Journal of The Korea Institute of Information Security & Cryptology ISSN 1598-3986(Print)
VOL.34, NO.1, Feb. 2024 ISSN 2288-2715(Online)
https://doi.org/10.13089/JKIISC.2024.34.1.71

효율적인 HWP 악성코드 탐지를 위한 데이터 유용성 검증 및 확보 기반 준지도학습 기법*

손 진 혁,[1†] 고 기 혁,[2] 조 호 묵,[3] 김 영 국[4‡]
[1,2,3]KAIST 사이버보안연구센터 (연구원, 선임연구원, 책임연구원), [4]충남대학교 (교수)

Efficient Hangul Word Processor (HWP) Malware Detection Using Semi-Supervised Learning with Augmented Data Utility Valuation*

JinHyuk Son,[1†] Gihyuk Ko,[2] Ho-Mook Cho,[3] Young-Kuk Kim[4‡]
[1,2,3]KAIST Cyber Security Research Center (Researcher, Senior Researcher, Principle Researcher), [4]Chungnam National University (Professor)

요 약

정보통신기술(ICT) 고도화에 따라 PDF, MS Office, HWP 파일로 대표되는 전자 문서형 파일의 활용이 많아졌고, 공격자들은 이 상황을 놓치지 않고 문서형 악성코드를 이메일과 메신저를 통해 전달하여 감염시키는 피해사례가 많아졌다. 이러한 피해를 막고자 AI를 사용한 악성코드 탐지 연구가 진행되고 있으나, PDF나 MS-Office와 같이 전 세계적으로 활용성이 높은 전자 문서형 파일에 비해 주로 국내에서만 활용되는 HWP(한글 워드 프로세서) 문서 파일은 양질의 정상 또는 악성 데이터가 부족하여 지속되는 공격에 강건한 모델 생성에 한계점이 존재한다. 이러한 한계점을 해결하기 위해 기존 수집된 데이터를 변형하여 학습 데이터 규모를 늘리는 데이터 증강 방식이 제안되었으나, 증강된 데이터의 유용성을 평가하지 않아 불확실한 데이터를 모델 학습에 활용할 가능성이 있다. 본 논문에서는 HWP 악성코드 탐지에 있어 데이터의 유용성을 정량화하고 이에 기반하여 학습에 유용한 증강 데이터만을 활용하여 기존보다 우수한 성능의 AI 모델을 학습하는 준지도학습 기법을 제안한다.

ABSTRACT

With the advancement of information and communication technology (ICT), the use of electronic document types such as PDF, MS Office, and HWP files has increased. Such trend has led the cyber attackers increasingly try to spread malicious documents through e-mails and messengers. To counter such attacks, AI-based methodologies have been actively employed in order to detect malicious document files. The main challenge in detecting malicious HWP(Hangul Word Processor) files is the lack of quality dataset due to its usage is limited in Korea, compared to PDF and MS-Office files that are highly being utilized worldwide. To address this limitation, data augmentation have been proposed to diversify training data by transforming existing dataset, but as the usefulness of the augmented data is not evaluated, augmented data could end up harming model's performance. In this paper, we propose an effective semi-supervised learning technique in detecting malicious HWP document files, which improves overall AI model performance via quantifying the utility of augmented data and filtering out useless training data.

Keywords: Malware Detection, Semi-supervised Learning, Data Utility, Artificial Intelligence, Cybersecurity

Received(11. 09. 2023), Modified(1st: 01. 04. 2024, 2nd: 01. 23. 2024), Accepted(01. 23. 2024)

* 본 연구는 과학기술정보통신부 IT-OT 통합보안을 위한 인공지능 기반 실시간 이상탐지 고도화 및 XAI 기반의 이상징후 간 연관분석 기술 개발(과제고유번호: 1711195576) 사업의 지원을 받아 수행된 연구임

† 주저자, jhson21@kaist.ac.kr
‡ 교신저자, ykim@cnu.ac.kr(Corresponding author)

I. 서 론

정보통신기술(ICT) 기술이 발전으로, 많은 업무 활동이 비대면화(재택근무)되면서 이메일 및 메신저의 활용이 증가했고, 이는 필연적으로 PDF, MS-Office, HWP 등으로 대표되는 전자 문서형 파일 활용의 증가로 이어지게 되었다. 공격자들은 이러한 환경을 이용하여 이메일과 메신저 등으로 전자문서 형태를 띤 문서형 악성코드를 유포하였고, 그 결과로 2020년 초 문서형 악성코드로 인하여 시스템이 감염되는 침해 사례가 속속 발생한 바 있다[1]. 방어자들은 이러한 위협에 효율적으로 대응하기 위해 전자 문서형 악성코드를 탐지하기 위해 다양한 방안을 시도해 왔다.

악성코드를 탐지하는 방법에는 크게 동적 및 정적 분석 방법이 있다. 동적 분석은 실제 파일을 실행하여 악성 행위를 모니터링하는 방식으로[2], 실제 악성코드를 실행하여 분석하기 때문에 탐지의 정확성은 높으나 실행 환경 및 하드웨어 구축에 많은 시간과 높은 비용이 발생한다. 반면 정적 분석은 악성코드를 실행하지 않고 데이터 자체에 존재하는 특성을 분석하여 탐지하는 방식으로, 과거에는 주로 수기로 도출한 규칙(signature)을 기반으로 탐지하였기 때문에 기존에 분석되지 않은 새로운 방식의 악성 공격의 탐지가 어렵다는 단점이 존재한다. 이와 같은 문제를 효과적으로 해결하고자 최근 기계학습 및 AI를 활용한 악성코드 탐지 연구가 시도되어 높은 탐지 성능을 보이고 있으며[3], 하드웨어 및 알고리즘의 발전으로 AI의 전성기가 찾아오면서 다양한 AI 아키텍처에 기반한 악성코드 및 문서형 악성코드 탐지 연구가 활발하게 진행되고 있다.

