사전학습 모델을 활용한 효과적인 Http Payload 이상 탐지 방법

이웅기 [1], 김원철 [1]
[1] 롯데정보통신 R&D 센터
Unggi.lee@lotte.net, wonchul_kim@lotte.net

Effective Payload-based Anomaly Detection Method Using Pre-trained Model

Unggi LEE[1], Wonchul KIM[1]
[1] R&D Center, Lotte Data Communication

요 약

딥러닝 기반의 인공지능 기술이 발달함에 따라 이상 탐지 방법에도 딥러닝이 적용되었다. 네트워크 트래픽으로부터 요약 및 집계된 Feature 를 학습하는 방법과 Packet 자체를 학습하는 등의 방법이 있었다. 그러나 모두 정보의 제한적으로 사용한다는 단점이 있었다. 본 연구에서는 Http Request 에 대한 사전학습 기반의 효과적인 이상 탐지 방법을 제안한다. 사전학습에 고려되는 토큰화 방법, Padding 방법, Feature 결합 방법, Feature 선택 방법과 전이학습 시 Numerical 정보를 추가하는 방법을 소개하고 각 실험을 통해 최적의 방법을 제안한다.

1. 서론

보안 위협으로부터 자산을 보호하기 위해 다양한 이상 탐지 방법들이 발전해왔다. 최근 컴퓨터 비전이나 자연어처리와 같은 딥러닝 기반의 인공지능 기술이 발달함에 따라 특정 분야에서 정확도가 인간과 동등하거나 상회하는 경우도 나타나고 있다. 이러한 배경 속에 보안에도 딥러닝을 적용한 연구와 서비스 도입이 활발히 진행되고 있다.

이러한 연구 중, 딥러닝 모델에 Packet Count, Flow Size 등 네트워크 트래픽 Feature 값을 활용하는 방법의 연구가 있다[1,2]. 이러한 방법은 연산양이 적은 이점이 있지만 숫자나 범주형 데이터처럼 축약된 정보만 활용하기 때문에 데이터 활용 정도에 한계가 있다.

반면에 연속으로 등장하는 문자의 패턴을 정의 및 치환과 Gated CNN 과 LSTM 기반의 모델을 사용하는 Unggi Lee 등[3]의 연구와 바이트 단위의 토큰화 기반으로 Word2Vec 을 진행한 Mehedi Hassan 등[4]의 연구처럼 자연어처리 기술을 사용해 앞의 한계를 극복한 연구가 있다. 그러나 이전 두 개의 연구는 숫자를 텍스트처럼 토큰화하여 사용하므로 숫자가 가진 연속성과 대소 관계의 특성이 충분히 사용되었다고 보기 어렵다. 그리고 최근 자연어처리 여러 과제에서 기존 임베딩 방법보다 Transformer 기반의 사전학습 모델이 두드러지는 바, 해당 기술 적용을 통한 성능 개선점이 존재한다.

사전학습 모델 기반의 연구 중 ALBERT 를 적용한 연구[5]는 Transformer 기반의 사전학습 모델을 활용했다는 면에서 이전 연구보다 진보된 방법이나 앞선 연구와 마찬가지로 숫자 정보의 특성을 온전히 사용하지 않는다는 점에서 동일한 한계가 있다. 본 연구에서는 BERT 의 효과적인 사전학습 방법을 제안하고, 텍스트 정보에 숫자 정보를 연결하여 이상 탐지에 활용할 수 있는 새로운 사전학습 기반 이상 탐지 방법을 제안한다.

2. 모델 학습

BERT 를 포함하여 GPT2, XLM 등의 사전학습 모델은 대용량의 데이터를 사전학습한 모델을 과제에 알맞게 전이학습 함으로써 여러 과제에서 높은 정확도를 달성했다. 이전에 Word2Vec, Fasttext 등의 임베딩

방법이 존재하였으나 최근 사전학습 기반 방법의 우수성이 증명되면서 널리 사용되고 있다. 그러나 사전학습 모델 사용 시, 기존 연구에서 제안하지 않은 새로운 도메인에 적용하기 위해서는 응용이 필요하다. 본 장에서 Http Payload 데이터를 BERT 에 사전학습에 적용 가능한 토큰화 기법과 Masked Language Model(이하 MLM) 방법, 입력 데이터 구성 방법을 설명한다. 마지막으로 전이학습에 Numerical 정보를 추가하는 방법을 설명한다.

