Journal of Korea Multimedia Society Vol. 23, No. 5, May 2020(pp. 641-649)
https://doi.org/10.9717/kmms.2020.23.5.641

LSTM 및 정보이득 기반의 악성 안드로이드 앱 탐지연구

안유림[†], 홍승아[††], 김지연[†††], 최은정[††††]

A Study on Detection of Malicious Android Apps based on LSTM and Information Gain

Yulim Ahn[†], Seungah Hong[††], Jiyeon Kim[†††], Eunjung Choi[††††]

ABSTRACT

As the usage of mobile devices extremely increases, malicious mobile apps(applications) that target mobile users are also increasing. It is challenging to detect these malicious apps using traditional malware detection techniques due to intelligence of today's attack mechanisms. Deep learning (DL) is an alternative technique of traditional signature and rule-based anomaly detection techniques and thus have actively been used in numerous recent studies on malware detection. In order to develop DL-based defense mechanisms against intelligent malicious apps, feeding recent datasets into DL models is important. In this paper, we develop a DL-based model for detecting intelligent malicious apps using KU-CISC 2018-Android, the most up-to-date dataset consisting of benign and malicious Android apps. This dataset has hardly been addressed in other studies so far. We extract OPcode sequences from the Android apps and preprocess the OPcode sequences using an N-gram model. We then feed the preprocessed data into LSTM and apply the concept of Information Gain to improve performance of detecting malicious apps. Furthermore, we evaluate our model with numerous scenarios in order to verify the model's design and performance.

Key words: Mobile Malicious Apps, Android Malware, Deep Learning, Long Short-term Memory, Information Gain, Shannon Entropy

1. 서 론

모바일 앱 인텔리전스 플랫폼 앱 애니(App Annie)에서는 2019년 모바일 현황 보고서를 통해 2018년 전 세계 앱의 총 다운로드 횟수를 1,940억 건으로 발표했다. 다양하고 유용한 앱이 많이 등장하면서 사용자들의 편의성이 증가하였지만, 모바일 앱을 악성코드 감염 경로로 악용하는 해커또한 증가하면서 사용자들의 모바일 단말 및 정보 자산이 위협에 노출되고 있다. 맥아피(MacAfee)의 2019년 사이버보안 동향 예측 발표에 따르면 보안기술이 향상될수록 보안기술을 우회하기 위한 공격기술 역시 진화되기 때문에 모바일 앱 사용자들이 점점 더 심각한 보안 위협에 노출 것으로 예상된다. 그러나 전통적인 서명(sig-

※ Corresponding Author : Eunjung Choi, Address: (01797) Hwarang-ro 621, Nowon-gu, Seoul, South Korea, TEL : +82-2-970-5338, FAX : +82-2-970-5981, E-mail : chej@swu.ac.kr
Receipt date : Mar. 31, 2020, Revision date : Apr. 27, 2020
Approval date : Apr. 28, 2020
[†] Dept. of Information Security, Seoul Women's University (E-mail : ahnyulim@swu.ac.kr)
[††] Dept. of Information Security, Seoul Women's University (E-mail : ghdtmddk1516@swu.ac.kr)
[†††] Center for Software Educational Innovation, Right AI with Security & Ethics Research Center, Seoul Women's University (E-mail : jykim07@swu.ac.kr)
[††††] Dept. of Information Security, Right AI with Security & Ethics Research Center, Seoul Women's University
※ This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (NRF-2018R1D1A1B07050543). This work was supported by a research grant from Seoul Women's University(2020-0183).

nature) 기반의 악성 앱 탐지 기술이나 규칙 기반의 이상(anomaly) 탐지 기술만으로는 진화하는 신/변종 악성 앱에 대응하는 데에 많은 한계가 존재한다. 머신러닝 및 딥러닝은 이러한 전통적인 정보보호 기술의 한계를 극복하게 하는 대안기술로서 악성코드 연구를 포함하여 다양한 정보보호 분야 연구에 활발히 도입되고 있다[1].

