산업융합연구 제21권 제12호 pp. 55-62, 2023
ISSN 2635-8875 / e-ISSN 2672-0124
DOI : https://doi.org/10.22678/JIC.2023.21.12.055

랜섬웨어 탐지를 위한 머신러닝 기반 암호화 행위 감지 기법

황윤철*
한남대학교 탈메이지교양 · 융합대학 교수

A Machine Learning-Based Encryption Behavior Cognitive Technique for Ransomware Detection

Yoon-Cheol Hwang*
Professor, Department of Talmage Liberal Arts · Convergence College, Hannam University

요 약 최근 등장하는 랜섬웨어들은 다양한 공격 기법과 다양한 경로를 통해 공격을 수행하고 있어 조기 탐지와 방어에 많은 어려움을 겪고 있으며, 그 피해 규모도 날로 증가하고 있다. 따라서 본 논문에서는 효과적인 랜섬웨어 탐지를 위하여 파일 암호화와 암호화 패턴을 머신러닝 기반으로 하는 감지 기법을 제안한다. 파일 암호화는 랜섬웨어가 공격하는데 필수적으로 사용하는 기능으로 암호 행위와 암호화 패턴을 분석함으로써 랜섬웨어를 탐지하고 랜섬웨어의 특정 변종이나 새로운 유형의 랜섬웨어를 탐지할 수 있기 때문에 랜섬웨어 공격을 식별하고 차단하는 데 매우 효과적이다. 제안한 머신러닝 기반의 암호화 행위 감지 기법은 암호화 특성과 암호화 패턴 특성을 추출하여 머신러닝 기반의 분류기를 통해 각각 학습을 시켜 해당 행위에 대한 탐지를 진행하고 최종 결과는 두 분류기의 평가 결과를 기반으로 앙상블 분류기에서 랜섬웨어 유무를 판별하여 좀 더 정확도를 높였다. 또한, 제안한 기법을 numpy와 pandas, 파이썬의 사이킷런 라이브러리를 사용하여 구현하여 평가지표를 사용한 성능를 평가한 결과 평균적으로 94%,의 정확도와 95%의 정밀도, 93%의 재현률과 95%의 F1 스코어가 산출되었다. 성능 평가 결과를 보면 암호화 행위 감지를 통해 랜섬웨어 탐지가 가능하다는 것을 확인할 수 있었고 랜섬웨어의 사전 탐지를 위해 제안한 기법의 성능을 높이기 위한 연구도 계속해서 진행되어야 한다.

키워드 : 랜섬웨어, 탐지, 머신러닝, 암호화 행위, 감지 기법

Abstract Recent ransomware attacks employ various techniques and pathways, posing significant challenges in early detection and defense. Consequently, the scale of damage is continually growing. This paper introduces a machine learning-based approach for effective ransomware detection by focusing on file encryption and encryption patterns, which are pivotal functionalities utilized by ransomware. Ransomware is identified by analyzing password behavior and encryption patterns, making it possible to detect specific ransomware variants and new types of ransomware, thereby mitigating ransomware attacks effectively. The proposed machine learning-based encryption behavior detection technique extracts encryption and encryption pattern characteristics and trains them using a machine learning classifier. The final outcome is an ensemble of results from two classifiers. The classifier plays a key role in determining the presence or absence of ransomware, leading to enhanced accuracy. The proposed technique is implemented using the numpy, pandas, and Python's Scikit-Learn library. Evaluation indicators reveal an average accuracy of 94%, precision of 95%, recall rate of 93%, and an F1 score of 95%. These performance results validate the feasibility of ransomware detection through encryption behavior analysis, and further research is encouraged to enhance the technique for proactive ransomware detection.

Key Words : Ransomware, Detection, Machinelearnning, Encryption behavior, Cognitive technique

This work was supported by 2023 Hannam University Research Fund.
*Corresponding Author : Yoon-Cheol Hwang(dolpin2010@gmail.com)
Received October 6, 2023 Revised November 8, 2023
Accepted December 20, 2023 Published December 28, 2023

1. 서론

현재 급속하게 발전되고 있는 통신기기와 통신 기술은 우리의 삶에 편리성과 신속성을 제공하고 있지만 이를 이용한 악의적인 행동으로 우리의 일상을 파괴되는 일이 흔하게 발생되고 있는 실정이다[1]. 악의적인 행동 중 하나인 랜섬웨어 공격은 최근에 지속적으로 증가하고 있으며, 사이버 범죄자들이 더욱 지능적이고 정교한 기술을 사용하여 새로운 형태의 랜섬웨어를 개발하고 있다. 랜섬웨어는 사이버 범죄자들에 의해 개발되고 확산되는 악성 소프트웨어로, 기기나 데이터에 접근하거나 사용자의 파일을 암호화하여 데이터를 블로킹하고, 복원하기 위해 일정 금액의 금전 보상을 요구하는 공격이다[2]. 최근 랜섬웨어 공격자들은 초기 침투부터 피해자 확보, 탐지 우회에 다양한 수단을 사용하여 전략적으로 공격을 수행하고 있으며 공격 대상도 금융 기관, 의료 기관, 정부 기관, 학교 및 대학, 에너지 회사, 소매업체 등과 같은 다양한 산업과 조직을 겨냥하고 있다. 이러한 산업과 기관들은 중요한 고객 데이터, 재무 정보, 기밀 연구 등을 보유하고 있어서 랜섬웨어 공격에 의한 피해가 크고 심각하다. 그리고 최근 랜섬웨어들은 따라서 랜섬웨어를 신속하고 정확하게 공격 초기에 탐지하고 대응하는 기술이 필요하여 현재 여러 연구기관에서도 꾸준히 연구를 진행하고 있다[3,4].