2.1 전처리

하나의 Http Request 를 한 개의 인스턴스로 정의하고 토큰화를 진행하였다. 토큰화 알고리즘은 WordPiec 를 사용하였다. WordPiece 는 텍스트 데이터를 초기에 공백 기준으로 분리한 다음, 빈도 기반 점수를 산정하여 토큰 내부의 문자를 결합 및 치환하는 방법이다.

그러나 Http Payload 의 경우, 초기 분리자인 공백이 적기 때문에 긴 문자의 토큰들이 추가되어 Unknown 토큰의 빈번한 발생과 의미가 희석되는 한계점이 있다. 그래서 Payload 에서 주로 사용되는 괄호나 조건 기호를 추가하여 초기 분리 기준으로 활용하였다. 기호들은 분리 이후에 개별 토큰으로 사용하였다. 분리 기준이 되는 15 개 기호는 아래와 같다.

<Space> & ( ) = [ ] { } < > ? / , .

그 외에 데이터의 수에 맞춰 토큰 최소 빈도, Vocab Size 를 정하였다.

2.2 사전학습 방법

사전학습 방법인 MLM 은 인스턴스 문장 내 일부 토큰을 제공하지 않거나 변형하는 마스킹 작업을 진행한 다음, 입력된 주변 정보만으로 가려진 토큰을 맞추는 과제이다.

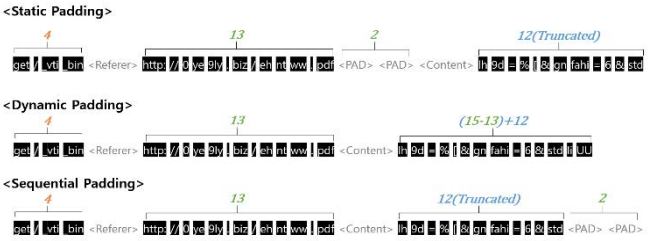
본 연구에서는 성능을 높이고자 RoBERTa 연구를 참고하여 NSP 과제를 진행하지 않았다. 그리고 MLM 도 Dynamic Masking 을 적용하였다.

기존 BERT 연구에는 다양한 과제에 대한 BERT 학습 방법을 제안하였으나 네트워크 트래픽에 대한 사전학습 방법에 대해선 미흡하였다. 그래서 본 연구에서는 효과적인 네트워크 트래픽의 사전학습 입력형태를 찾기 위해 Feature 결합과 Padding 방법 차이에 대한 실험을 진행하였다.

먼저 Http Request 데이터를 구성하고 디코딩하여 읽을 수 있는 형태로 변형하였다. 그런 다음 Header 와 Body 를 다음의 세 가지 방법으로 결합하였다.

첫 번째는 공백을 사용해 결합하는 방법이고 두 번째는 Separate 토큰만 사용하여 결합하는 방법, 마지막은 각 Feature 마다 개별 토큰을 사용하는 방법이다. 이 세 가지에 대한 성능 비교 실험을 진행하여 최종 결합 방법을 선정하였다.

결합 후, Padding 은 다음의 세 가지 방법(그림 1)으로 진행하였다. 첫 번째, Static Padding 방법은 각 Feature 마다 고정된 최대 토큰 수를 갖도록 처리하는 방법으로 짧으면 Pad 를 추가하고 길면 Truncate 를 진행한다. 두 번째, Dynamic Padding 방법은 모든 Feature



(그림 1) 샘플 데이터에 대한 Padding 예시

의 토큰을 연결한 다음, 각 Feature 중에 길이 분포를 크게 벗어난 Feature 만 길이를 줄인다. 그리고 결합 후에 맨 뒤에 Pad 를 추가하는 방법이다. 마지막 Sequential Padding 방법은 개별 Feature 의 최대 길이보다 큰 경우에 대해서 Truncate 만 진행하고 마지막 토큰에 부족한 길이 만큼 Pad 를 추가하는 방법이다.