최근 많은 연구들이 머신러닝 및 딥러닝 기반의 악성코드 연구를 수행하고 있지만, 지능형 악성 앱 탐지에서는 머신러닝보다 딥러닝 모델이 더 효과적인 것으로 알려진다[2,3]. 기존의 딥러닝 기반 악성 앱 탐지 연구에서는 Drebin 데이터셋과 같이 상대적으로 오래된 데이터셋을 활용한 연구가 많이 존재한다[4-9]. 생성된 지 오래된 데이터셋은 최신 악성 앱의 공격특징을 잘 반영하지 못하기 때문에 이를 기반으로 탐지 모델을 개발할 경우, 최신 악성 앱 탐지에 효과적이지 못할 것이다. 본 논문에서는 최신 악성 앱의 공격 특징을 포함하고 있는 KU-CISC2018-Android 데이터셋을 활용하여 딥러닝 기반의 악성 앱 탐지 모델을 개발하고자 한다. 딥러닝 모델 개발 시에는 특징 추출(feature selection) 기법, 데이터 전처리(pre-processing) 기법, 그리고 적용하는 딥러닝 모델 유형에 따라 모델의 성능에 차이가 발생한다. 따라서 본 연구에서는 이러한 요소들을 모두 고려하여 모델을 설계하고자 한다. 먼저 특징 추출기법으로서 실행 파일의 명령어 중 연산자에 해당하는 OPcode (operation code) 시퀀스를 추출하고, 이를 N-gram 기반으로 전처리하여 딥러닝 모델에 입력데이터로 제공한다. 딥러닝 모델로는 대표적인 딥러닝 모델 비교를 통해 가장 탐지 성능이 좋은 LSTM(Long Short-Term Memory) 모델을 최종적으로 채택하였다. 또한, 섀넌 엔트로피 개념을 적용하여 정보이득(Information Gain)을 계산하고, 이를 LSTM 모델 결과에 반영하여 탐지 성능을 높이는 방법을 제안한다. 특히, 다양한 실험을 통한 탐지 정확성 분석을 통해 제안된 모델의 설계방법 및 성능을 검증한다.

본 논문의 구성은 다음과 같다. 2장에서는 제안된 모델의 배경이론 및 관련연구를 제시하고, 3장에서는 LSTM 및 정보이득 기반의 악성 안드로이드 앱 탐지 모델을 설계한다. 4장에서는 다양한 실험 시나리오를 개발하여 제안된 모델의 성능을 검증하고, 5장에서 결론 및 향후 연구를 제시한다.

2. 배경이론 및 관련연구

2.1 OPcode 시퀀스 추출

실행파일은 OPcode 및 주소(operand) 정보로 이루어진 명령어 집합으로 구성된다. 악성코드들은 일련의 행위를 통해 감염 목표를 달성하기 때문에 OPcode 시퀀스 추출을 통해 공격의 행위를 파악할 수 있다. 많은 연구들이 OPcode 시퀀스를 이용하여 머신러닝 및 딥러닝 기반의 악성코드 연구를 수행하고 있으며 많은 논문에서 OPcode 시퀀스 추출에 대한 타당성이 검증되었다.

악성코드의 특징 추출에 OPcode 시퀀스를 활용한 연구로는 Genome 프로젝트 데이터셋 및 McAfee 데이터셋의 OPcode를 추출한 후, CNN(Convolutional Neural Network) 모델 기반으로 악성 앱을 탐지하는 연구 [10], Microsoft 사에서 제공한 데이터셋에서 OPcode 시퀀스를 추출한 후 임베딩 기법 및 LSTM 모델을 적용하는 연구 [11], 그리고 Microsoft 데이터셋의 OPcode 추출 후, LSTM 기반으로 분류하는 스택 앙상블 모델 연구[12] 등이 존재한다.

2.2 N-gram

N-gram이란, 다음 항목을 예측하기 위한 대표적인 확률적 언어 모델의 한 방법으로 문자열이나 음성 샘플에서 n개 항목의 연속적인 시퀀스를 나타내는 것이다. 본 논문에서는 추출한 OPcode 시퀀스에 N-gram을 적용시켜 악성 앱의 특징을 전처리한다. 악성 앱 탐지에 N-gram을 적용한 연구로는 Genome 데이터셋에 대해 1-gram부터 10-gram까지 OPcode sequence를 추출하여 빈도를 측정한 후, Naïve Bayes, PART(partial decision tree), SVM(Support Vector Machine), Random Forest 모델을 적용하여 악성 앱을 탐지하는 연구[13], 3-gram OPcode sequence와 PE(Portable Executable) 헤더 정보를 추출하여 악성코드의 다형성과 변종을 탐지하고 SVM 기반으로 분류하는 연구[14] 등이 수행되었다.

2.3 RNN (Recurrent Neural Networks)

RNN은 시계열 데이터의 학습을 위한 대표적인 딥러닝 모델이다. 일반적인 인공신경망은 Feed-forward neural networks (FFNets)라고 하는데 입력

테이터가 모든 은닉층을 단 한 번씩만 거쳐가는 모델이다. 즉, FFNets는 입력 테이터의 시간적인 순서를 고려하지 않는 구조라고 한다면, RNN의 경우는 은닉층의 결과가 같은 은닉층의 입력으로 다시 들어가는 구조로 설계되어 시간 순서를 고려해야하는 시계열 데이터를 처리하는데 효과적이다. 그러나 RNN은 단 하나의 tanh 혹은 ReLU 활성화 함수를 가진 구조이기 때문에 입력된 데이터와 참고해야 할 데이터를 사용하는 지점 사이의 거리 차가 커지면 데이터들의 연관성이 떨어져 학습능력이 저하되는 'Vanishing Gradient Problem'이 발생한다.