암호화는 랜섬웨어가 가장 흔히 사용하는 기능 중 하나이고, 암호화 패턴을 분석하면 랜섬웨어의 특정 변종이나 새로운 유형의 랜섬웨어를 탐지할 수 있으므로 랜섬웨어 공격 초기에 랜섬웨어를 식별하고 차단하는 것에 암호화 특성과 암호화 패턴을 분석하고이용하는 것은 매우 효과적이다[5-8]. 따라서, 본 논문에서는 랜섬웨어를 조기에 탐지하기 위해 랜섬웨어가 목표 시스템에서 반드시 행하는 행동인 암호화 과정에 초점을 맞추어 암호화 특성과 암호화 패턴을 기초 데이터로 사용하여 머신러닝 기반으로 랜섬웨어를 탐지하는 기법을 제안하고 구현한다.

본 논문에서 제안한 머신러닝 기반의 암호화 행위 감지 기법은 암호화 특성과 암호화 패턴 특성을 추출하여 머신러닝 기반의 분류기를 통해 각각 학습을 시켜 해당 행위에 대한 탐지를 진행하고 최종 결과는 두 분류기의 평가 결과를 기반으로 앙상블 분류기에서 랜섬웨어 유무를 판별하게 하여 좀 더 정확도를 높이는데 주안점을 두었다. 제안한 암호화 행위 감지 기법은 numpy와 pandas, 파이썬의 사이킷런 라이브러를 사용하여 구현하였고, 평가지표를 사용한 성능 평가 결과 평균적으로 94%,의 정확도와 95%의 정밀도, 93%의 재현률과 95%의 F1 스코어가 산출되었다. 평가 결과를 보면, 암호화 행위 감지를 통해 랜섬웨어 탐지가 가능하다는 것을 확인할 수 있었다.

논문의 구성은 다음과 같다. 2장에서는 랜섬웨어공격의 암호화 과정과 암호화 기법들을 관련 연구로 살펴보고, 3장에서는 암호화 행위와 암호화 패턴을 감지하여 랜섬웨어를 탐지하는 머신러닝 기반의 탐지 기법을 제안한 다음, 4장에서는 제안된 기법을 numpy와 pandas, 파이썬의 사이킷런 라이브러를 사용하여 구현하고 평가지표를 통해 랜섬웨어의 탐지 성능을 평가한다. 그리고 5장에서 연구에 대한 평가와 향후 연구과제로 끝을 맺는다

2. 관련연구

랜섬웨어는 악성 소프트웨어의 한 유형으로, 사용자의 파일을 암호화하여 접근을 차단하고 해독키 또는 복호화 키를 제공하면서 금전적 보상을 요구하는 공격을 수행한다. 랜섬웨어는 일반적으로 불법적으로 원치 않는 소프트웨어가 설치되거나 피싱 이메일 첨부 파일과 같은 보안 취약점을 통해 목표 시스템에 침투한다. 랜섬웨어가 목표 시스템에 있는 파일을 암호화하는 과정은 침투, 실행 및 암호화, 암호화 키 생성, 해독키 저장 및 전송, 금전적 보상 요구와 같이 5단계로 진행한다[9,10].

침투단계에서는 주로 스팸 메일, 악의적인 다운로드 링크, 취약점을 이용한 공격 등 다양한 방법으로 목표 시스템에 침투한다. 실행 및 암호화 단계에서는 랜섬웨어가 목표 시스템에 침투하여 악성코드를 실행한다. 이 코드는 사용자의 중요한 파일을 검색하고 이러한 파일을 암호화하고, 파일 확장자나 내용을 변경하여 액세스가 불가능하게 만든다. 암호화 키 생성단계에서 랜섬웨어는 대개 암호화 라이브러리를 사용하여 파일을 암호화하는데 필요한 키를 생성하고, 생성된 키는 공개키 암호화 방식을 사용하여 암호화한다. 해독 키 저장 및 전송 단계에서 랜섬웨어는 생성된 해독 키를 원격 서버에 저장하고, 피해자에게 금전적 보상을 요구하는 메시지와 함께 이 키를 전송한다. 해독 키가 피해자에게 제공되지 않으면 암호화된 파일들은 거의 복원 불가능하다. 마지막으로 금전적 보상 요구 단계에서 랜섬웨어는 해독 키를 받기 위해 피해자로부터 비트코인 또는 다른 암호화폐를 요구하고, 이 보상

을 지불하면 해독 키를 제공한다고 사용자에게 알린다. 일부 랜섬웨어는 제한된 시간 내에 보상을 요구하여 압박감을 조장하기도 한다[11].

