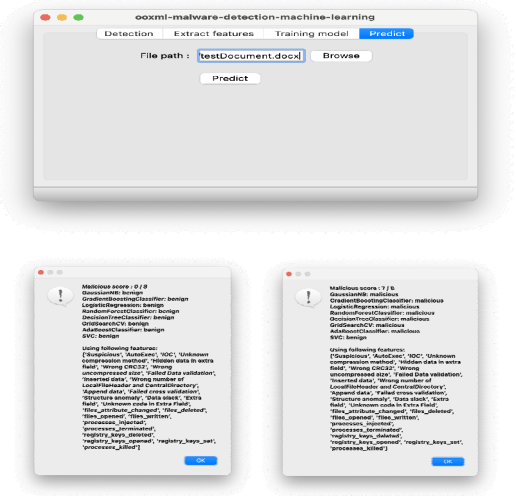
[Fig. 10] Feature Extraction & Integration Function

특징 정보가 통합된 CSV 파일을 이용하여 8개의 머신러닝 모델에 선택적으로 학습 과정을 진행할 수 있도록 구현하였으며 별도의 선택 없이 전체 머신러닝 모델을 자동적으로 적용할 수 있는 모드(Auto mode)로 구현하였다.



[Fig. 11] ML-based Model Traning Function

이제 학습 과정을 수행한 후에는 그림 12와 같이 분석하고자 하는 임의의 MS Office 계열 파일을 입력하여 머신러닝 기반 자동 판별 과정을 수행하도록 구현하였다. 그림에서 확인할 수 있듯이 특정 파일을 입력하면 학습된 8개의 머신러닝 모델을 적용하여 정상(benign)/악성(malware) 여부를 자동 판별할 수 있도록 구현하였다.



[Fig. 12] Benign/Malware Auto-Detection Function

본 연구에서 구현한 시스템은 8개의 머신러닝 모델을 이용하여 세 가지 형태의 악성 스크립트 검출 시스템을 통합하였다. 개별적으로 검출/분석 기능을 수행할 수도 있을 뿐만아니라, 선택적으로 머신러닝 기법을 적용하여 MS Office 계열 파일(워드, 파워포인트, 엑셀) 각각의 특성에 따라 유용한 분석 과정을 수행하도록 설계 및 구현하였다.

4. 실험 결과 분석

4.1 실험 환경 및 머신러닝 모델

MS Office 계열의 디지털 파일에 대한 손상 여부 및 악성 스크립트 은닉 여부를 자동 판별하기 위해 머신러닝 기반 학습 과정은 다음 〈Table 1〉과 같은 실험 환경에서 실행하였다. 운영체제는 Ubuntu Server를 기반으로 하였으며 시스템 내부에 가상 머신을 설치하여 파이썬 프로그래밍을 할 수 있는 통합개발환경, 그리고 파이썬 기반으로 MS Office 파일 내부 OLE VBA 매크로 정보 유무를 검출 할 수 있는 모듈과 OOXML 기반 구조분석 과정을 수행할 수 있는 모듈 및 외부 VirusTotal 시스템을 이용하여 정상/악성 여부를 확인할 수 있는 모듈을 결합하여 학습 과정을 진행하였다. 머신러닝 모델을 구현하기 위한 scikit-learn 라이브러리를 이용하여 각각의 머신러닝 모델별로 판별 성능을 비교 분석하였다.

〈Table 1〉 Machine Learning Environment

Environment	Description
CPU	Intel® Core™ i7-8550U CPU @ 1.80GHz × 2
RAM	15GB
OS	Ubuntu 22.04 LTS
Python 3.11	interactive programming language [17]
scikit-learn 1.2.2	Python-based machine learning library [18]

머신러닝 과정에 필요한 샘플 파일(MS Office 계열 파일)을 확대해 나가면서 판별 성능을 비교하였고, 정상/비정상 자동 분류 과정에 사용 가능한 특징 정보의 대상과 목록을 달리하면서 각각의 머신러닝 모델별로 정확도(accuracy)와 F1 스코어를 비교하여 최종적으로 비정상 MS Office 파일을 대상으로 자동 판별 성능이 우수한 최적의 머신러닝 모델을 선정하였다. 자동 판별 과정에 사용 가능한 머신러닝 모델을 선정하기 위해 〈Table 2〉와 같이 총 8개의 머신러닝 모델과 옵션을 설정하여 성능을 비교 분석하였다.