RNN을 적용한 악성코드 연구로는 Genome 데이터셋에 대해 정적 분석 및 동적 분석을 실시한 후, RNN과 LSTM 기반으로 모델을 개발하는 연구 [15], Drebin 데이터셋과 AndroZoo 데이터셋에 대해 LSTM, DNN, RNN 모델들을 permission 결과를 기반으로 비교하는 연구[16], APK Opera Mobile Store의 데이터셋을 2개 계층으로 구성된 RNN 및 LSTM 모델로 생성하여 성능을 비교한 연구[17] 등이 수행되었다.

2.4 LSTM (Long Short-Term Memory)

LSTM은 RNN에서 나타나는 'Vanishing Gradient Problem'을 극복하기 위해서 고안된 모델이다. RNN은 2.3에서 설명한 바와 같이 단일 활성화 함수를 가진 구조인 반면 LSTM은 상호작용을 하는 4개 계층이 존재한다. LSTM에는 장단기 기억이 모두 존재하며 장기 기억을 가능하게 하는 'cell state'에 의해 정보는 끊임없이 다음 단계에 전달된다.

LSTM을 적용한 악성코드 연구로는 악성코드의 시스템 콜(system call) 시퀀스를 추출하고, LSTM 언어 모델을 기반으로 악성코드 분류 모델을 제시한 연구[3], 악성 파일에서 OPcode 시퀀스를 추출할 때 불필요한 데이터를 제거하기 위해 MBDM(Method Block Denoise Module)을 적용한 LSTM 기반 HDN (Hierarchical Denoise Network) 분류 모델을 개발한 연구[18], Drebin 데이터셋과 AndroZoo 데이터셋을 CNN 및 LSTM 모델로 생성하고 성능을 비교한 연구[19] 등이 존재한다.

2.5 Shannon Entropy

섀넌 엔트로피(Shannon Entropy)는 주어진 데이터 집합의 '정보의 무질서도' 혹은 '정보의 불확실성'을 의미한다. 섀넌 엔트로피 수치는 0에서 1사이의 값을 가지며, 가장 혼잡도가 낮은 수치는 0이고 혼잡도가 높은 수치는 1이다. 즉, 엔트로피가 0인 경우는 '질서정연한 상태'로써 모든 데이터는 한 곳에 집중되어 있으며, 엔트로피가 1인 경우에는 '완벽하게 무질서한 상태'로써 데이터가 흩어져 있음을 의미한다. 엔트로피의 수식은 아래 (1)과 같다.

$$H(P) = H(x) = -\sum_{x} P(x)\log P(x) \quad (1)$$

섀넌 엔트로피를 적용한 악성코드 연구로는 Drebin 데이터셋과 Google Play의 안드로이드 앱에 대해 Hidden Markov Model과 엔트로피를 이용해 특징을 추출하고 분류하는 모델을 제안한 연구[20], VX heavens 데이터셋과 Offensive computing 데이터셋에 대해 엔트로피의 SAX(Symbolic Aggregate Approximation)를 적용하고, Naïve Bayes 및 SVM을 기반으로 패킹된 악성코드를 탐지하는 연구가 존재한다[21].

2.6 IG(Information Gain)

정보이득은 데이터 샘플을 구성하는 다양한 속성 중, 특정 속성을 기준으로 데이터를 분류할 때 감소하는 엔트로피의 양으로서 어떤 속성이 데이터셋 분류에 높은 영향을 미치는지를 파악할 수 있는 지표로 활용될 수 있다. 정보이득의 계산식은 (2)와 같다.

$$IG(A,S) = H(S) - \sum_{t \in T} p(t)H(t) \quad (2)$$

IG를 적용한 악성코드 연구로는 AndroidManifest.xml에서 가져온 안드로이드 앱에 대해 정보 이득을 계산하여 상위에 속하는 특징들을 도출하고, 이 특징들을 활용하여 Naïve Bayes, Random Forest, Decision Table 등 다양한 머신러닝 알고리즘 기반으로 악성코드를 탐지하는 연구[22], Contagio 프로젝트 데이터셋과 구글 스토어의 안드로이드 앱에서 시스템 호출에 대한 정보이득을 계산하고, 이 중 높은 정보이득을 갖는 시스템 호출에 대해 다양한 머신러닝 및 딥러닝 기반으로 악성코드를 분류하는 연구[23] 등이 존재한다.