최근 랜섬웨어들의 암호화 방법을 살펴보면, RSA, AES, Salsa20, ChaCha 등과 같은 고급 암호화 알고리즘을 사용하여 키를 생성하고 파일을 암호화하고, 해독과 파일 복원을 더욱 어렵게 하기 위해 여러 개의 암호화 키를 사용하여 파일을 다단계로 암호화하는 방법을 사용한다. 그리고, 랜섬웨어를 사전에 탐지하는 것을 방지하기 위해 인터넷에 연결되지 않은 환경에서도 암호화를 수행하는 오프라인 암호화 방법과 특정 산업이나 조직을 표적으로 정하는 타겟 지정 암호화 방법을 사용하기도 한다. 일부 랜섬웨어는 파일 암호화뿐만 아니라 네트워크와 연결된 다른 디지털 기기를 암호화하거나 다른 시스템을 감염시키는 기능을 사용하기도 한다[12].

Table 1. Ransomware by encryption type

Encryption Type	A Typical Ransomware
RSA encryption	CryptoLocker,TeslaCryptCryptoWall
AES encryption	Cerber,Locky,CryptoMix Petya/NotPetya
Salsa20 & ChaCha encryption	Ryuk, STOP/Djvu GandCrab
hybrid encryption	Dharma/CrySis,MazeSnatch, Sodinokibi/REvil

Table 1과 같이 암호화 유형별로 랜섬웨어들은 다양한 암호화 방법과 기술을 사용하여 파일을 암호화하기 때문에 파일 복구가 점점 어려워지고 피해도 점점 커지고 있다.

암호화 패턴은 랜섬웨어와 같은 악성 소프트웨어의 특정 변종이나 같은 유형의 랜섬웨어에서 사용되는 암호화 기술과 패턴을 분석하는 것을 의미한다. 이를 통해 랜섬웨어의 동작 방식을 이해하고, 더 나아가 새로운 랜섬웨어 변종에 대응할 수 있는 보안 솔루션 개발에 사용한다. 암호화 패턴에서 분석하는 내용은 암호화 알고리즘과 키와 파일 암호화 방법, 암호화된 파일의 특징과 변종 간의 차이점이다. 랜섬웨어가 사용하는 암호화 알고리즘과 키를 분석하여 랜섬웨어가 사용하는 암호화 방식을 파악하고, 암호화된 파일을 복구하기 위한 키를 찾는다. 그리고 랜섬웨어가 파일을 어떻게 암호화하는지, 암호화된 파일의 특징 및 패턴을 분석하여 랜섬웨어의 파일 암호화 특성을 이해하고 랜섬웨어를 식별하는데 활용한다. 또한, 랜섬웨어에 의해 암호화된 파일은 특정한 패턴이나 특징을 가지기 때문에 이러한 특징은 랜섬웨어와 정상 파일을 구분하는데 사용되고, 랜섬웨어의 변종은 암호화 패턴과 동작 방식에서 기존의 랜섬웨어와 다소 차이가 있을 수 있다. 따라서 다양한 랜섬웨어 변종 간의 차이점을 분석하면 새로운 변종에 대응하는 기술을 개발할 수 있다[13].

3. 머신러닝 기반 암호화 행위 감지 기법

제안한 머신러닝 기반 암호화 행위 감지 기법은 암호화 특성 추출, 암호화 행동 패턴 특성 추출, 데이터 전처리, 암호화 특성 학습, 암호화 패턴 학습, 랜섬웨어 판별 과정을 통해서 진행되고, 이를 도식화하면 Fig. 1과 같다. 단계별로 진행되는 세부 내용을 살펴보면 다음과 같다.