한편, 효율적인 AI 학습을 위해서는 양질의 학습 데이터셋이 전제되어야 하며, 사이버보안 분야는 위협에 따라 데이터가 지속해서 변화하므로 양질의 학습 데이터 수급에 어려움이 따른다. 이러한 데이터 수급 문제를 해결하지 않고 제한된 학습 데이터로 복잡한 AI 모델을 학습할 때 적은 수의 학습 데이터 자체를 암기하여 일반적인 표상을 학습하지 못하게 되는 과적합(Overfitting) 현상 또는, 전체적으로 학습이 충분히 되지 못해 성능이 떨어지는 과소적합(Underfitting) 현상이 발생한다[4]. 데이터의 부족에서 야기되는 이러한 문제를 해결하는 효과적인 방법으로 데이터 증강(Data Augmentation)을 통한 준지도학습(Semi-supervised Learning) 기법을 들 수 있다. 데이터 증강이란 적은 양의 데이터를 기반으로 수정, 치환, 합성 등의 다양한 알고리즘을 통해 데이터 수를 늘리는 기법을 말하며[4], 이를 통해 학습 데이터에 존재하는 불균형 현상도 해소할 수 있다. 실제로 악성코드 판별에 있어 학습 데이터의 불균형성은 가장 큰 애로사항으로 꼽히는 문제로[5], 이를 해결하기 위해 소수 클래스인 악성코드와 관련이 있는 데이터를 증강하여 정상 데이터와 비슷한 비율을 맞추는 등의 다양한 해결책이 제시되었다.

그러나 데이터 증강 기법은 데이터를 다양한 방식으로 증강할 뿐 새로 생성된 데이터가 과연 주어진 문제를 수행하는 데에 도움이 되는지 아닌지를 판단하지 않는다. 학습을 위해 대량의 데이터를 보유하고 있다 하더라도 그 데이터가 학습에 방해되는 데이터, 즉, 유용성이 떨어지는 데이터를 사용해 AI 학습을 할 경우 오히려 성능이 떨어지는 문제가 발생할 수 있기 때문에 증강된 데이터의 신뢰성을 판단하는 과정은 매우 중요하다. 특히, 사이버보안 분야와 같이 AI 모델이 안전과 관련된 결정을 내리는 경우 학습 데이터의 신뢰도를 검증하는 것은 더더욱 중요하다고 할 수 있다. 실제로 데이터 부족 현상을 해결하기 위해 데이터 증강을 적용하여 악성코드 탐지 AI 모델의 성능을 높이는 방안이 여럿 제안되었으나[1,6,7], 데이터 증강 기법만 활용할 뿐 증강된 데이터 각각의 유용성을 확인하지 않고 학습에 사용한 바 있다.

따라서 본 논문에서는 학습 데이터가 부족한 상황에서 데이터 증강을 통해 보완된 데이터 각각에 대해 그 유용성을 정량화함으로써 학습에 도움이 되지 않는, 혹은 학습에 방해되는 데이터를 걸러내는 준지도학습 기법을 제안한다. 본 논문에서 제안하는 기법에 대하여 최근 많은 문제를 야기하고 있는 HWP 문서형 악성코드를 사용한 실험을 통해 증강 데이터의 유용성 점수 및 필터링이 더욱 효율적인 AI 기반 악성코드 탐지를 가능케 함을 보인다.

II. 관련 연구

2.1 인공지능 기반 문서형 악성코드 탐지

AI 모델을 활용하여 문서형 악성코드를 효율적으로 탐지하기 위해 다양한 연구들이 제안되어왔다. 먼저 MS Office 문서형 악성코드를 탐지하기 위해 악성파일을 구분할 수 있는 특성(feature)을 추출하여

SVM(Support Vector Machine), MLP (Multi-Layer Perceptron) 등의 탐지기를 학습하는 연구들이 존재한다[8,9,10], 또한 [11,12]에서는 CNN(Convolutional Neural Network)을, [13]에서는 언어 기반 모델링을 사용하여 악성 MS Office 문서를 탐지하였으며, [14]에서는 MS Office 파일 내 바이트 스트림(Byte Stream) 각각의 악성 여부를 판단하는 AI 모델을 활용하여 악성코드를 탐지하는 기법을 제안하였다.

우리나라 정부와 공공기관에서 주로 활용하는 HWP 문서에 대한 악성코드를 AI 모델을 활용하여 탐지하려는 연구 또한 존재한다. [5]에서는 HWP 문서에 존재하는 키워드 및 데이터를 기반으로 다양한 특성을 추출하여 AI 기반 탐지기를 학습하였다. [15,16]에서는 HWP 파일 내의 바이트 스트림 각각의 악성 여부를 CNN 기반의 모델로 판단하는 기법을 제안하였다.

본 연구에서는 HWP 문서형 악성코드 탐지를 위한 새로운 AI 모델 제안에 중점을 두는 것이 아닌 HWP 문서형 악성코드 탐지에 데이터 증강 및 유용성 측정 기법을 접목하는 데에 두고 있음을 명기한다. 이를 위해, 본 논문에서는 가장 기본적인 HWP 악성코드 탐지 방법인 특성 기반 탐지 방법[5]에 그 초점을 맞춘다.

2.2 데이터 증강을 통한 준지도 학습

데이터 증강(Data Augmentation)을 통한 준지도학습(Semi-supervised Learning)은 학습에 필요한 데이터의 절대적인 수나 그 다양성이 부족할 때, 기존의 데이터를 활용하여 새로운 데이터를 만들어 학습하는 기법을 말한다. 데이터 증강에는 기존 데이터에 무작위 잡음을 주입[17]하는 간단한 방법에서부터 여러 데이터에 대한 가중치 평균치를 새로운 데이터로 사용[18]하거나, GAN(Generative Adversarial Network)[19]과 같은 생성형 AI를 활용하는 등 다양한 방법이 존재한다.

데이터 증강을 통한 준지도학습은 컴퓨터 비전 분야에서 일찍이 그 효용성을 보였[18,20]을 뿐만 아니라 양질의 데이터셋을 구축하기 까다로운 사이버보안 분야에서도 적극적으로 활용하였다. [6]과 [7]에서는 악성코드를 이미지로 변환한 데이터에 대해 회전 등 이미지 특화 데이터 증강 기법 및 생성형 AI를 적용하여 데이터 증강의 효과를 보였으며, [1]에서는 HWP 문서형 데이터에 잡음 주입 및 여러 데이터의 가중치를 적용하는 방식의 Mix-up 등 다양한 데이터 증강 방법을 적용하여 그 효과를 입증하였다. 이들 연구는 데이터 증강을 통해 효과적으로 AI 기반 사이버보안 위협 탐지 성능을 높였지만, 증강된 데이터 중 어떤 데이터가 학습에 유용한지에 대한 분석은 수행된 바 없다.

2.3 데이터 유용성 측정 및 필터링

학습 데이터의 유용성은 AI 모델의 품질 및 신뢰성과도 직결되며, 이에 따라 AI 학습 데이터의 유용성을 판단하여 유용하지 않은 데이터를 걸러내는 필터링 과정은 기계학습의 다양한 분야에서 널리 활용. 일례로 AI 학습 데이터의 품질을 높이기 위해 주어진 데이터 분포 및 데이터 간 유사도(Similarity)에 기반하여 극단치(Outlier)를 제거할 수 있다. 이러한 데이터 정제(Data cleansing) 프로세스는 이미지[21] 및 텍스트[22]뿐만 아니라 악성코드[23]나 네트워크 로그[24]와 같은 사이버보안 데이터에도 적용되어 AI 모델의 성능 및 신뢰성을 향상할 수 있다.