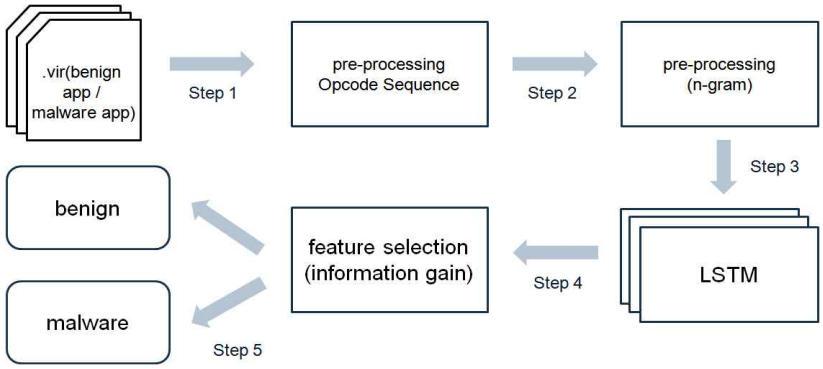

Fig. 1. A process for detecting malicious code based on LSTM and Information Gain.

Binary	030414	0008	0008	004809	99
Opcode	03	00	00	00	99

	label	
0	malware	03 00 00 00 8a 00 00 00 00 14
1	malware	03 00 00 00 b0 43 00 00 00 00

Fig. 2. Extraction of OPcode sequence from an Android malicious application.

3. 제안 모델

본 논문은 '2018 정보보호 R&D 데이터 챌린지 대회'에서 제공된 KU-CISC2018-Android 데이터셋을 활용하여 제안 모델을 설계한다. 본 논문에서는 2,000개의 정상 및 악성 앱을 활용하였으며 모든 악성 앱은 'malwares.com' 및 'VirusTotal'에서 검증되었기 때문에 데이터셋 및 이를 활용한 제안 모델의 신뢰성을 확보할 수 있다. 본 논문에서 제안하는 LSTM 및 정보이득 기반의 안드로이드 악성앱 탐지 모델은 Fig. 1과 같이 5단계로 이루어진다.

- Step 1

안드로이드 악성 앱의 특징을 추출하기 위하여 OPcode 시퀀스를 추출하는 단계이다. Fig. 2는 KU-CISC2018-Android 데이터셋에서 추출한 OPcode 시퀀스 추출 결과 예를 보여준다.

- Step 2

추출된 OPcode 시퀀스에 존재하는 문자열의 빈도수를 이용하여 다음에 등장할 문자열을 예측하는 확률 모델로서 문자열 형태의 데이터를 분석할 때 주로 사용되는 'N-gram'기법을 적용하여 전처리하는 단계이다. Fig. 3은 2-gram을 적용하여 추출된 OPcode를 전처리하는 예를 보여준다.

N은 trade-off로 정할 수 있는데, 1보다는 2, 그리고 3으로 선택하는 것이 특징 추출에 효과적이다. 3 이상의 N을 선택할 경우에는 실제 데이터 훈련 시 해당 특징에 대한 희소 문제가 발생할 뿐 아니라, 모델 사이즈 또한 커지는 문제점 또한 존재하므로 본 논문에서는 3-gram을 적용하여 전처리를 수행하였다.

- Step 3

본 논문에서는 전처리한 데이터를 LSTM에 학습시키기 위해 Keras기반으로 LSTM모델을 구현하였다. LSTM은 다양한 계층유형을 고려하여 설계될 수 있는데 본 논문에서는 각각 1개의 입력계층, 임베딩 계층, 완전연결계층으로 구성된 모델을 설계하였다. 임베딩 계층은 3-gram으로 이미 변환된 OPcode 시퀀스에 대해 16개의 뉴런으로 출력되도록 설계하였고, 완전연결계층은 16개의 입력 뉴런에 대해 1개의 출력뉴런을 갖도록 설계하였다. 활성화 함수는 이진 분류 문제에서 출력 층에 주로 사용되는 sigmoid를

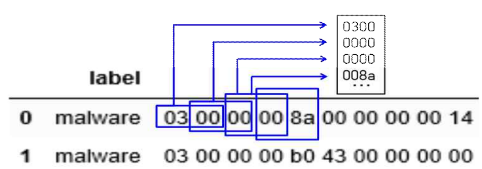

Fig. 3. Applying N-gram to extracted OPcode 시퀀스.

```
000000,0.3384586822061518
040000,0.26422613009263374
000d00,0.2551228238525792
0d0000,0.24397822430786306
444154,0.2390885369967764
820400,0.2323423308480256
426082,0.2323423308480256
608204,0.23223524516221317
080000,0.2321978364837399
ae4260,0.2321283887637827
```

Fig. 4. Top 10 OPcode sequence with high information gain.