〈Table 2〉 List of Machine Learning Models Used

No.	Model	Options
1	Random Forest	N estimators=50. criterion="entropy"
2	SVC	-
3	k-NN	n neighbors=5~30
4	Gaussian Naïve Bayes	-
5	Logistic Regression	Max iter=400
6	Decision Tree	-
7	Ada Boost	N estimators=100
8	Gradient Boosting	N estimators=100, learning rate=1

4.2 성능 비교 평가 기준 및 지표

머신러닝 기법을 적용하였을 경우 〈Table 3〉과 같이 네 가지 형태의 판단 행렬(decision matrix)을 도출할 수 있으며, 이를 토대로 각각의 머신러닝 모델에 대한 자동 판별 성능을 비교 평가할 수 있다.

〈Table 3〉 Decision Matrix

	Predicted Malware : Yes	Predicted Malware : No
Malware	True Positive(TP)	False Negative(FN)
Normal	False Positive(FP)	True Negative(TN)

머신러닝 모델의 성능을 비교/평가하는 지표로는 정확도(Accuracy), 정밀도(Precision), 재현율(Recall), F1 score(F-Score or F-measure) 등이 있다. 이 중에서 F1 score는 정밀도와 재현율을 모두 반영하는 평가지표로 머신러닝 모델의 성능 평가지표로 가장 많이 사용하고 있다.

$$Accuracy = \frac{(TP + TN)}{100} \quad (1)$$

$$\text{Precision} = \frac{TP}{(TP + FP)} \quad (2)$$

$$Recall = \frac{TP}{(TP + FN)} \quad (3)$$

$$F1\,score = \frac{2\,(\text{Precision} * Recall)}{(\text{Precision} + Recall)} \quad (4)$$

따라서, MS Office 계열의 디지털 파일에 대한 위변조 및 악성 스크립트 포함 여부를 판별하는 성능을 비교하기 위해 〈Table 3〉에 제시된 8가지 머신러닝 모델을 대상으로 각 머신러닝 모델별로 정확도와 F1 스코어를 측정하여 최적의 머신러닝 모델을 도출/선별하는 과정을 수행하였다.

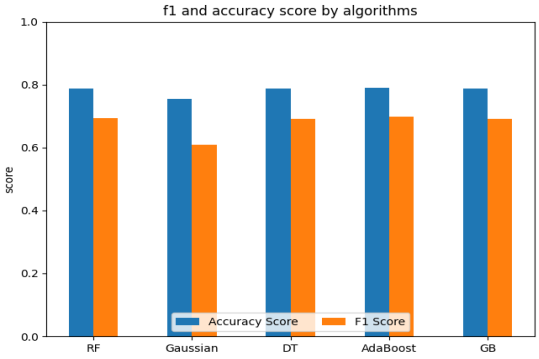
실험 데이터로는 MalwareBazzar, DigitalCorpora 사이트에서 오픈한 MS Office 계열 데이터셋을 사용하였다. 정상 파일 150개와 악성 스크립트를 포함한 비정상 데이터 파일 169개 등 총 319개의 MS Office 파일을 대상으로 실험을 수행하였다. 머신러닝 과정에 사용한 총 26개의 특징 정보의 반영 비율은 그림 13과 같이 각각의 가중치가 자동 산정되었음을 확인할 수 있었다.


```
{
  "Wrong CRC32": 0.0,
  "Wrong uncompressed size": 0.0,
  "Wrong number of LocalFileHeader and CentralDirectory": 0.0001,
  "Unknown compression method": 0.0003,
  "AutoExec": 0.0005,
  "Append data": 0.0019,
  "Failed Data validation": 0.0023,
  "Inserted data": 0.003,
  "Hidden data in extra field": 0.0055,
  "IOC": 0.0069,
  "processes_killed": 0.0073,
  "Unknown code in Extra Field": 0.02798,
  "processes_injected": 0.0352,
  "Failed cross validation": 0.037,
  "Suspicious": 0.0562,
  "Data slack": 0.0604,
  "Extra field": 0.0615,
  "files_deleted": 0.0754,
  "Structure anomaly": 0.0908,
  "registry_keys_deleted": 0.1014,
  "files_attribute_changed": 0.1173,
  "registry_keys_opened": 0.6614,
  "files_opened": 0.6766,
  "registry_keys_set": 1.108,
  "files_written": 1.2349,
  "processes_terminated": 1.4071,
}
```