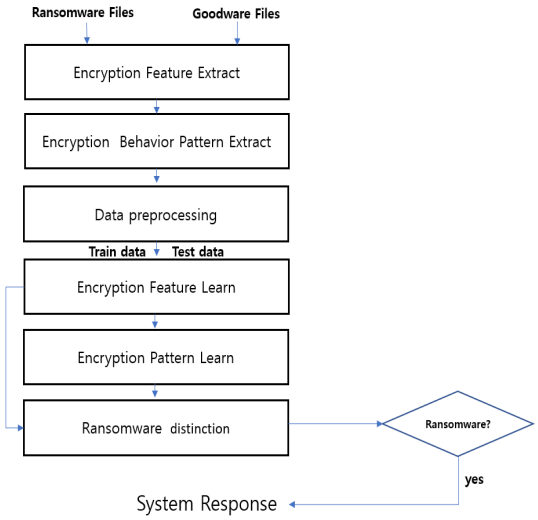

Fig. 1. Encryption behavior cognitive process

3.1 암호화 특성 추출

파일의 바이너리 코드를 분석하여 특정 헤더 정보를 추출하여 파일 이름, 경로, 크기, 생성 날짜, 수정 날짜, 액세스 날짜 등과 같은 속성을 추출한다. 랜섬웨어 파일은 일반적으로 특정 크기 범위에 속할 수 있으므로 파일 크기를 추출하고 파일이 특정 암호화 알고리즘을 사용하여 암호화 되었는지 여부를 확인하기 위해 암호화 알고리즘과 관련된 특성을 추출하여 파일 내에서 특정 패턴이나 키 생성 알고리즘과 관련된 특성을 찾아낸다. 본 기법에서 [14]에서 사용한 특성 추출과 분류 방법을 사용하여 랜섬노트, 대상 파일 암호화, 암호화 키 교환, 파일 형식 변경, 암호화 행위 감지, 암호화 시간과 관련된 특성들을 랜

섬웨어의 암호화 특성으로 분류하여 사용한다.

3.2 암호화 행동 패턴 특성 추출

랜섬웨어를 식별할 수 있는 암호화 패턴의 특성을 추출한다. 암호화 패턴의 특성인 암호화 알고리즘 식별, 랜섬노트 특성, 암호화 키 교환 특성, 파일 시그니처 변조, 암호화 대상 파일 특성, 암호화된 파일 패턴, 암호화 활동 등을 랜섬웨어의 암호화 행동 패턴 특성으로 분류하여 사용한다.

3.3 데이터 전처리

랜섬웨어와 정상 파일에서 추출한 특성 데이터들을 대상으로 불필요한 데이터를 정제하고 각 특성 데이터를 일관된 형식으로 변환한다. 데이터 전처리 단계에서는 데이터를 학습 데이터와 테스트 데이터로 분할하고, 데이터 값의 범위를 정규화 하고 스케일링을 진행한 후 데이터를 압축하고 계산의 효율성을 높이기 위해 PCA(Principal Component Analysis)를 사용하여 데이터의 차원 축소가 진행된다.

3.4 암호화 특성 학습

암호화 특성 추출 단계에서 추출된 데이터를 입력으로 받아 암호화 특성 분류기에서 학습을 진행한 후, 해당 파일이 랜섬웨어인지 아닌지를 분류하는 역할을 암호화 특성 분류기에서 한다. 이 분류기는 머신러닝 알고리즘을 사용하여 암호화 특성들간의 패턴을 학습하고, 이를 기반으로 랜섬웨어와 일반 파일을 구분하는 모델을 구축하고 랜섬웨어를 탐지하기 위한 학습을 진행한다.

3.5 암호화 패턴 학습

암호화 특성 분류기에서 분류한 결과를 입력으로 받아, 랜섬웨어의 특정 암호화 패턴을 탐지하는데 사용되는 데이터를 준비하는 단계이다. 이 단계에서는 랜섬웨어의 특정 행위 또는 특징을 분석하여 랜섬웨어를 식별하는 머신러닝 분류 모델을 구축하고 랜섬웨어를 탐지하기 위한 학습을 암호화 패턴 분류기에서 진행한다. 이 분류기는 랜섬웨어의 특정 행위 또는 특징을 분석하여 랜섬웨어를 식별하는 역할을 한다.

3.6 랜섬웨어 판별

암호화 특성 분류기와 암호화 패턴 분류기에서의 결과를 종합하여 최종적으로 랜섬웨어인지 아닌지를 앙상블 분류기에서 결정한다. 이러한 앙상블 방법은 암호화 특성 분류기와 암호화 패턴 분류기의 예측 결과를 조합함으로써 랜섬웨어의 탐지 정확도를 향상시키고, 잘못된 판단을 최소화한다. 이 연구에서는 암호화 특성 분류기와 암호화 패턴 분류기에서 판별한 결과를 가지고 다수결 투표 기반으로 최종적으로 랜섬웨어인지 아닌지를 결정한다.

4. 머신러닝 기반 암호화 행위 감지 기법 구현 및 성능 평가

4.1 머신러닝 기반 암호화 행위 감지 기법 구현

본 논문에서 랜섬웨어를 효과적으로 탐지하기 위해 사용할 머신러닝 기반 암호화 행위 감지 시스템은 Fig. 2와 같고, 시스템은 특성 추출기와 전처리기, 암호화 특성 추출 분류기, 암호화 패턴 분류기, 앙상블 분류기로 이루어져 있다.