사용하고, 학습을 위한 가중치 최적화 기법으로는 RMSProp 기법을 사용하였다. 또한, 학습과정에서의 손실함수로서 교차 엔트로피를 사용하여 모델을 개발하였다.

• Step 4

LSTM을 활용하여 학습이 완료된 후, 정확도 향상을 위해 정보이득 개념을 반영하여 악성앱에 대한 탐지를 추가로 수행하는 단계이다. 이 단계에서는 3-gram으로 전처리된 OPcode 시퀀스 중, 이진 분류에 많은 영향을 미치는 OPcode 시퀀스를 파악하기 위하여 정보이득을 계산하고, 이 중 상위에 속하는 OPcode 시퀀스에 대하여 각각 악성 및 정상 앱 중 어떤 분류에 가까운 특징을 가지는 지를 파악한다. Fig. 4는 상위에 분포하는 OPcode 시퀀스 정보이득 계산 결과를 보여준다.

• Step 5

정보이득 계산을 통해 악성앱 및 정상 앱 탐지에 큰 영향을 미치는 OPcode 시퀀스가 도출되면, 이를 LSTM 기반 악성앱 탐지 결과에 추가하여 반영하여 최종적으로 악성앱을 탐지하는 단계이다. 악성앱 탐지에 정보이득 계산 결과를 추가적으로 반영함으로써 LSTM 기반 악성앱 탐지 정확도가 향상되는 효과가 기대할 수 있다.

4. 실험 결과 및 고찰

4.1 실험 시나리오

3장에서 제안한 모델의 성능을 검증하기 위하여 본 논문에서는 4개의 실험시나리오를 개발하여 KU-CISC2018-Android에 대한 악성 앱 탐지를 수행하였다. 첫 번째 시나리오는 OPcode 시퀀스 추출의 효과를 분석하기 위한 실험으로, OPcode 시퀀스를 추출 여부에 따른 악성앱 탐지 정확도를 측정한다.

두 번째 시나리오는 딥러닝 모델의 유형에 따른 정확도를 비교하기 위한 실험으로서, CNN, RNN, LSTM 기반의 악성앱 탐지 결과를 측정하고, 본 논문에서 제시하는 LSTM 기반 모델의 타당성을 분석하고자 한다. 세 번째 시나리오는 'N-gram' 기반의 전처리 효과를 분석하기 위한 실험으로서 전처리를 하지 않았을 때(1-gram)와 3-gram 기반으로 전처리를 했을 때의 정확도를 비교하여 제안된 모델의 전처리기술의 타당성을 제시하고자 한다. 마지막 시나리오는 정보이득의 적용에 대한 효과를 분석하기 위한 목적으로 LSTM 기반의 악성앱 탐지 결과 대비 LSTM 및 정보이득 조합을 통한 악성앱 탐지 결과를 비교한다.

4.2 실험 결과 및 분석

4.1의 네 가지 시나리오별로 악성앱 탐지 성능을 측정하기 위하여 본 연구에서는 f1-score를 기반으로 정확도를 측정한다. f1-score는 재현율(Recall)과 정밀도(Precision)를 이용하여 조화 평균(harmonic mean)을 구한 값이다. 재현율은 입력된 데이터에 대해 모델이 데이터를 어떤 클래스로 예측을 하는지에 대한 척도이며, 정밀도는 예측한 값에 대해 정확성을 판단하는 척도이다. f1-score를 측정하기 위한 수식은 (3)과 같으며 수식에 기술된 TP(True Positive)는 악성앱을 악성앱으로 판단한 샘플 수, FP(False Positive)는 정상 앱을 악성앱으로 판단한 샘플 수, FN(False Negative)는 악성앱을 악성앱으로 판단한 샘플 수, TN(True Negative)는 정상 앱을 악성앱으로 판단한 샘플 수를 의미한다.

$$f1-score = \frac{2 \times precision \times recall}{precision + recall} \quad (3)$$

단, $precision = \frac{TP}{TP+FP}$, $recall = \frac{TP}{FN+TP}$

각 시나리오별 악성앱 탐지 정확도는 Fig. 5와 같다.