단일 데이터가 지니는 유용성의 측정은 AI 신뢰성 향상 외에도 그 자체로서 의미를 갖기도 한다. 대표적인 예로서 데이터 평가(Data Valuation)는 각각 데이터가 가지는 가치를 수치화하여 평가함으로 데이터의 거래 시 유용한 지표를 제공한다. 데이터 평가 방식에는 Leave-One-Out(LOO), Shapley Value 등의 기법이 존재한다[25].

준지도학습 환경에서 데이터 유용성을 판단하는 연구 또한 진행된 바 있다. [26]에서는 협력 게임이론으로부터 도출한 Shapley Value를 기반으로 주어진 데이터의 유용성을 평가하여 유용한 데이터만 활용하였을 때 준지도학습 환경에서 더 높은 정확도를 보일 수 있음을 보였다. 하지만 [26]의 방법론은 Shapley Value의 계산에 많은 컴퓨팅 자원이 소모되며 데이터 증강 세팅 및 사이버보안 분야에 적용되지 않았다. 이에 반하여 본 논문에서는 데이터가 부족한 사이버보안 분야, 특히 HWP 문서형 악성코드 탐지에 데이터 유용성 기반의 준지도학습 기법이 얼마나 효율적인지에 초점을 맞추어 연구한다.

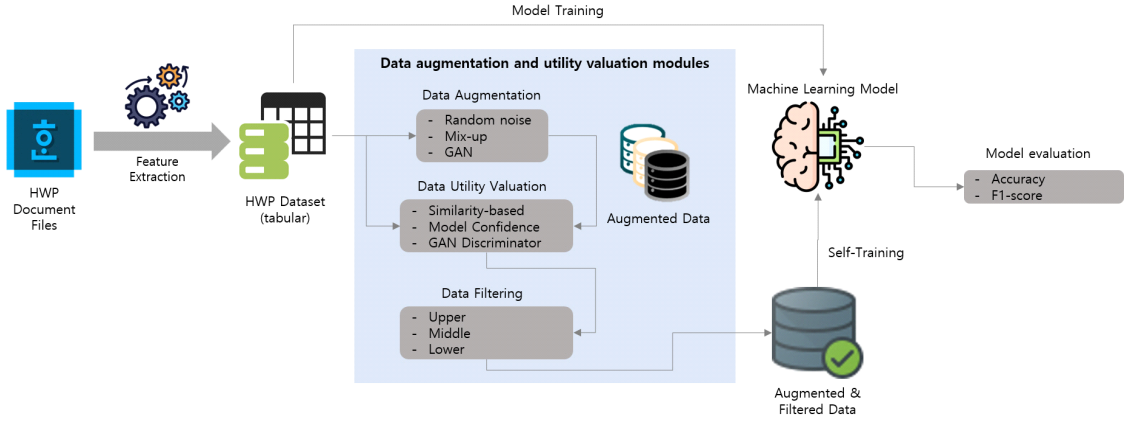


Fig. 1. Semi-supervised Learning with Augmented Data Utility-based Filtering for HWP Malware Detection

III. 데이터 유용성 기반 준지도학습 기법

본 장에서는 보다 효율적인 악성코드 탐지를 위해 데이터 유용성을 기반으로 한 준지도학습 기법을 제안한다. 본 연구에서 제안하는 기법은 크게 두 가지 절차를 통해 진행된다. 먼저 주어진 데이터셋을 다양한 기법을 사용하여 증강함으로써 증강 데이터셋을 확보한다. 다음으로는 증강된 데이터 각각에 대하여 그 유용성을 정량화하고, 데이터를 필터링하는 기준에 따라 최종 학습 및 증강 데이터를 확보한다. Fig 1에서 본 연구에서 제안하는 데이터 유용성 기반 준지도학습 기법의 흐름을 확인할 수 있다.

본 연구에서는 HWP 파일을 전처리하여 특정 키워드의 개수, 존재 여부, 파일 크기 등을 추출한 테이블형 데이터를 사용한다. 데이터 전처리에 관한 설명은 제4장에 상세하며, 본 장에서는 준지도학습 방법론에 그 초점을 두어 설명한다.

3.1 데이터 증강

데이터 증강은 학습 데이터가 부족하거나 레이블 간 비율이 치우친 불균형 데이터셋에서 기존 데이터에 조금 변형을 가하여 새로운 학습 데이터를 생성하는 기법이다. 본 연구에서는 증강 방식에 다양성을 부여하기 위해 대표적인 데이터 증강 방식들로 꼽히는 무작위 잡음(Gaussian Noise) 주입, 데이터 혼합(Mix-up), 그리고 GAN 기반 증강(TGAN)의 총 세 종류의 증강 방식을 구현하였다.

3.1.1 Gaussian Noise

무작위 잡음 주입 기법이란 데이터 값에 잡음을 넣어 증강하는 방식으로, 매우 오래된 증강 방법(Holmstrom & Koistinen, 1992[17])중 하나이지만 적용하기 쉽고 효과가 좋기 때문에 널리 이용되는 증강 기법이다. 잡음 추가 기법을 활용한 데이터 증강은 AI 학습 모델의 각 클래스의 다양한 측면을 학습하도록 할 수 있을 뿐만 아니라, 좀 더 견고한 AI 모델을 학습할 수 있도록 하는 장점이 있다.

본 연구에서는 전처리된 HWP 테이블형 데이터에 특성 별 잡음을 주입하기 위해 먼저 최대-최소(Min-Max) 정규화를 적용하여 데이터를 변형한 후, 특성 값의 분포에 따라 Gaussian 잡음을 생성하여 추가하는 방식을 사용하였다. 이러한 잡음 주입은 증강 전후 데이터 분포를 보존한다.

3.1.2 Mix-up

Mix-up이란 데이터 혼합을 의미한다. 두 개 혹은 그 이상의 데이터를 섞어서 새로운 데이터를 생성하는 방법으로 데이터의 부족 현상을 해결하기 위해 사용한다. Fig 2는 데이터 혼합의 한 예로, 두 개의 데이터의 특성은 4개를 가지고 있다. 두 데이터는 합이 1인 가중치를 통해 두 데이터를 이은 선 위에 존재하며 비율에 따라 무한하게 데이터를 생성할 수 있다. 본 연구에서는 전처리된 HWP 테이블형 학습 데이터 각각에 대해 무작위로 다른 데이터를 뽑아 두 개의 데이터를 각각 0.5의 가중치를 적용하여 혼합함으로써 새로운 데이터를 만들었다.