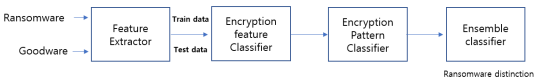

Fig. 2. Encryption behavior cognitive system

Fig. 2에서 특성 추출기는 암호화 행위와 암호화 패턴과 관련된 특성을 추출한다. 이 특성 추출기를 통해서 랜섬웨어와 정상 파일의 특성들이 추출된다. 전처리기에서는 추출된 암호화 특성들을 분류기에 입력으로 사용할 수 있는 형태로 전처리하며, 이 단계에서 특성들을 정규화하거나 벡터화하는 등의 작업이 이루어진다. 암호화 특성 분류기는 전처리기에서 정제되고 분할된 데이터를 입력으로 받아, 해당 파일이 랜섬웨어인지 아닌지를 분류하는 역할을 한다. 이 분류기는 머신러닝 알고리즘을 사용하여 암호화 특성들 간의 패턴을 학습하고, 이를 기반으로 랜섬웨어와 일반 파일을 구분하는 시스템을 구축한다. 암호화 패턴 분류기에서는 암호화 특성 분류기에서의 결과를 입력으로 받아 랜섬웨어의 특정 암호화 패턴을 탐지한다. 랜섬웨어의 특정 행위 또는 특징인 랜섬노트의 패턴, 암호화 알고리즘 패턴, 대상 파일 특성 등을 사용하여 머신러닝 분류 시스템을 구축한다. 앙상블 분류기에서는 암호화 특성 분류기와 암호화 패턴 분류기에서의 결과를 종합하여 최종적으로 랜섬웨어인지 아닌지를 결정한다. 암호화 특성 분류기와 암호화 패턴 분류기의 예측 결과를 조

합함으로써 정확도를 향상 시키고, 잘못된 판단을 최소화 할 수 있다.

제안한 암호화 행위 감지 시스템은 numpy와 pandas, 파이썬의 Scikit-Learn 라이브러리를 사용하여 윈도우 11 환경에서 구현하였다. 주요 부분의 구현 내용을 살펴보면 다음과 같다.

```
def extract_encryption_features(file_path):
    # Extract file size
    file_size = os.path.getsize(file_path)
    # Return the extracted features as a list
    features = [file_size]
    return features
def create_encryption_dataset(data_directory):
    # Distinguish ransomware files from normal files and
    # extract feature
    ransomware_samples = []
    normal_samples = []
    labels = np.array([1] * len(ransomware_samples) + [0] *
    len(normal_samples))
     data = np.vstack((ransomware_samples, normal_samples))
     # Split into training data and test data
     X_train, X_test, y_train, y_test = train_test_split(data,      labels,
test_size=0.2, random_state=42)
     return X_train, X_test, y_train, y_test
```

Fig. 3. Encryption feature extraction and preprocessing implementation code

암호화 특성 추출은 랜섬웨어와 일반 파일로부터 추출한 암호화 특성을 기반으로 분류를 위한 데이터를 준비한다. 이 특성들은 파일의 속성, 크기, 파일 시그니처, 암호화 키 교환 특성, 암호화 알고리즘 패턴 등을 포함하고, 추출한 암호화 특성들을 전처리하여 분류기의 입력으로 사용할 수 있는 형태로 가공하였다.

암호화 특성 중 파일 크기를 추출하는 것과 전처리에 대한 주요 코드는 Fig. 3과 같다. Fig. 3에서 extract_encryption_features() 함수는 해당 파일의 크기를 추출하여 리스트로 반환하기 위해 사용하였고, 랜섬웨어 파일과 일반 파일을 폴더 구조를 기반으로 구분하고 데이터셋을 만들기 위해 create_encryption _dataset() 함수를 사용하였다. ransomware_samples와 normal_samples는 각각 랜섬웨어 파일과 일반 파일의 특징을 저장하는 리스트이고, labels는 NumPy 배열로, 각 샘플들에 대한 레이블을 저장한다. sklearn.model_selection 모듈에서 가져와서 데이터와 레이블을 훈련 데이터와 테스트 데이터로 분할하기 위해 train_test_split 함수를 사용하였다.

암호화 특성 분류기는 암호화 특성 추출 단계에서 가공된 데이터를 사용하여 랜섬웨어와 일반 파일을 분류한다. 머신러닝 알고리즘을 사용하여 추출된 암호

화 특성들을 학습하고, 이를 기반으로 랜섬웨어와 일반 파일을 식별하는 분류 시스템을 구축하였다. 암호화 특성 분류기의 주요 코드는 Fig. 4와 같다.

```
# Create and train a encryption feature classifier
def train_encryption_classifier(X_train, y_train):
    classifier = RandomForestClassifier()
    classifier.fit(X_train, y_train)
# Create a dataset for training a encryption feature
# classifier with a dataset
data_directory = "."
X_train_encryption, X_test_encryption, y_train_encryption,
y_test_encryption = create_encryption_dataset(data_directory)
# Training a cencryption feature classifier
encryption_classifier=train_encryption_classifier(X_train_encryption,
y_train_encryption)
```

Fig. 4. Enctyption feature classifier implementation code

Fig. 4의 train_encryption_classifier() 함수에서는 RandomForestClassifier를 사용하여 랜덤 포레스트 기반 분류기를 만들고, fit() 메서드를 호출하여 훈련 데이터에 맞게 분류기를 학습한다..

