악성코드 탐지를 위한 동적 분석 데이터 전처리 기법

김해수[1], 김미희[1]★
[1]한경국립대학교 컴퓨터응용수학부, 컴퓨터시스템연구소
e-mail:{ww232330, mhkim}@hknu.ac.kr
★교신저자

Dynamic Analytic Data Preprocessing Techniques for Malware Detection

Hae-Soo Kim[1], Mi-Hui Kim[1]
[1]School of Computer Engineering & Applied Mathematics, Computer System Institute Hankyong National University

요 약

악성코드를 탐지하는 기법 중 동적 분석데이터와 같은 시계열 데이터는 프로그램마다 호출되는 API의 수가 모두 다르다. 하지만 딥러닝 모델을 통해 분석할 때는 모델의 입력이 되는 데이터의 크기가 모두 같아야 한다. 이에 본 논문은 TF-IDF(Term Frequency-Inverse Document Frequency)와 슬라이딩 윈도우 기법을 이용해 프로그램의 동적 특성을 유지하면서 데이터의 길이를 일정하게 만들 수 있는 전처리 기법과 LSTM(Long Short-Term Memory) 모델을 통해 정확도(Accuracy) 95.89%, 재현율(Recall) 97.08%, 정밀도(Precision) 95.9%, F1-score 96.48%를 달성했다.

1. 서론

악성코드를 탐지하는 기법 중 동적 분석데이터의 API call sequence와 같은 시계열 데이터 분석을 통한 탐지는 LSTM[1]과 같은 딥러닝 모델을 이용하는데 모델을 훈련하기 위해 입력되는 데이터의 길이는 모두 같아야 한다. 그러나 프로그램이 실행될 때 호출되는 API들의 순서를 보여주는 API call sequence는 프로그램마다 다르며 짧게는 10개 내외, 길게는 수백 만개의 API들을 호출한다. 이러한 데이터를 탐지 모델링에 이용하기 위해 텍스트 분류에서 주로 쓰이는 zero padding 기법[3]을 사용하거나 딥러닝 모델의 입력 크기를 길이가 가장 긴 데이터를 기준으로 지정하고 그 크기에 맞춰 전처리한다. 길이가 짧은 데이터의 경우 더미 데이터인 0을 추가하는 방식으로 전처리를 할 수 있다. 그러나 데이터 길이의 표준 편차가 커질수록 추가되는 더미 데이터가 많아져 딥러닝 모델은 제대로 훈련할 수 없다. 따라서 정상, 악성코드의 주요한 API 정보들을 유지하면서 고정된 크기로 데이터를 전처리하는 기법이 중요하다. 이에 본 논문에서는 TF-IDF[2]와 슬라이딩 윈도우 기법을 이용해 프로그램의 동적 특성을 유지하면서 데이터의 길이를 일정하게 만들 수 있는 전처리 기법을 제안한다.

2. 관련연구

[4]의 연구에서는 정적 분석 데이터와 API 카테고리 정보를 이미지로 생성하여 딥러닝 모델의 입력에 맞춰 크기를 조정하는 방식으로 데이터를 전처리하여 CNN 모델로 탐지하는 기법을 제안했다.

[5]의 연구에서 API2Vec 임베딩과 BiLSTM을 통해 서로 다른 길이의 데이터에서 정보의 차원을 축소하고 API의 기능적인 특성을 추출하고 각 API들의 오퍼레이션과 카테고리 페어에서 유사한 특성을 추출하고 추출된 정보에 가중치를 계산해서 Fully Connected Layer를 통해 악성의 유무를 판단하는 기법을 제안했다. 그러나 [4]의 경우 API 카테고리만을 이용해 같은 카테고리에 정상과 악성 프로그램에 쓰이는 API가 있을 수 있어 동적 정보를 제대로 반영하기 어렵고, [5]는 차원을 축소하고 특성을 추출하는 데에 여러 딥러닝 모델을 사용해 오버헤드가 크다. 따라서 오버헤드가 적고 API의 정보를 반영할 수 있는 전처리 기법이 필요하다.