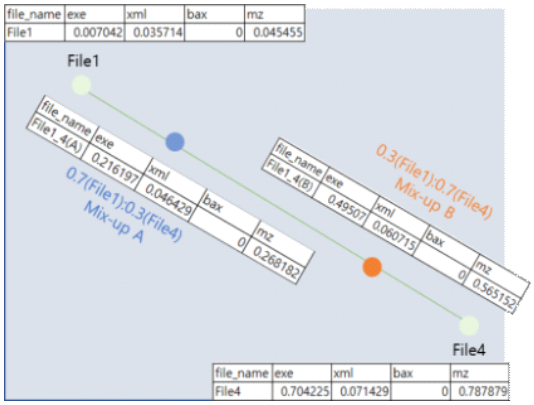


Fig. 2. Mix-up Data Augmentation

3.1.3 TGAN[28]

GAN(Generative Adversarial Network)는 생성 AI 모델의 대표적인 예시로서 학습한 데이터와 유사한 데이터를 생성하는 데에 유용하다[19]. GAN은 생성 모델(Generator)과 판별 모델(Discriminator)의 두 가지 모델로 구성되는데, 생성 모델은 임의의 잡음을 사용하여 학습 데이터와 유사한 가짜 데이터를 생성하고, 판별 모델은 생성 모델이 생성한 가짜 데이터와 실제 학습 데이터와 구분하는 모델이다. GAN 학습이 진행되면서 생성 모델은 점차 실제 데이터와 유사한 가짜 데이터를 만들 수 있게 되며, 판별 모델은 가짜 데이터를 효과적으로 분류할 수 있게 된다. Fig 3은 일반적인 GAN 추론 프로세스를 표현한 도식이다.

본 연구에서는 테이블형으로 전처리된 HWP 데이터를 활용하였기 때문에, 테이블형 데이터 증강에 적합한 Tabular GAN(TGAN)[27] 아키텍처를 활용하였다. 테이블형 데이터의 다양한 데이터 형식과 분포, 불균형한 범주형 데이터 등으로 인해 발생하는

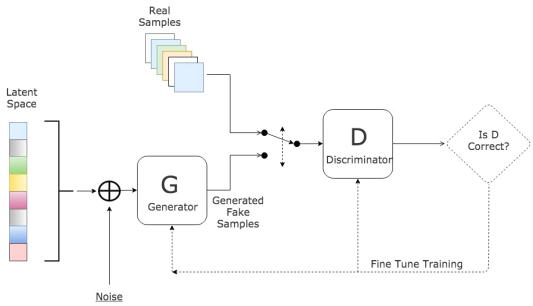


Fig. 3. GAN Inference Process [27]

제한사항[28]을 극복하고자 테이블의 특징점을 one-hot-encoding 형태로 변형하여 동일한 분포의 데이터 형태와, 동일한 범주의 형태로 변환할 수 있는 모델을 생성하였으며, 유용성 점수 필터링시 생성 데이터를 변형할 수 있도록 모델을 저장하였다.

3.2 증강된 데이터의 유용성 정량화

본 연구에서는 증강된 데이터의 유용성을 정량화하여 그 점수에 따라 필터링함으로써 데이터 증강의 효율성을 극대화하고자 한다. 이를 위해 증강된 데이터의 유용성을 측정하는 데에 세 가지 방법을 사용하였다. 가장 먼저 증강된 데이터와 학습 데이터와의 유사도(Similarity)를 측정한다. 다음으로는 증강 없이 원본 학습 데이터만을 활용하여 학습한 AI 판별 모델이 출력하는 예측 점수(Confidence Score)를 활용하였다. 마지막으로는 3.1.3절에서 학습한 TGAN 모델의 구분자 모델을 활용하여 증강된 데이터가 얼마나 실제와 유사한지를 평가하였다.

본 연구에서는 이처럼 세 가지 서로 다른 방식으로 측정한 유용성 점수를 기반으로 증강 데이터를 필터링하여 사용한다. 필터링은 증강된 데이터 중 유용성 점수가 높은 순서대로 상위 25% 혹은 하위 25%만을 학습 데이터에 추가하여 사용하였다. 상위 25%의 경우 기존 데이터와 유사한 증강 데이터를, 하위 25%의 경우 기존 데이터와 유사하지 않은 증강 데이터를 추가하였을 때의 효과 등 다양한 데이터 조합에서의 성능 효과를 살펴볼 수 있다.

본 연구에서 제안하는 유용성 점수는 증강된 데이터의 실제 정상 혹은 악성 여부를 측정하는 것이 아닌, 어디까지나 각각의 데이터가 AI 탐지 모델의 성능 향상에 도움이 되는지 여부를 판단하기 위하여 활용하는 간접적 수치임을 명시한다. 즉, 유용성 점수가 높다고 반드시 증강된 데이터가 특정 레이블에 속한다는 보장은 없으며, 본 연구에서는 그 의미보다는 데이터를 순차적으로 나열하기 위한 도구 지표로서 사용하는 것이다.

3.2.1 데이터 유사도(Similarity) 기반 유용성 점수

데이터 유사도란 서로 다른 데이터 간 얼마나 유사한가를 정량화한 것으로, 거리 기반, 각도 기반 등 다양한 계측 방식이 존재한다. 본 연구에서는 증강된 데이터 각각에 대해, 학습 데이터셋 중 동일한 클래

스 데이터 군집과의 평균적인 유사도를 계측하였다.

본 논문에서는 데이터 유사도 측정에 맨하탄(L1) 거리와 유클리디안(L2) 거리, 및 코사인(Cosine) 거리 유사도를 활용하였다. 맨하탄 유사도는 두 벡터 간 차이의 절대값을 취하는 방식이고, 유클리디안 유사도는 두 벡터 간 차이 제곱의 합을 제곱근 한 값을 의미하며, 마지막으로 코사인 유사도는 내적공간의 두 벡터 간 각도의 코사인값에 기인한 거리이다.