암호화 패턴 분류기는 암호화 특성 분류기에서의 결과를 입력으로 받아 랜섬웨어의 특정 암호화 패턴을 탐지한다. 랜섬웨어의 특정 행위 또는 특징을 분석하여 랜섬웨어를 식별하는 랜섬노트의 패턴, 암호화 알고리즘 패턴, 대상 파일 특성 등을 활용하여 머신러닝 분류 시스템을 구축하였다. 암호화 패턴 분류기의 의 주요 코드는 Fig. 5와 같다.

```
# Create and train a encryption pattern classifier
def train_pattern_classifier(X_train, y_train):
    classifier = RandomForestClassifier()
    classifier.fit(X_train, y_train)
    return classifier
# Create a dataset for training a encryption feature
# classifier with a dataset
data_directory = "."
X_train_encryption,  X_test_encryption,  y_train_encryption,  y_test_
encryption = create_encryption_dataset(data_directory)
# Training a encryption pattern classifier
pattern_classifier  =  train_pattern_classifier(X_train_pattern,  y_train_
pattern)
```

Fig. 5. Encryption pattern classifier implementation code

Fig. 5에서 train_pattern_classifier() 함수는 Random

ForestClassifier를 사용하여 랜덤 포레스트 기반 분류기를 만들고, fit() 메서드를 호출하여 훈련 데이터에 맞게 분류기를 학습시킨다.

```
class EnsembleClassifier:
    def __init__(self, classifier1, classifier2):
        self.classifier1 = classifier1
        self.classifier2 = classifier2
    def predict(self, X):
        y_pred1 = self.classifier1.predict(X)
        y_pred2 = self.classifier2.predict(X)
     # Perform final predictions with ensemble methods
     # The results of the two classifiers are averaged to a
     # value between 0 and 1.
        y_pred_ensemble = (y_pred1 + y_pred2) / 2
     # Based on the threshold, 0.5 or higher is classified as
     # ransomware, and less than 0.5 as normal files.
        y_pred_ensemble = np.where(y_pred_ensemble >= 0.5, 1, 0)
        return y_pred_ensemble
# Create an ensemble classifier
ensemble_classifier = EnsembleClassifier(encryption_classifier,
pattern_classifier)
# Make predictions with test data
y_pred_ensemble = ensemble_classifier.predict(X_test_encryption)
```

Fig. 6. Ensemble classifier implementation code

앙상블 분류기는 앙상블 분류기는 암호화 특성 분류기와 암호화 패턴 분류기에서의 결과를 종합하여 최종적으로 랜섬웨어인지 아닌지를 결정하는 역할을 하게 구현하였다. 개별 분류기들의 예측 결과를 조합하여 좀 더 탐지의 정확도를 향상시키고, 잘못된 판단을 최소화하기 위해 앙상블 분류기를 사용하였다. 앙상블 분류기의 주요 코드는 Fig. 6과 같고, Fig. 6에서 EnsembleClassifier 클래스는 두 개의 분류기 암호화 특성 분류기과 암호화 패턴 분류기를 이용하여 앙상블 분류기를 구성한다. 그리고 이 클래스를 사용하여 앙상블 분류기를 생성하고, 테스트 데이터를 이용하여 예측을 수행한다.

시스템의 평가는 정확도, 정밀도, 재현율, f1 스코어와 같은 일반 평가 지표로 산출하였다. 시스템 평가 부분의 주요 코드는 Fig. 7과 같다.

Fig. 7의 구현 코드에서는 각각의 평가 지표를 각각의 변수에 저장하여 출력한다. 이상에서 살펴본 코드들은 모든 사항이 포함된 완벽한 구현 코드는 아니지만, 암호화 특성 추출과 사용 데이터의 전처리, 암호화 특성 분류기 생성 및 학습, 암호화 패턴 분류기 생성 및 학습, 앙상블 분류기의 생성과 랜섬웨어의 유무 판별 방법, 암호화 행위 감지 기법에 대한 평가 지표 등과 같이 3장에서 제안했던 암호화 행위 감지 기법의 구현 사항들을 확인할 수 있다.

```
# Evaluation metric calculation
accuracy = accuracy_score(y_test_encryption, y_pred_ensemble)
precision = precision_score(y_test_encryption, y_pred_ensemble)
recall = recall_score(y_test_encryption, y_pred_ensemble)
f1 = f1_score(y_test_encryption, y_pred_ensemble)
# Evaluation result output
print("Ensemble Classifier Accuracy: {:.2f}".format(accuracy))
print("Ensemble Classifier Precision: {:.2f}".format(precision))
print("Ensemble Classifier Recall: {:.2f}".format(recall))
print("Ensemble Classifier F1 Score: {:.2f}".format(f1))
```

Fig. 7. Evaluation metric implementation code

4.2 성능 평가

구현한 기법의 성능에 사용된 데이터는 로그 파일 유효성 검증 과정을 거친 후, 정상 프로그램 약 400개, 랜섬웨어 약 1000개를 사용하여, 평가 데이터 샘플의 개수를 다르게 하여 일반적인 평가 지표인 정확도와, 정밀도, 재현율, F1 스코어로 측정하였다[15,16]. 제안 기법의 성능 평가 결과는 Table 2와 같다.