[Fig. 13] Estimated Feature's Weight Value

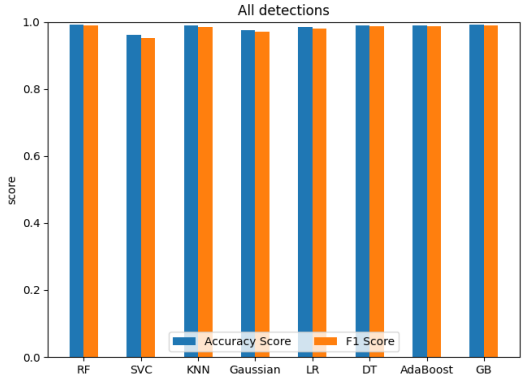
4.3 성능 평가 결과 분석

본 연구에서 사용한 세 가지 방법을 통합하여 적용하지 않고 개별적인 방식으로 각각 머신러닝 기법을 적용할 경우 그림 14와 같이 정확도 및 F1 스코어가 자동 측정되도록 구현하였다.



[Fig. 14] ML Model based Detection Results (1)

하지만, 정상/비정상이 포함된 총 100개의 MS Office 파일을 대상으로 통합적인 형태로 자동 판별 과정을 실험한 결과 그림 15와 같이 랜덤 포레스트와 로지스틱 회귀 머신러닝 모델이 대체로 높은 정확도(96.67%)로 정상/악성 여부를 자동 판별하였으며, F1 스코어를 측정해 보면 로지스틱 회귀 모델(95.76%)이 다른 모델에 비해 상대적으로 높은 판별 성능을 나타냈다. 성능 평가 결과 SVC가 가장 낮은 성능을 보이는 것을 확인할 수 있었고 회귀 모델, 랜덤 포레스트 등이 대체적으로 우수한 판별 성능을 나타냄을 확인할 수 있었다.



[Fig. 15] ML Model based Detection Results (2)

5. 결론

본 연구에서는 OOXML 구조 기반 MS Office 계열 파일과 같이 사용자가 급증하고 있는 디지털 파일 내에 악성 스크립트 등을 은닉하는 등의 디지털 파일을 손상/위변조하거나, 적법 형태로 위장한 악성 파일에 대해 머신러닝 기법을 적용하여 자동으로 검출할 수 있는 시스템을 설계 및 구현하였다. MS Office 파일 내 OLE VBA 매크로 기능을 악용하여 악성 스크립트를 포함한 경우에 대한 검출, OOXML 구조 분석을 통해 CDH/LFH/ECDH 내부 필드 값에 데이터를 은닉하거나 OOXML 구조 내에서 참조되지 않는 비정상적인 CDH/LFH 정보를 추가하여 악성 스크립트를 은닉하는 경우에 대한 검출 그리고 VirusTotal과 같은 공인된 악성 스크립트 판별 기능을 접목하여 MS Office 파일에 대한 악의적 손상 및 위변조 여부를 자동 판별할 수 있는 메커니즘을 제시하고 소프트웨어를 설계 및 구현하였다. 실험 결과 디지털 파일 손상 여부를 판별할 수 있는 최적의 머신러닝 모델을 도출할 수 있었으며 임의의 MS Office 파일에 대해 95.76%의 확률로 손상된 악성 디지털 파일을 검출할 수 있다는 것을 확인할 수 있었다. 본 연구에서 제시한 메커니즘 및 시스템을 더욱 발전시켜 향후 대단위 네트워크 환경에서 악성 스크립트 기반 디지털 파일 기반 사이버 침해공격에 능동적으로 대응할 수 있는 시스템을 연구할 필요가 있다.