3.2.2 모델 예측(Confidence) 기반 유용성 점수

원본 학습 데이터를 활용하여 만든 AI 탐지 모델을 활용하여 증강된 데이터의 점수를 측정할 수 있다. 먼저 데이터 증강 없이 원본 학습 데이터를 이용하여 HWP 악성코드 탐지 모델을 생성한다. 생성한 모델에 데이터를 입력하면 Benign(정상)의 확률과 Malware(악성)의 확률이 나오는데, 예를 들어 하나의 입력 파일을 넣었을 때 [0.01, 0.99]의 값이 나온다면, 이는 Benign의 확률은 1%이고 Malware의 확률은 99%라는 의미다. 이 모델에 새로이 증강된 데이터를 입력하여 해당 데이터 클래스의 확률값을 점수로 활용할 수 있다. 예를 들어, 악성(Malware) 증강 데이터를 모델에 입력하였을 때 Benign 0.2, Malware 0.8이 각각 출력되었을 경우 해당 데이터의 점수는 0.8로 산정한다.

3.2.3 GAN 판별 모델 기반 유용성 점수

GAN의 판별 모델(Discriminator)은 학습이 진행되면서 주어진 데이터가 실제(Real) 학습 데이터인지 생성된 가짜(Fake) 데이터인지를 잘 구분할 수 있게 된다. 본 연구에서는 앞서 3.1.3절에서 데이터 증강을 위해 학습한 TGAN의 판별 모델을 이용하여 증강 데이터의 유용성 점수를 계산한다. TGAN의 판별 모델을 활용하기 위해서는 먼저 입력 데이터의 변환이 필요한데, 본 연구에서는 TGAN 증강에 사용한 one-hot-encoding 변형 모델을 사용하여 데이터를 변형한 후 판별 모델에 입력하여 유용성 점수를 계산하였다.

IV. 실험 및 결과

4.1 실험 방법

본 연구에서는 MLP(Multi-Layer Perceptron), SVM(Support Vector Machine), 그리고 LR(Logistic Regression) 총 세 종류의 HWP 악성코드 탐지 AI 모델을 사용하여 데이터 증강의 효용성을 실험하였다. 인공 신경망 구조를 띈 MLP는 각각 256, 16의 크기를 가진 2개의 hidden layer로 구성하였다. SVM과 LR 모델은 Python 기계학습 패키지인 sklearn에서 제공하는 기본 모델(SVM: RBF 커널, C=1.0, LR: L2 페널티, C=1.0)을 활용하였다.

HWP 악성코드 탐지 성능을 계측하기 위해 정확도 및 F1-score(Precision/Recall)를 측정하였다. 이때 학습 및 테스트 데이터셋 구분에 따른 영향이 최소화하도록 5-Fold Cross Validation을 통해 데이터 증강의 일반적인 성능을 측정하였다. 최종 성능 평가는 5개의 Confusion Matrix의 평균값을 사용하여 측정하였다. 데이터 증강으로는 3장에서 소개한 Gaussian Noise, Mix-up, 그리고 TGAN 방식을 활용하였으며, 증강 후 데이터의 필터링에는 유사도(Similarity) 세 가지(L1, L2, Cosine), 모델 예측 점수(Confidence), GAN 판별 모델(GAN Discriminator)의 총 5가지를 활용하였다. 증강 후 필터링 적용시 각 점수의 상위(High) 또는 하위(Low) 25%에 해당하는 증강 데이터만을 학습데이터로서 선택하고 나머지 75%를 필터링(Filter Out)하여 새로운 AI모델을 학습함으로써 증강 및 필터링 전후 성능을 분석하였다.

4.2 데이터 및 전처리

본 실험을 위해 총 3,071개(정상 2,500개, 악성 571개)의 HWP 파일을 사용하였다. 정상 파일의 경우 공공기관 및 국립대학교의 공개 게시판에서, 악성파일은 VirusTotal에서 다운로드 받아 수집하였다. 5-Fold Cross Validation 실험을 위해 먼저 전체 데이터셋을 균등한 정상:악성 비율을 가지는 5개의 군집(split)으로 나누고, 1개의 군집을 테스트 데이터셋으로, 나머지 4개의 군집을 학습 데이터셋으로 사용하여 각 군집에서의 성능 평가를 5회 반복하였다. 각 실험에서 데이터 증강은 총 20번 적용하

였으며, 증강된 데이터에 대해 상위 혹은 하위 25% 데이터만을 필터링하여 사용하여 본 논문에서 제안한 기법의 효용성을 측정하였다. Table 1은 Split 0의 원본 학습 및 테스트 데이터 구성과 데이터 증강(Aug.), 그리고 필터링(A+F) 이후 학습 데이터 수를 나타낸다. Split 1-4까지는 악성 학습 및 테스트 데이터가 각각 457, 114개로 구성되며, 증강 및 필터링에 Table 1과 동일한 산식이 적용된다.

HWP 데이터를 AI 모델이 학습할 수 있는 테이블형 데이터로 전처리하기 위해 Table 2와 같이 총 129개의 특성을 추출하였다. 추출한 특성은 'exe', 'xml', 'bax', 'jar', 'dll'등 의심스러운 키워드 기반으로 등장 횟수, 'yara', 'cve'처럼 yara 탐지 결과, 'copyfile', 'convert', 'threadid'와 같은 window api keyword 횟수, 'eval', 'unescape', 'js size' 등과 같은 javascript와 관련된 특성들, 'eps', 'ps', 'string dup'과 같은 shell 혹은 ghost script와 관련된 특성에, HWP에만 존재하는 tag_id가 있는데, 그 중 일부 tag_id인 'tag_id_67(generic43)', 'tag_id_66(generic44)'등의 등장 횟수를 특성으로 사용하며, 마지막으로 파일 내부 스트림의 개수(length)를 사용한다.

TGAN 모델을 사용하기 위해서는 전처리된 데이터의 형식을 변환해서 사용해야 하는데, TGAN 증강 모델 생성시 저장한 변형모델을 통해 각 특징점을 one-hot-encoding 데이터로 변형하였다. 기존 129개의 특징은 총 302개의 특징으로 변형했으며, 이 변형 모델은 하나의 특징을 2개부터 8개까지 다량변형하였고, 최소 변형 개수는 'bax', 'jar', 'shell' 등으로 2개이며 최대 변형 개수는 8개로 'ip', 'tag_id_76' 등이 있으며 평균적으로 3.31개의 변형 형태를 가진다.