2.3 전이학습 방법

본 연구에서는 기존의 BERT 분류 방법을 응용하여 Classification 토큰의 Representation 벡터에 숫자 정보를 연결한 다음, 정규화를 진행 후에 다층 퍼셉트론으로 분류하도록 구성하였다.

3. 연구 결과

3.1 데이터

공개된 데이터와 내부 데이터를 사용하여 실험을 진행하였다. 데이터에서 Method, URI, User Agent, Host, Referer 정보를 추출하여 사용하였다.

공개된 데이터는 ECML/PKDD 2007 의 Web Traffic 데이터로 총 50,000 개의 인스턴스 구성과 다양한 공격을 포함하고 있다. 본 연구에서는 공격과 정상 두 가지로 구분하여 8:2 의 비율로 사용하였다.

내부 데이터는 2022.06.10~16, 7 일간 실제 서비스 운영으로 발생한 Web Traffic 데이터와 프로그램 생성 공격 데이터를 결합하여 구성하였다. 총 548,000 개의 인스턴스 구성과 공격과 정상이 9:1 로 구성되어 있다.

위의 데이터는 8:1:1 의 비율로 Train, Validation, Test 데이터를 구성하여 실험을 진행하였다.

3.2 사전학습 모델 선정

동일한 데이터와 네트워크에서 최적의 언어모델 선정 기준으로 PPPL 을 사용하였다. PPPL 은 MLM 평가 방법[6]으로 PPPL 은 작을수록 해당 언어 재현 성능이 더 높은 것을 의미한다.

다른 구성의 네트워크 모델 간의 비교 시에는 전이학습을 진행하여 가장 높은 F1 Score 를 달성한 모델을 최종 모델로 선정하였다.

3.3 Padding 방법 비교실험

세 가지 Padding 방법에 따른 실험 결과를 비교한다. 내부 데이터에 대해 각각 Static Padding, Dynamic Padding, Sequential Padding 방법으로 사전학습 및 전이학습을 진행하였다.

<표 1>의 결과를 산출하여 Sequential Padding 이 가

장 우수한 방법임을 확인할 수 있다. Static Padding 의 경우 Padding 토큰이 문장 중간에 등장하는 것이 학습을 어렵게 만든 요인으로 추측된다. Dynamic Padding 은 유동적으로 각 Feature 의 길이를 조절하는 것이 각 Feature 의 위치 정보가 달라지는 요인이 되어 학습에 부정적 영향을 미친 것으로 여겨진다.

<표 1> Padding 차이에 따른 분류 모델 성능

Type	F1	ACC
Static	0.935	0.939
Dynamic	0.937	0.933
Sequential	**0.979**	**0.979**

3.4 토큰 구성 비교실험

PKDD 데이터에 대해 Special 토큰 선정 방법으로 세 가지 방법을 비교한다. Padding 실험과 마찬가지로 동일한 조건에서 문장 결합에 사용한 Special 토큰만 차이를 두어 사전학습과 전이학습을 진행하였다. 실험 결과, Feature 결합에 공백이나 Separate 토큰을 사용하는 방법보다 Feature 마다 별도의 토큰으로 구분하는 방법이 더 좋은 결과를 보였다. Feature 마다 다른 Special 토큰으로 경계를 표시해 한 문장으로 결합함으로써 Feature 간에 다른 특성이라는 것을 활용해 학습하는 것으로 보인다.

<표 2> Feature 결합 토큰에 따른 모델 성능

Type	F1	ACC
Only [SEP]	0.916	0.952
Only Space	0.918	0.953
Token of Feature	**0.922**	**0.955**

3.5 Feature 선정 비교실험

EDA 를 통해 Http Request 의 각 Feature 별 정상과 공격 데이터 간의 분포 차이를 확인했다. 값의 다양성, 크기 및 길이 분포를 확인한 결과, Body 의 텍스트와 Header 의 URL, User-Agent, Host, Referer 가 선정되었다. 모델 최대 입력 길이의 한계와 불필요한 변수 입력을 줄이기 위해, 동일한 구조의 모델에서 각 변수 조합을 달리하여 성능을 산출하였다.