REFERENCES

[1] J. Paoli, I. Valet-Harper, A. Farquhar, and I. Sebestyen, “Ecma-376 office open xml file formats,” URL https://ecma-international.org/publications-and-standards/standards/ecma-376/

[2] A. M. Naser, M. H. Btoush, and A. H. Hadi, “Analyzing and detecting malicious content: Docxfiles,” International Journal of Computer Science and Information Security, Vol.14, No.8, pp.404, 2016.

[3] P. Singh, “Detection of Malicious OOXML Documents Using Domain Specific Features”, Master’s thesis, Indian Institute of Information Technology and Management, 2017.

[4] Z. Wang, J. Wang, “Applications of Machine Learning in Public Security Informatin and Resource Management”, Hindawi Scientific Programming, Vol.2021, Article ID 4734187, 2021.

[5] Y. Liu, F. R. Yu, X. Li, H. Ji, and V. C. M. Leung, "Blockchain and machine learning for communications and networking systems," IEEE Communications Surveys & Tutorials, Vol.22, No.2, pp.1392-1431, 2020.

[6] R. Gupta, S. Tanwar, S. Tyagi, and N. Kumar, "Machine learning models for secure data analytics: a taxonomy and threat model," Computer Communications, Vol.153, pp.406-440, 2020.

[7] A. Cohen, N. Nissim, L. Rokach, and Y. Elovici, “Sfem: Structural feature extraction methodology for the detection of malicious office documents using machine learning methods,” ExpertSystems with Applications, Vol.63, pp. 324–343, 2016.

[8] H. S. Lee, H.-W. Lee, "Forgery Detection Mechanism with Abnormal Structure Analysis on Office Open XML based MS-Word File," IJASC, Vol.8, No.4, 2019.

[9] S. Na and H.-W. Lee, "Implementation of Malicious Data Analysis and Detection System Hidden in the Slack Space of Corrupted OOXML-based MS-Office Digital Files", Advanced and Applied Convergence Letters AACL 21 (9th International Joint Conference on Convergence, IJCC2023), pp.97-103, 2023.

[10] A. Catsiglione, B. D’Alessio, A. D. Santis, “Hiding Information into OOXML Documents: New Steganographic Perspectives”, Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications, Vol.2, No.4, pp.59-83, 2011.

[11] S. D. l. Santos and J. Torres, “Macro malware detection using machine learning techniques - a new approach,” in Proceedings of the 3rd International Conference on Information SystemsSecurity and Privacy - Volume 1: ICISSP, INSTICC. SciTePress, pp.295–302, 2017.

[12] S. Kim, S. Hong, J. Oh, and H. Lee, “Obfuscated vba macro detection using machine learning,” in 2018 48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), pp.490–501, June 2018.

[13] FireEye, "Malicious PowerShell Detection via Machine Learning," 2018.

[14] B. Mahesh, “Machine Learning Algorithms – A Review”, International Journal of Science and Research, Vol.9, No.1, 2020.

[15] W. Richert, L. P. Coelho, "Building Machine Learning Systems with Python", Packt Publishing Ltd., ISBN 978-1-78216-140-0.

[16] VirusTotal, https://www.virustotal.com/.

[17] Python. https://www.python.org/.

[18] scikit-learn. https://scikit-learn.org/.

이 형 우(Hyung-Woo Lee) [종신회원]

- 1994년 2월 : 고려대학교 컴퓨터학과 (학사)
- 1996년 2월 : 고려대학교 컴퓨터학과 (석사)
- 1999년 2월 : 고려대학교 컴퓨터학과 (박사)
- 1999년 3월 ~ 2003년 2월 : 백석대학교 정보통신학부 교수
- 2003년 3월 ~ 현재 : 한신대학교 컴퓨터공학부 교수

〈관심분야〉
사물인터넷, 정보보호, 모바일 보안 및 디지털 포렌식, 지능형 사이버공격 대응

나 상 원(Sangwon Na) [준회원]

- 2015년 3월 ~ 2018년 2월 : 청명고등학교
- 2018년 3월 ~ 현재 : 한신대학교 컴퓨터공학부 학생(학부과정)

〈관심분야〉
정보보호, 머신러닝