Table 1. Training and Test Data Counts (Split 0)

	Training			Test
	Orig.	Aug.*	A+F**	
Benign	2,000	42,000	12,000	500
Malware	456	9,576	2,736	115
Total	2,456	51,576	14,736	615

* Aug.: Data after Augmentation (20 times)
Data Count = Orig + Orig*20
** A+F: Augmented then Filtered (25%)
Data Count = Orig + Orig*20*0.25

Table 2. Summary on HWP File Features

Feature Type	Data Type	Examples	Cnt*
Malicious keywords	INT	exe, xml, bax	53
YARA detect result	INT	yara, cve	2
API keywords	INT	copyfile, convert, threadid	35
JS-related	INT	eval, unescape, js size	5
Shell code realted	INT	eps, ps, string dup	5
Existence	BOOL	%u9090%u9090, exec|xor, exec|cvx	3
# of Tag_ids	INT	generic43, generic44, tag_id68	28
# of Streams	INT	length	1

*Cnt: Feature Counts

4.3 실험 결과 및 분석

데이터 증강의 효과를 확인하기 위해 MLP, SVM, LR 모델에 잡음 추가, 혼합 방식, GAN 증강 방식 세 가지 증강법을 적용하여 기준 성능 지표와 데이터 증강 후 모델 지표를 생성하였다. 그리고 해당 증강 데이터를 필터링 후 모델의 성능일 비교 분석하였다. Table 3은 MLP, SVM, LR 모델 별 기준 성능, 증강 성능, 증강 및 필터링 중 가장 좋은 성능을 비교한 테이블로 기준 성능보다 좋은 지표를 보인 필터링의 경우는 밑줄을 표기하였으며 증강 성능보다 좋은 지표를 보이면 **굵은** 글자로 표기하였다. 먼저 모델별로 살펴보면, MLP 모델에서 가장 좋은 분류 성능을 보이지만, 전체적으로 성능이 떨어진 것을 볼 수 있는데, 이는 데이터를 증강할 때 학습을 방해하는 데이터들이 증강되는 것을 의미하며, 이 경우 데이터 필터링을 통해 기준 성능 및 증강 성능보다 우수한 것을 볼 수 있다. SVM 모델과 LR 모델의 경우에는 GAN 증강 방식을 제외하면 증강의 성능이 향상된 것을 볼 수 있지만, GAN 증강 방법은 다른 증강 방법에 비해 성능이 떨어지는 것을 확인할 수 있다, 필터링의 경우 증강 방법보다 F1-score 0.0329, 정확도 약 6.35% 향상을 보이는 것을 확인할 수 있다. LR모델은 나머지 2개의 모델에 비해

Table 3. Effect of Augmentation (and Filtering) on the Prediction Performance under Different ML models

Model	Aug. method	Baseline		Augmented		Augmented+Filtered (Best)**	
		F1-Score*	Acc.	F1-Score	Acc.	F1-Score	Acc.
						Filtering Method	
MLP	Gaussian Noise	0.9628 (0.9800/ 0.9464)	0.9384	0.9613 (0.9692/0.9535)	0.9365	**0.9658** (0.9788/0.9532)	**0.9436**
						Confidence-based	
	Mix-up			0.9594 (0.9648/0.9541)	0.9335	**0.9638** (0.9796/0.9485)	**0.9400**
						Similarity-based (Cosine)	
	TGAN			0.9514 (0.9764/0.9280)	0.9189	**0.9660** (0.9784/0.9540)	**0.9439**
						Confidence-based	
SVM	Gaussian Noise	0.9462 (0.9888/ 0.9072)	0.9084	0.9550 (0.9884/0.9230)	0.9244	0.9537 (0.9860/0.9236)	0.9221
						Similarity-based (L1)	
	Mix-up			0.9546 (0.9884/0.9234)	0.9238	0.9529 (0.9836/0.9241)	0.9208
						Similarity-based (Cosine)	
	TGAN			0.9052 (0.9996/0.8272)	0.8296	**0.9381** (0.9940/0.8881)	**0.8931**
						Confidence-based	
LR	Gaussian Noise	0.9295 (0.9900/ 0.8791)	0.8778	0.9403 (0.9832/0.9010)	0.8984	0.9363 (0.9856/0.8918)	0.8909
						Similarity-based (L1)	
	Mix-up			0.9403 (0.9832/0.9010)	0.8984	0.9356 (0.9784/0.8997)	0.8909
						Similarity-based (L2)	
	TGAN			0.9020 (0.9980/0.8212)	0.8215	**0.9162** (0.9840/0.8571)	**0.8534**
						Similarity-based (Cosine)	

* F1-Scores are presented in the form "F1-Score(Precision/Recall)"
** 25% of the augmented data are filtered and used in training. Among 3 filtering methods, the best performance and corresponding filtering method are shown
- Underline: performance better than Baseline
- **Bold**: performance better than Augmented

가장 안좋은 성능을 보였으며, SVM과 마찬가지로 GAN 증강 방식을 제외하면 데이터 증강이 효과적인 것을 볼 수 있었다. 필터링을 적용할 경우 SVM과 GAN 증강 방식에서 효과를 볼 수 있었는데, F1-score의 경우 0.0142 정확도는 약 3.19% 향상된 것을 볼 수 있다.

MLP, SVM, LR 모두 기준 모델보다는 필터링을 적용한 증강 방법을 했을 때 성능 향상을 보였으며, 그중 MLP 모델이 가장 우수한 성능을 보였다. 증강 방법론 관점으로 살펴보면, 잡음 추가 방식과 혼합 방식에서는 필터링 적용과 관련없이 향상된 성능을 보였지만, GAN증강 방식에서는 MLP를 제외하고 증강만 적용할 때는 오히려 성능이 악화한 것을 확인할 수 있었지만, 증강 필터링을 적용할 시 가장 높은 향상 폭이 있음을 확인할 수 있었다. 이는 GAN증강시에는 모델 학습에 악영향을 끼칠 데이터가 생성되며 이를 적절하게 처리하지 않으면 문제가 있음을 알 수 있다.

Table 4. Effect of Filtering Methods and Criteria on the Prediction Performance (MLP/TGAN)

Filtering Method	Criteria	F1-Score*	Acc.
Baseline		0.9628 (0.9800/0.9464)	0.9384
Augmented**		0.9514 (0.9764/0.9280)	0.9189
Augmented+Filtered***			
Similarity (L1)	High	**0.9644** (0.9824/0.9472)	**0.9410**
	Low	**0.9539** (0.9812/0.9282)	**0.9228**
Similarity (L2)	High	**0.9639** (0.9796/0.9489)	**0.9404**
	Low	**0.9604** (0.9812/0.9406)	**0.9342**
Similarity (Cosine)	High	**0.9592** (0.9788/0.9404)	**0.9322**
	Low	**0.9640** (0.9808/0.9478)	**0.9404**
Model Confidence	High	**0.9660** (0.9784/0.9540)	**0.9439**
	Low	**0.9605** (0.9748/0.9468)	**0.9348**
GAN Disc.	High	**0.9566** (0.9804/0.9341)	**0.9277**
	Low	0.9514 (0.9828/0.9220)	0.9182