Table 2. Evaluation results of proposal technique

support	accuracy	precision	recall	f1-score
600	0.92	0.91	0.94	0.92
800	0.94	0.93	0.94	0.93
1000	0.94	0.93	0.95	0.94
1200	0.95	0.94	0.95	0.94
1400	0.95	0.94	0.95	0.94
avg	0.94	0.93	0.95	0.93

Table 2에서 support는 평가에 사용한 샘플 데이터의 개수이고, 성능 평가 결과 통계치로 precision, recall, f1-score, accuracy 총 4개의 수치를 제시하였으며, 각 통계치마다 샘플 수에 따른 평균 수치를 산출하였다. 정확도(Accuracy)는 전체 샘플 중 올바르게 분류된 샘플의 비율을 나타내는 지표로 올바르게분류된 샘플 수를 전체 샘플 수로 나눈 값이다. 정밀도(Precision)는 랜섬웨어로 분류된 샘플 중에서 실제로 랜섬웨어인 샘플의 비율을 나타내는 지표로 실제 랜섬웨어로 분류된 랜섬웨어 샘플 수를 랜섬웨어로 분류된 샘플 수로 나누어 나온 값이고, 재현율(Recall)은 실제 랜섬웨어인 샘플 중에서 랜섬웨어로 정확하게 분류된 샘플의 비율을 나타내는 지표로 실제 랜섬웨어로 분류된 랜섬웨어 샘플 수를 실제 랜섬웨어 샘플 수로 나누어 나온 값이다. 그리고 F1 Score는 정밀도와 재현율의 조화 평균으로 계산되는 지표로, 정밀도와 재현

율을 곱한 값에 2를 곱해서 나온 값을 정밀도와 재현율을 더해 나온 값으로 나누어 나온 값으로 정밀도와 재현율의 균형을 나타내는데 사용한다.

Table 2에서 제안한 암호화 행위 감지 기법의 성능 평가 결과 통계치로 precision, recall, f1-score, accuracy 총 4개의 수치로 제시하였으며, 각 통계치마다 샘플 수에 따른 평균 수치를 산출하였다. 사용한 데이터 샘플의 수가 600개 일때는 정확도는 약 0.92, 정밀도는 약 0.91, 재현율은 약 0.94, F1 스코어는 약 0.92 이였고, 데이터의 샘플 수를 증가시켜 가면서 데이터 샘플의 수가 1400개 일때는 정확도는 약 0.95, 정밀도는 약 0.94, 재현율은 약 0.95, F1 스코어는 약 0.94로 측정되었다. 제안된 기법은 데이터 샘플을 늘려가면서 학습량을 늘려가면 평가 지표가 다소 향상되는 것을 Table 2를 통해 알 수 있고, 대부분의 랜섬웨어를 탐지할 수 있음을 확인할 수 있다.

5. 결론

현재 정보 통신 기술의 발달로 일반인들과 기업들이 처리하고 보유하게 되는 데이터의 양이 꾸준히 증가하고 있다. 이에 따라 디스크 내의 중요 데이터들을 암호화하고 복원을 원하는 사용자들에게 금품을 요구하는 랜섬웨어가 급증하고 있고 피해 또한 증가하고 있다. 랜섬웨어는 암호화 키가 없으면 복호화할 수 없기 때문에 공격을 당한 사용자는 요구하는 금품을 지불할 수밖에 없으므로 랜섬웨어를 사전에 탐지하여 랜섬웨어로 발생하는 개인과 기업의 피해를 줄일 필요가 있다.

본 논문은 다양한 랜섬웨어를 분석하여 랜섬웨어가 공격을 수행하면서 반드시 실행하는 암호화 행위를 기반으로 하는 랜섬웨어 탐지기법을 제안하였다. 암호화 행위에 관련된 암호화 특성과 암호화 패턴 특성을 도출하여 암호화 특성 분류기와 암호화 패턴 분류기를 머신러닝 기반으로 학습하여 탐지하는 기법을 사용하였고 두 분류기의 탐지 효과를 높이기 위해 앙상블 분류기를 통해 최종 랜섬웨어 유무를 판단하게 하였다.

제안된 암호화 행위 감지 기법을 numpy와 pandas, 파이썬의 사이킷런 라이브러리를 사용해 구현하였으며, 구현 결과 나온 평가지표로 일반적으로 사용하는 머신러닝기반 랜섬웨어 탐지기법과 대응하거나 약간 좋은 성능이 있음을 알 수 있었다. 또한, 이를 통해 랜섬웨어가 행하는 암호화 행위 감지로 랜섬웨어 탐지가 가능하다는 것도 확인하였다. 기존의 탐지 시스템은 기존에 존재하는 랜섬웨어는 대부분은 탐지하고 새로운 변종이나 새로운 유형의 랜섬웨어는 오탐률이 높다. 그러나 본 논문에서 제안된 기법을 탑재한 탐지 시스템은 랜섬웨어의 변종이나 새로운 유형의 랜섬웨어에 대해서도 발생 초기에 탐지와 대응이 가능하다. 현재보다 정확하고 효과적인 랜섬웨어를 탐지하는 시스템을 구축하기 위해 최근에 발생되고 있는 신종 랜섬웨어를 신속하게 발견하고 분석하여 새로운 암호화 특성을 추출한 후 기존의 탐지 시스템의 추론엔진에 추가하는 연구가 지속적으로 진행되어야 한다.