<표 3> Feature 조합에 따른 모델 성능

Model	F1	ACC
URL+Host+Referer+User-Agent	0.863	0.926
URL+Content+User-Agent	**0.922**	**0.955**
URL+Content+Host+Referer	0.907	0.946
ALL	**0.922**	**0.955**

<표 3>의 결과를 산출하여 PKDD 데이터에 대해 두 가지 모델이 동일한 성능을 보였다. 그러나 각 Feature 의 길이 제한이 필요하다는 점을 감안하면 더 적은 Feature 를 사용하는 모델이 길이 제한 문제에 더 유리하다. 그러므로 URL, Content, User-Agent 를 최적의 조합으로 선정하였다.

3.6 Feature 추가 비교실험

선정된 사전학습 모델에 대해 분류 과제를 수행할 수 있도록 전이학습을 진행하였다. 전이학습에는 텍스트 Feature 만 학습하는 모델과 Body 에서 얻을 수 있는 길이 정보인 Numerical 정보를 추가했을 때의 비교 실험을 진행하였다.

<표 4> 데이터별 모델 성능 비교

Data	Model	F1	ACC
PKDD 07	BERT-base(Only Finetuned)	0.894	0.940
	Variable Selected	0.922	0.955
	Variable Selected + Numeric	**0.923**	**0.956**
내부 데이터	BERT-base(Only Finetuned)	0.876	0.901
	Variable Selected	0.979	0.979
	Variable Selected + Numeric	**0.979**	**0.979**

분류 모델 비교 결과 <표 4>를 통해 Numerical 정보를 추가했을 때 전이학습 결과가 소폭 상승하는 것을 확인하였다.

4. 결론

보안 데이터에 대한 사전학습과 전이학습을 진행하여 분류 모델을 개발하였다. 실험을 통해 Http Request 데이터에 대한 사전학습 시, Sequential Padding 과 Special 토큰을 활용이 분류 성능에 긍정적 효과를 준다는 것을 확인하였다. 그리고 적절한 Feature 조합과 Numerical 정보의 활용이 성능 개선에 도움이 됨을 확인하였다. 앞으로 본 연구의 내용을 적용하여 다양한 공개 데이터에 대한 실험을 진행하고 Classification 층을 다양화하여 연구를 발전시킬 수 있을 것이다.

참고문헌

[1] Kolias, Constantinos, Georgios Kambourakis, Angelos Stavrou, and Jeffrey Voas. "DDoS in the IoT: Mirai and other botnets." Computer 50, no. 7 (2017): 80-84.

[2] Hallman, Roger, Josiah Bryan, Geancarlo Palavicini, Joseph Divita, and Jose Romero-Mariona. "IoDDoS-the internet of distributed denial of sevice attacks." In 2nd international conference on internet of things, big data and security. SCITEPRESS, pp. 47-58. 2017

[3] Unggi Lee, II-Ho Choi, SeulKi Jun, Woo-hyuk Jung. “Cyber treat detection methodology using semi-supervised learning based on pattern”, CISC-W21, Seongnam, November 2021, pp 556-559

[4] Mehedi Hassan, Md Enamul Haque, Mehmet Engin Tozal, Vijay Raghavan, and Rajeev Agrawal. “Intrusion detection using payload embeddings”, IEEE Access, 10:4015–4030, 2021

[5] Luchao Han, Xuewen Zeng and Lei Song, “A NOVEL TRANSFER LEARNING BASED ON ALBERT FOR MALICIOUS NETWORK TRAFFIC CLASSIFICATIO N", International Journal of Innovative Computing, Information and Control Volume 16, Number 6, December 2020

[6] Julian Salazar, Davis Liang, Toan Q. Nguyen and KatrinKirchhoff, “Masked Language Model Scoring”, In Proceedings of the 58th Annual Meeting of the ACL, Online, July 2020, pp. 2699–2712