• *시나리오 1:* OPcode 시퀀스 추출 여부에 따른 탐지 정확도 분석 (Fig. 5(a))

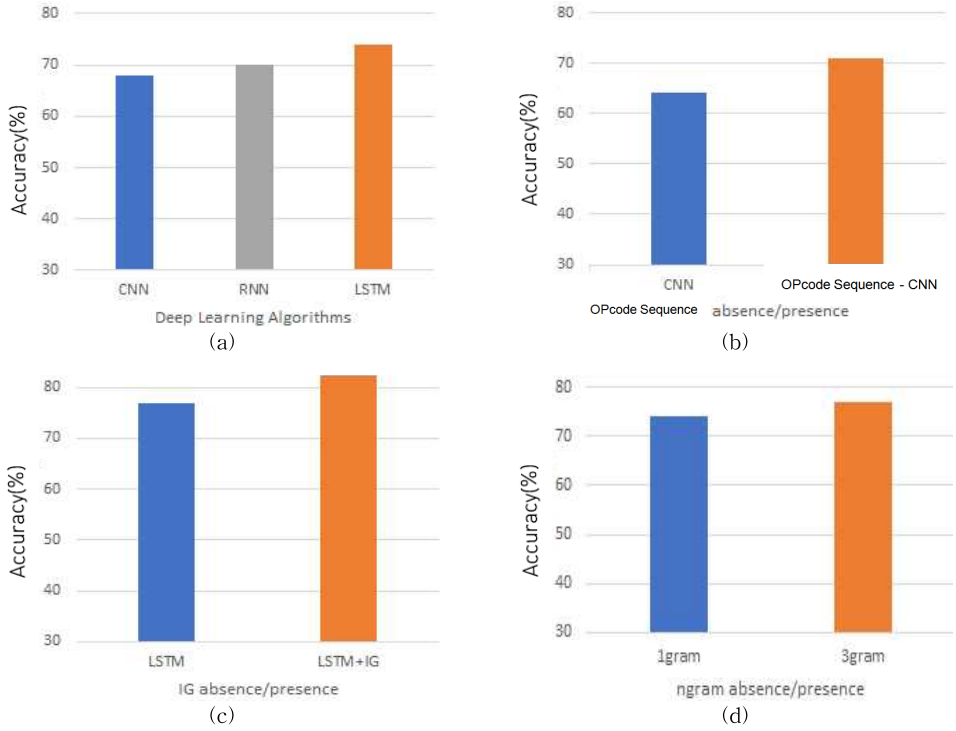

Fig. 5. Experimental Evaluation with four types of experimental scenarios – (a) Feature Selection (b) Deep Learning Models (c) N-gram based data preprocessing (d) Information Gain

첫 번째 실험인 OPcode 시퀀스 추출 여부에 따른 악성앱 탐지 분석 결과, OPcode 시퀀스를 추출하여 CNN에 적용시켰을 때가 71%, OPcode 시퀀스를 추출하지 않고 CNN에 적용시켰을 때가 64%로 OPcode 시퀀스를 추출하였을 때가 추출하지 않았을 때보다 약 7% 더 높은 결과를 보이는 것을 알 수 있다. 즉, OPcode 시퀀스를 추출할 때의 정확도가 추출하지 않았을 때보다 성능이 높으므로 제안된 모델의 설계 단계 중, OPcode 시퀀스 추출이 효과적임을 증명할 수 있다.

• *시나리오 2:* 딥러닝 모델 유형에 따른 탐지 정확도 분석 (Fig. 5(b))

딥러닝의 대표적인 모델 CNN, RNN, LSTM을 시나리오 1에서 좋은 성능을 보인 OPcode 시퀀스 추출 데이터를 활용하여 학습한 결과, CNN이 68%, RNN이 70%, LSTM이 74%로, 본 논문에서 활용한 LSTM 모델이 CNN보다 약 6%, RNN보다 약 4% 높은 정확도를 보이는 것을 알 수 있다.

• *시나리오 3:* N-gram 기반 데이터 전처리 유무에 따른 탐지 정확도 분석 (Fig. 5(c))

시나리오 2에서 좋은 성능을 보인 LSTM 기반 모델을 활용하여 OPcode 시퀀스의 3-gram 기반 전처리 유무에 따른 정확도를 분석한 결과, 전처리를 수행한 실험결과의 성능이 더 높게 나온 것을 확인할 수 있다. 전처리를 수행하지 않았을 때(1-gram)의 정확도가 약 74%, 3-gram 기반으로 전처리를 수행했을 때의 정확도가 약 77%로 3% 높은 정확도를 보이는 것을 보아 제안된 모델의 3-gram 기반 전처리 단계가 모델의 성능 향상에 효과적임을 검증할 수 있다.