* Presented as: "F1-Score(Precision/Recall)"
** All data augmented using TGAN
*** 25% of the augmented data are filtered and used in training.
- Underline: performance better than Baseline
- **Bold**: performance better than Augmented

필터링 조건별 성능을 확인하기 위해 Table 4와 같이 정리하였다. AI 탐지 모델은 Table 3에서 가장 성능이 좋았던 MLP 모델을 활용했고, 증강 방법으로는 증가 폭이 큰 GAN 증강 방법으로 고정하여 기준 모델과 증강 모델, 그리고 다섯 가지 필터링 종류에 두 가지 조건의 모델을 포함하여 10개를 비교하였다. 밑줄과 굵은 글자는 Table 3과 마찬가지로 기준 모델보다 좋을 때 밑줄을 표기했고, 증강 모델보다 좋을 때 굵은 글자로 표기하였다. GAN 방식의 낮은 점수 필터링을 제외하고, 전체적으로 증강 필터링을 적용했을 때 성능이 좋은 것을 볼 수 있다. 먼저 유사도 점수 중 거리기반 점수(L1, L2)를 보면 모두 높은 점수로 필터링했을 경우 성능이 향상된 것을 볼 수 있다. 점수가 크다는 의미는 증강된 데이터가 기존 데이터와의 거리가 먼 것을 의미할 수 있는데, 점수 계산 시 증강 데이터와 기존 데이터 중 가장 가까운(L1_min) 집단의 거리로 유사한 데이터지만, 그중에서 가장 유사하지 않은 데이터를 추가할 경우 효과적인 것을 볼 수 있다. 유사도 기반 중 각도 기반인 cosine의 경우 증강 데이터와 기존 데이터의 cosine 값을 점수로 했으며, 각도가 클수록 서로 다른 방향의 데이터로 해석할 수 있는데, 낮은 점수 필터링의 결과가 좋다는 점은 유사도가 높은 데이터를 추가하는 것이 유사도가 적은 데이터를 추가하는 것보다 모델 성능 향상에 도움을 준 것을 확인할 수 있다. Confidence Value기반 증강의 점수는 기존 모델이 예측한 확률값으로 높은 점수일수록 모델이 더 확신하는 값을 점수로 했으며, 이는 높은 점수라 하면 모델이 더 높은 해당 데이터를 더 높은 확신을 하고 판단했다는 의미로 해석할 수 있다. 해당 방식의 필터링을 적용할 경우, 모델 친화적인 증강 데이터가 그렇지 않은 데이터보다 좀 더 높은 성능을 보이는 것을 확인할 수 있다. 마지막으로 GAN 판별자 필터링은 데이터가 실제 데이터인지 생성 데이터인지에 대한 점수를 표기하는데, 이 역시 높은 점수일수록 잘 만들어진, 잘 속일 수 있는 데이터라는 의미이며 높은 점수일수록 조금 더 좋은 성능을 확인할 수 있다.

본 연구에서 제안한 준지도학습 기법이 상대적으로는 우수한 성능을 보였으나 절대적으로 큰 효과를 보이지 못했는데, 이는 주어진 HWP 데이터셋에서 기본 모델이 이미 높은 성능(>90%)을 보였기 때문으로 생각된다. 이 밖에도 자체 구축한 데이터셋에서 실험하여 과거 결과들과 직접적인 비교가 어려운 점, 데이터 유용성 점수 기반 필터링에 대해서 정밀한 실험 고안을 통한 효과를 검증하지 못한 점, 데이터의 증강 및 필터링 적용시 일괄적으로 성능을 향상시키는 기법이 존재하지 않는 것 등의 한계점이 존재한다. HWP 문서형 악성코드 탐지 외에 더 범용적이고 어려운 사이버위협 대응 분야 등 다양한 데이터셋에 본 논문에서 제안한 기법을 적용 및 실증하여 범용적이고 실질적인 유용성 점수에 기반한 준지도학습 기법을 개발하는 것은 추후 연구로 남긴다.

V. 결 론

본 논문에서는 데이터 증강을 통해 부족한 한글 문서형 악성코드 데이터를 추가 확보하여 증강 전 데이터만을 활용한 학습 모델보다 좋은 성능의 모델을 생성하였다. 기존의 데이터 증강 기법에서는 데이터 증강에만 초점을 맞추고 증강된 데이터의 유용성에 대해서는 검증을 하지 않지만, 본 논문에서는 증강된 데이터에 대한 유용성 점수를 도입함으로써 보다 유의미한 데이터를 취득할 수 있고 이로 인해 그로 인해 좋은 탐지 모델 결과를 얻을 수 있었다. 그러나 모든 증강 방법과 모든 필터링에서 좋은 결과를 보인 것은 아니며 데이터의 형태 및 특징에 따라 다른 방법론을 이용하여 필터링을 적용하기 위해서는 더 많이 증강해야 한다는 단점 또한 존재하였다.

향후 연구에서는 보다 범용적인 사이버위협 대응 분야에 본 연구에서 제안한 기법을 적용하여, 데이터 증강 기법에 따른 최적의 유용성 점수 및 필터링 기술을 고안하여 AI 기반의 탐지 성능을 향상하는 준지도학습 기법을 개발할 예정이다.

References

[1] J.H Son, G. Ko, and H. Cho, "Learning Data augmentation Method for Effective Detection of HWP Malware," Korea Software Congress(KSC), pp. 923-931, Dec. 2022

[2] K.C Yeon, "Malicious factor analysis using HWP document format structure," MA Thesis, Department of Information Security Graduate School of Information Security Korea University, Jun. 2016

[3] T.C. Truong and Z. Ivan, "A survey on artificial intelligence in malware as next-generation threats," Mendel, Vol. 25. No. 2, pp. 27-34, Dec. 2019.

[4] H.C Cho and J. Moon. "A layered-wise data augmenting algorithm for small sampling data," Journal of Korean Society for Internet Information, vol. 20 no. 6, pp. 65-72, Dec. 2020

[5] J. Zhang, Z. Qin, H. Yin, L. Ou, S. Xiao, and Y. Hu, "Malware variant detection using opcode image recognition with small training sets." 2016 25th International Conference on Computer Communication and Networks(ICCCN), pp.1-9, Aug. 2016.

[6] R. Burks, KA. Islam, Y. Lu, and J. Li, "Data augmentation with generative models for improved malware detection: A comparative study." 2019 IEEE 10th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON), pp. 660-665, Oct. 2019.