REFERENCES

[1] Mos, M. A., & Chowdhury, M. M. (2020, July). The growing influence of ransomware. In 2020 IEEE International Conference on Electro Information Technology (EIT) (pp. 643-647). IEEE. DOI : 10.1109/EIT48999.2020.9208254

[2] Y. C. Hwang. (2022). Extraction and classification of malicious code feature information for intelligent detection model. Industrial Convergence Research (formerly Journal of the Korean Society of Industrial Management), 20(5), 61-68. DOI : 10.22678/JIC.2022.20.5.061

[3] K. W. Moon, J. H. Lee. (2022). Recent Ransomware Trends and Development Direction. Journal of Information Security Society, 32(3), 33-39.

[4] Kok, S., Abdullah, A.,Jhanjhi,N.,& Supramaniam, M. (2019). Ransomware, threat and detection techniques: A review. Int. J. Comput. Sci. Netw. Secur, 19(2), 136.

[5] H. S. Kim,, & S. J. Lee. (2023). Comparative analysis of effective feature extraction techniques for machine learning-based ransomware attack detection. Convergence Security Thesis, 23(1), 117-123.

[6] Wan, Y. L., Chang, J. C., Chen, R. J., & Wang, S. J. (2018, April). Feature-selection-based ransomware detection with machine learning of data analysis. In 2018 3rd international conference on computer and communication systems (ICCCS) (pp. 85-88). IEEE.
DOI : 10.1109/CCOMS.2018.8463300

[7] H. G. Lee, J. H. Sung, Y. C. Kim, J. B. Kim, & K.

Y. Kim. (2017). A study on ransomware analysis and detection pattern automation model. Journal of the Korea Institute of Information & Communication Engineering, 21(8).

[8] H. G. Kim, D. H. Jeong, P. K.. Jin, C. M. Han, & G. B. Kim. (2017). $ UsnJrnl Based ransomware encryption pattern typology and detection model. Digital forensic research, 11(3), 71-80.

[9] Berrueta, E., Morato, D., Magaña, E., & Izal, M. (2019). A survey on detection techniques for cryptographic ransomware. IEEE Access, 7, 144925-144944.
DOI : 10.1109/ACCESS.2019.2945839

[10] Kok, S. H., Abdullah, A., Jhanjhi, N. Z., & Supramaniam, M. (2019). Prevention of crypto-ransomware using a pre-encryption detection algorithm. Computers, 8(4), 79.
DOI : 10.3390/computers8040079

[11] Gonzalez, D., & Hayajneh, T. (2017, October). Detection and prevention of crypto-ransomware. In 2017 IEEE 8th Annual Ubiquitous Computing, Electronics and Mobile Communication Conference (UEMCON) (pp. 472-478). IEEE.
DOI : 10.1109/UEMCON.2017.8249052

[12] Usharani, S., Bala, P. M., & Mary, M. M. J. (2021). Dynamic analysis on crypto-ransomware by using machine learning: Gandcrab ransomware. In Journal of Physics: Conference Series (Vol. 1717, No. 1, p. 012024). IOP Publishing.
DOI : 10.1088/1742-6596/1717/1/012024

[13] Y. S. Lee, H. J. Choi, D. M. Shin, & J. J. Lee, (2019). Evaluation of User Abnormal Behavior Detection Performance Based on Deep Learning for Ransomware Prevention. Journal of the Korea Software Appraisal Society, 15(2), 43-50.

[14] Y. C. Hwang. (2023). Extraction and Taxonomy of ransomware features for proactive detection and prevention. Industrial Convergence Research (formerly Journal of the Korean Society of Industrial Management), 21(9), 61-68.
DOI : 10.22678/JIC.2023.21.9.041

[15] Juba, B., & Le, H. S. (2019, July). Precision-recall versus accuracy and the role of large data sets. In Proceedings of the AAAI conference on artificial intelligence (Vol. 33, No. 01, pp. 4039-4048). DOI : 10.1609/aaai.v33i01.33014039

[16] Powers, D. M. (2020). Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation. arXiv preprint arXiv:2010.16061.

황 윤 철(Yoon-cheol Hwang) [정회원]

- 2008년 2월 : 충북대학교 전자계산학과(이학박사)
- 2019년 3월~2021년 2월 : 가천대학교 소프트웨어 중심대학 사업단 소프트웨어교육센터 초빙교수
- 2021년 3월~현재 : 한남대학교 탈메이지 교양・융합대학 조교수
- 관심분야 : 네트워크 및 웹 보안(IDS, ITS), Fusion IT Technology(AI, machine learning)
- E-Mail : dolpin2010@gmail.com