• *시나리오 4: 정보이득 적용* 유무에 따른 탐지 정확도 분석 (Fig. 5(d))

시나리오 1부터 3까지 좋은 성능을 보인 조건을 모두 반영하여 LSTM 기반으로 학습한 결과 77%의 악성 앱 탐지 정확도를 보이는 것을 시나리오 3에서 확인할 수 있다. 시나리오 4에서는 여기에 추가하여 정보이득 계산 결과를 반영할 때와 반영하지 않을 때의 정확도를 측정하였다. Fig. 5(d)와 같이 정보이

득을 적용했을 때 탐지 정확도가 82.3%까지 향상된 것으로 보아 제안된 모델의 정보이득 적용 단계가 모델의 성능을 위해 효과적임을 검증할 수 있다.

5. 결 론

본 논문은 최신 악성 앱의 공격 특징을 포함하는 KU-CISC2018-Android 데이터셋을 활용하여 LSTM 및 정보이득 개념을 조합하여 설계한 악성앱 탐지 모델을 제안하고 있다. 제안된 모델은 악성 앱 탐지를 위한 OPcode 시퀀스 추출단계, 3-gram 기반 전처리 단계, LSTM 기반의 악성앱 탐지 단계, 정보이득 계산을 통한 OPcode 시퀀스의 가중치 분석단계, 그리고 LSTM 및 정보이득을 조합하여 악성앱을 최종적으로 탐지하는 단계로 구성된다. 본 논문에서 사용한 데이터셋을 활용한 국내외 기존 연구가 존재하지 않아 제안된 모델의 성능을 다른 연구와의 비교를 통해 제시하는 것은 어렵지만, 본 논문에서는 제안된 모델의 설계 단계별로 성능을 측정함으로써 모델 설계의 타당성 및 우수성을 검증하고자 하였다.

먼저 각 단계별 성능을 비교할 수 있는 4개의 시나리오를 개발하고, 각 시나리오를 제안된 모델 기반으로 실험한 결과, OPcode 시퀀스 추출의 타당성, 3-gram 기반 전처리의 타당성, 다양한 딥러닝 모델 중 LSTM 모델 적용의 타당성, 그리고 정보이득 개념 적용의 타당성을 객관적인 정확성 비교를 통해 증명하였다. 본 연구는 최신 지능형 악성 앱 탐지의 딥러닝 기반 모델을 선도적으로 제안하는 데에 의의가 있으며 향후에는 OPcode 시퀀스 및 N-gram 외의 다양한 특징 추출 및 전처리 기법을 고려하여 모델을 개선하는 연구를 수행할 예정이다. 또한, 제안된 모델을 최신 안드로이드 데이터셋에 지속적으로 적용함으로써 모델의 성능을 개선할 것이다.

REFERENCE

[1] Jin-Gul Joo, In-Seon Jeong, and Seung-Ho Kang, “An Optimal Feature Selection Method to Detect Malwares in Real Time Using Machine Learning,” *Journal of Korea Multimedia Society*, Vol. 22, No. 2, pp. 203-209, 2019.

[2] M.K. Alzaylaee, S.Y. Yerima, and S. Sezer, “DL-Droid: Deep Learning Based Android Malware Detection Using Real Devices,” *Computers and Security*, Vol. 89, No. 101663, pp. 1-11, 2020.

[3] Z. Yuan, Y. Lu, Z. Wang, and Y. Xue, “Droid-Sec: Deep Learning in Android Malware Detection,” *ACM Special Interest Group on Data Communication Computer Communication Review*, Vol. 44, No. 4, pp. 371-372, 2014.

[4] A.Y, Saleh and C. Francis, “A Deep Learning Approach to Malware Detection in Android Platform,” *International Journal of Innovative Technology and Exploring Engineering*, Vol. 8, No. 8, pp. 1043-1048, 2019.

[5] X. Xiao, S. Zhang, and F. Mercaldo, “Android Malware Detection Based on System Call Sequences and LSTM,” *Multimedia Tool and Applications*, Vol. 78, No. 4, pp. 3979-3999, 2019.

[6] L. Shiqi, L. Zhiyuan, N. Bo, W. Huanhuan, S. Hua, and Y. Yong, “Android Malware Analysis and Detection Based on Attention-CNN-LSTM,” *Journal of Computers*, Vol. 14, No. 1, pp. 31-43, 2019.

[7] S. Vanjire and M. Lakshmi, “FNN and Auto Encoder Deep Learning-based Algorithm for Android Cyber Security,” *International Journal of Recent Technology and Engineering*, Vol. 8, No. 5, pp. 3292-3296, 2020.

[8] A. Naway and Y. Li, “Using Deep Neural Network for Android Malware Detection,” *International Journal of Advanced Studies in Computer Science and Engineering*, Vol. 7, No. 12, pp. 9-18, 2018.