3. 제안 기법

TF-IDF를 이용해 API call sequence에 대해 정상, 악성을 결정하는 주요한 요소는 높은 수치, 의미 없는 것은 낮은 수치로 매핑한다. 초기에 설정한 길

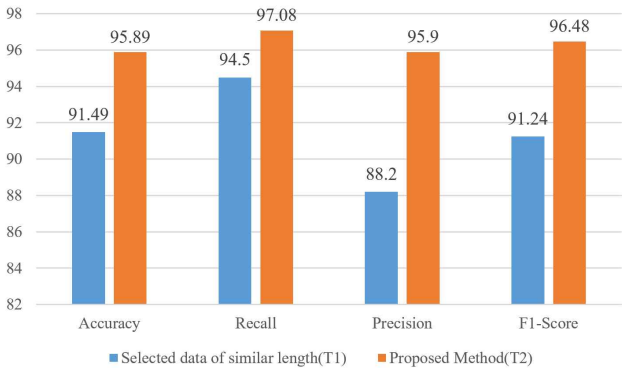
이가 N이라고 가정하고 API call sequence의 길이(S_{length})가 N보다 짧을 때는 0으로 패딩하고 길 때는 S_{length}를 N으로 나눠 w_s의 크기를 가진 윈도우를 생성한다.

$$\frac{S_{length}}{N} = w_s \quad \text{(a)}$$

생성된 윈도우는 w_s만큼 이동하면서 윈도우 내의 데이터의 평균을 계산하고 평균보다 큰 값이 더 많다면 가장 큰 값, 작은 값이 더 많다면 가장 작은 값 그리고 개수가 같으면 평균값이 그 윈도우의 대푯값이 된다. 모든 데이터에 대해 위 과정을 반복하면 총 N의 길이를 가진 데이터가 생성된다. 이를 통해 전처리 된 데이터는 LSTM 모델의 입력이 된다.

4. 실험 결과

실험 환경은 Intel Xeon(R) Silver 4215R CPU @ 3.20GHz CPU, 256GB RAM, NVIDIA RTX A6000 GPU 환경에서 진행하고 파이썬 3.7.13, 텐서플로우 2.7.0에서 실험을 진행했다. 실험에 사용한 데이터는 Practical Security Analytics에서 보안 및 AI 연구 목적으로 제공된 Raw PE 파일[6] 이다. 사용된 데이터는 총 16,590개 정상 9,756개 악성 6,834개이며 훈련, 테스트 비율은 8:2이다.



(그림 1) Comparison experiment results table with test data

그림 1은 실험 결과를 보여주는 표이다. 비슷한 길이의 API call sequence를 선택해서 패딩과 슬라이싱을 한 데이터(T1)와 제안 기법을 이용하여 전처리한 데이터(T1)를 LSTM 모델을 통해 비교한 결과이다. T1에 비해 T2의 정확도(Accuracy) 4.4%, 재현율(Recall) 2.58%, 정밀도(Precision) 7.7%, F1-score 5.24% 증가했다.

5. 결론

본 논문에서는 TF-IDF와 슬라이딩 윈도우 기법을 이용해 데이터의 길이를 일정하게 만들 수 있는 전처리 기법을 제안하였다. 해당 기법을 통해 프로그램의 특성을 유지하여, 실험을 통해 모든 데이터를 딥러닝 모델 훈련에 더 높은 성능을 위해 이용할 수 있음을 보였다. 향후 연구에서는 전처리 기법을 개선하여 정확도를 높이고 다양한 기법들과의 비교 성능 평가를 수행하고자 한다.

참고문헌

[1] S. Hochreiter and J. Schmidhuber, “LONG SHORT-TERM MEMORY”, Neural Computation, vol.9, no.8, pp. 1735-1780, 1997.

[2] J. Ramos, "Using tf-idf to determine word relevance in document queries", Proceedings of the first instructional conference on machine learning, vol.242, no.1, pp. 29-48, 2003.

[3] B. Hu, Z. Lu, H. Li and Q. Chen, "Convolutional neural network architectures for matching natural language sentences", Advances in Neural Information Processing Systems, pp. 2042-2050, 2014.

[4] X. Huang, L. Ma, W. Yang and Y. Zhong, “A Method for Windows Malware Detection Based on Deep Learning,” Journal of Signal Processing Systems, vol.93, pp.265-273, 2021.

[5] S. Zhang, J. Wu, M. Zhang, and W. Yang, “Dynamic Malware Analysis Based on API Sequence Semantic Fusion”, Applied Sciences, vol.13, no.11, pp.6526, 2023

[6] Practical Security Analytics, “PE Malware Machine Learning Dataset”, [Internet], https://practicalsecurityanalytics.com/pe-malware-machine-learning-dataset/