[7] F.O Catak, J. Ahmed, K. Sahinbas, and Z.H Khand, "Data augmentation based malware detection using convolutional neural networks". PeerJ Comput, vol. 7, no. e346, pp. 1-26, Jan. 2021

[8] N. Nissim, A. Cohen, and Y. Elovici, "ALDOCX: detection of unknown malicious microsoft office documents using designated active learning methods based on new structural feature extraction methodology," IEEE Transactions on Information Forensics and Security, vol. 12, no. 3, pp. 631-646, Mar. 2016.

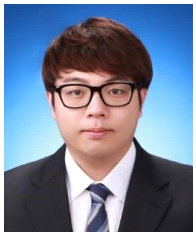
[9] S.W Kim, S. Hong, J. Oh, and H. Lee, "Obfuscated VBA macro detection using machine learning," 2018 48th annual ieee/ifip international conference on dependable systems and networks (dsn). pp. 490-501, Jun. 2018.

[10] V. Koutsokostas, N. Lykousas, T. Apostolopoulos, G. Orazi, A. Ghosal, F Casino, M. Conti, and C. Patsakis, "Invoice# 31415 attached: Automated analysis of malicious Microsoft Office

documents," Computers & Security vol. 114, no. 102582, pp. 1-13, Mar. 2022.
[11] S. Yang, W. Chen, S. Li, and Q. Xu, "Approach using transforming structural data into image for detection of malicious MS-DOC files based on deep learning models," 2019 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC). IEEE, pp. 28-32, Nov. 2019.
[12] H. Park and A.R. Kang, "MS Office Malicious Document Detection Based on CNN," Journal of the Korea Institute of Information Security & Cryptology, vol. 32, no. 2, pp. 439-446, Apr. 2022.
[13] M. Mimura, "An improved method of detecting macro malware on an imbalanced dataset," IEEE Access vol. 8, pp. 204709-204717, Nov. 2020.
[14] Y.S. Jeong, M.E. Mswahili, and A.R. Kang, "File-level malware detection using byte streams," Sci Rep, vol. 13, no.1 pp. 8925-8931, Jun. 2023.
[15] Y.S Jeong, S.M Lee, J.H Kim, J Woo, and A.R Kang, "Malware detection using byte streams of different file formats," IEEE Access, vol 10, pp. 51041-51047, May. 2022.
[16] Y.S. Jeong, J. Woo, and A.R. Kang, "Malware Detection on Byte Streams of Hangul Word Processor Files," Applied Sciences, vol. 9, no. 23, pp. 1-13, Nov. 2019.
[17] L. Holmstrom and P. Koistinen. "Using additive noise in back-propagation training," IEEE transactions on neural networks, vol. 3, no. 1 pp. 24-38, Jan. 1992
[18] H. Zhang, M. Cisse, Y.N Dauphin, and D. Lopez-Paz, "mixup: Beyond Empirical Risk Minimization," ICLR (Poster), pp. 1-13, Apr. 2018
[19] I. Goodfellow and J. Pouget-Abadie "Generative adversarial networks," Communications of the ACM, vol. 63, no. 11, pp. 139-144, Oct. 2020
[20] L. Taylor and G. Nitschke, "Improving deep learning with generic data augmentation," 2018 IEEE symposium series on computational intelligence (SSCI). IEEE, pp. 1542-1547, Nov. 2018.
[21] D.J Marchette and J.L Solka, "Using data images for outlier detection," Computational Statistics & Data Analysis vol. 43, no. 4, pp. 541-552, Jan. 2003
[22] R. Kannan, H. Woo, C.C Aggarwal, and H. Park, "Outlier detection for text data," Proceedings of the 2017 siam international conference on data mining. Society for Industrial and Applied Mathematics, pp. 489-497, Apr. 2017.
[23] S. Ho, A. Reddy, S. Venkatesan, R. Izmailov, R. Chadha, and A. Oprea, "Data Sanitization Approach to Mitigate Clean-Label Attacks Against Malware Detection Systems," MILCOM pp. 993-998, Nov. 2022.
[24] P. Porras and V. Shmatikov, "Large-scale collection and sanitization of network security data: risks and challenges," NSPW, pp. 57-64. Sep. 2006
[25] R.H.L Sim, X. Xu, and B.K.H Low, "Data valuation in machine learning:"ingredients", strategies, and open challenges," Proc. IJCAI. pp. 5607-5614, Jul. 2022.
[26] C. Courtnage and E. Smirnov, "Shapley-value data valuation for semi-supervised learning," Discovery Science: 24th International Conference, DS 2021, Halifax, NS,

Canada, October 11 - 13, 2021, Proceedings 24. Springer International Publishing, pp. 94-108, Oct, 2021.
〔27〕 I. Ashrapov, "Tabular GANs for uneven distribution," arXiv preprint arXiv:2010.00638, Oct. 2020.
〔28〕 L. Xu, and K. Veeramachaneni, "Synthesizing tabular data using generative adversarial networks," arXiv preprint arXiv:1811.11264, Nov. 2018

〈저자소개〉



손 진 혁 (JinHyuk Son) 정회원
2015년 2월: 충남대학교 컴퓨터공학과 (학사)
2017년 8월: 충남대학교 컴퓨터공학과 (석사)
2022년 8월: 충남대학교 컴퓨터공학과 (박사수료)
2022년~현재: 한국과학기술원 사이버보안연구센터 연구
〈관심분야〉 AI, 설명가능한 인공지능, 데이터 과학, 사이버보안

고 기 혁 (Gihyuk Ko) 정회원
2012년: 서울대학교 전기정보공학(학사)
2015년: Carnegie Mellon University, 컴퓨터공학 (석사)
2018년: Carnegie Mellon University, 컴퓨터공학 (박사수료)
2019년~현재: 한국과학기술원 사이버보안센터 선임연구원/팀장
〈관심분야〉 정형방법론, AI 보안 및 프라이버시, 설명가능한 인공지능

조 호 묵 (Ho-Mook Cho) 종신회원
2006년: 아주대학교 정보통신공학과 정보보호학 (공학석사)
2018년: 전남대학교 정보보안협동과정 (이학박사)
2014년~현재: 한국과학기술원 사이버보안연구센터 책임연구원/실장
〈관심분야〉 사이버보안, 악성코드 분석, XAI

김 영 국 (Young-Kuk Kim) 정회원
1985년: 서울대학교 계산통계학과 학사
1987년: 서울대학교 계산통계학과 석사
1995년: 미국 버지니아대학 컴퓨터과학과 박사
1995년~1996년: 핀란드 VTT연구소, 노르웨이 SINTEF연구소 방문연구원
2002년~2003년: 미국 UC Davis 방문교수
1996년~현재: 충남대학교 컴퓨터공학과 교수
〈관심분야〉 스마트정보시스템, 상황인지 추천시스템, 바이오AI융합, 산업AI응용