[9] K. Xu, Y. Li, R.H. Deng, and K. Chen, “Deep Refiner: Multi-layer Android Malware Detection System Applying Deep Neural Networks,” *Proceeding of IEEE European Symposium on Security and Privacy*, pp. 473-487, 2018.

[10] N. McLaughlin, J.M.d. Rincon, B.J. Kang, S. Yerima, P. Miller, and S. Sezer, “Deep Android Malware Detection,” *Proceedings of the*

Seventh ACM on Conference on Data and Application Security and Privacy, pp. 301-308, 2017.

[11] R. Lu, *Malware Detection with LSTM Using Opcode Language*, University of Chinese Academy of Sciences, Beijing, 2019.

[12] J. Yan, Y. Qi, and Q. Rao, "Detecting Malware with an Ensemble Method Based on Deep Neural Network," *Security and Communication Networks*, Vol. 2018, No. 7247095, pp. 1-16, 2018.

[13] B. Kang, S.Y. Yerima, S. Sezer, and K. McLaughlin, "N-gram Opcode Analysis for Android Malware Detection," *International Journal on Cyber Situational Awareness*, Vol. 1, No. 1, pp. 231-255, 2016.

[14] A.I. Elkhawas and N. Abdelbaki, "Malware Detection Using Opcode Trigram Sequence with SVM," *Proceeding of International Conference on Software, Telecommunications and Computer Networks*, pp 1-6, 2018.

[15] R. Vinayakumar and K.P. Soman, "Detecting Android Malware Using Long Short-term Memory (LSTM)," *Journal of Intelligent and Fuzzy Systems*, Vol. 34, No. 3, pp. 1277-1288, 2018.

[16] H. Alimardani and M. Nazeh, "Permission-based Analysis of Android Applications Using Categorization and Deep Learning Scheme," *Proceeding of MATEC Web of Conferences 2018,* pp. 1-7, 2019.

[17] R. Vinayakumar and K. P. Soman, "Deep Android Malware Detection and Classification," *Proceeding of 2017 International Conference on Advances in Computing, Communications and Informatics*, pp. 1677-1683, 2017.

[18] J. Yan, Y. Oi, and Q. Rao, "LSTM-based Hierarchical Denoising Network for Android Malware Detection," *Security and Communication Networks*, Vol. 2018, No. 5249190, pp. 1-18, 2018.

[19] A. Hota and P. Irolla, "Deep Neural Networks for Android Malware Detection," *Proceedings of the 5th International Conference on Information Systems Security and Privacy - Volume 1: ForSE*, pp. 657-663, 2019.

[20] G. Canfora, F. Mercaldo, and C.A. Visaggio, "An HMM and Structural Entropy Based Detector for Android Malware: An Empirical Study," *Computers and Security*, Vol. 61, pp. 1-18, 2016.

[21] M.B. Erdene, H. Park, H. Li, H. Lee, and M. S. Cho, "Entropy Analysis to Classify Unknown Packing Algorithms for Malware Detection," *International Journal of Information Security*, Vol. 16, No. 3, pp. 227-248, 2017.

[22] A. Bhattacharya and R.T. Goswami, "DMDAM: Data Mining Based Detection of Android Malware," *Proceedings of the First International Conference on Intelligent Computing and Communication,* pp. 187-194, 2016.

[23] L. Singh and M. Hofmann, "Dynamic Behavior Analysis of Android Applications for Malware Detection," *Proceeding of International Conference on Intelligent Communication and Computational Techniques*, pp. 1-7, 2017.

안 유 림

2018년 3월~현재 서울여자대학교 정보보호학과
관심분야 : 정보보호, 인공지능, 빅데이터

홍 승 아

2018년 3월~현재 서울여자대학교 정보보호학과
관심분야 : 정보보호, 인공지능, 빅데이터

김 지 연

2007년 2월 서울여자대학교 정보보호공학과(공학사)
2013년 8월 서울여자대학교 컴퓨터학과(이학박사)
2014년 3월~2017년 8월 Carnegie Mellon University 박사후연구원
2019년 3월~현재 서울여자대학교 소프트웨어교육혁신센터 전담교수
관심분야 : 네트워크 보안, 인공지능, 클라우드 보안, 사물인터넷보안

최 은 정

1997년 2월 서울여자대학교 컴퓨터학과(이학사)
2000년 2월 서울여자대학교 대학원 컴퓨터학과(이학석사)
2005년 8월 서울여자대학교 대학원 컴퓨터학과(이학박사)
2006년 3월~현재 서울여자대학교 정보보호학과 교수
관심분야 : 빅데이터분석, 인공지능, 악성코드