논문 19-22-01-03
KNOM Review '19-01 Vol.22 No.01
https://doi.org/10.22670/knom.2019.22.1.20

컨볼루션 신경망 기반 유해 네트워크 트래픽 탐지 기법 평가

염 성 웅◆, 뉘엔 반 퀴엣*, 김 경 백°

Assessing Convolutional Neural Network based Malicious Network Traffic Detection Methods

Sungwoong Yeom◆, Van-Quyet Nguyen*, Kyungbaek Kim°

요 약

최근 유해 네트워크 트래픽을 탐지하기 위해 머신러닝 기법을 활용하는 다양한 방법론들이 주목을 받고 있다. 이 논문에서는 컨볼루션 신경망 (Convolutioanl Neural Network)을 기반으로 유해 네트워크 트래픽을 분류하는 기법을 소개하고 그 성능을 평가한다. 이미지 처리에 강한 컨볼루션 신경망의 활용을 위해, 네트워크 트래픽의 주요 정보를 규격화된 이미지로 변환하는 방법을 제안하고, 변환된 이미지를 입력으로 컨볼루션 신경망을 학습시켜 유해 네트워크 트래픽의 분류를 수행하도록 한다. 실제 네트워크 트래픽 관련 데이터셋을 활용하여 이미지 변환 및 컨볼루션 신경망 기반 네트워크 트래픽 분류 기법의 성능을 검증하였다. 특히, 다양한 컨볼루션 신경망 기반 네트워크 모델 구성에 따른 트래픽 분류 기법의 성능을 평가하였다.

Key Words : Convolutional Neural Network, Traffic Classification, Image Transform, Configuration

ABSTRACT

Recently, various machine learning based traffic classification methods are focused on detecting malicious network traffic. In this paper, convolutional neural network based malicious network traffic classification method is introduced and its performance is evaluated. In order to utilize the convolutional neural network which is excellent in analyzing images, a image transform method from important information of network traffic to a standardized image is proposed, and the transformed images are used as learning input of a CNN network traffic classifier. By using the real network traffic dataset, the proposed image transform method and CNN based network traffic classification method are evaluated. Especially, under various configurations of CNN, the performance of the proposed method is evaluated.

※이 논문은 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임 (NRF-2017RIA2B4012559)
※본 연구는 과학기술정보통신부 및 정보통신기획평가원의 대학 ICT 연구센터 지원사업의 연구결과로 수행되었음 (IITP-2019-2016-0-00314)
◆ First Author : Chonnam National University, Department of Electronics and Computer Engineering, yeomsw0421@gmail.com
° Corresponding Author : Chonnam Ntional University, Department of Electronics and Computer Engineering, kyungbaekkim@jnu.ac.kr, 정회원
* Faculty of Information Technology, Hung Yen University of Technology and Education, quyetict@utehy.edu.vn

논문번호 : KNOM2019-01-05, Received June 23, 2019; Revised July 5, 2019; Accepted August 15, 2019

Ⅰ. 서 론

네트워크 기술의 발달과 IoT 기기의 활성화에 따른 네트워크상의 트래픽의 복잡도가 점점 높아지면서, 네트워크에 유해 트래픽을 유발시켜서 네트워크 서비스의 질을 저하시키거나 특정 서버 및 호스트의 동작에 피해를 입히는 네트워크 공격에 대한 탐지 및 방어가 더욱 중요해지고 있다.

최근, 머신러닝 기반의 네트워크 연구가 주목을 받고 있으며, 특히 네트워크 공격 트래픽 분류기법 또한 지속적으로 연구되고 있다.[1][2][3][4] 이전의 연구들에서는 주로 종단 간 연결정보, 도메인정보, 데이터전송정보와 같은 여러 네트워크 트래픽 정보를 특징벡터로 이용하는 다양한 분류기 (SVM, KNN, Naive Bayes)를 활용하여 네트워크 공격 트래픽을 분류하는 방법을 제안하였다.[1][2]

본 논문에서는 딥러닝 기법중 하나인 컨볼루션 신경망 (Convolutional Neural Network)를 이용하여 네트워크 공격 트래픽을 분류하는 기법을 소개한다. 제안하는 기법은 네트워크 트래픽 정보를 규격화된 이미지로 변환하고, 변환된 이미지들을 이용해 컨볼루션 신경망 모델을 학습시켜 향후 네트워크 트래픽 분류를 위한 모델을 도출한다. 학습된 컨볼루션 신경망 모델을 이용해 입력되는 트래픽 정보가 일반적인 네트워크 트래픽인지 네트워크 공격에 사용되는 트래픽인지를 구분한다.

제안된 CNN기반 네트워크 트래픽 분류기법의 성능 검증을 위해 KDD 1999 데이터 셋과 CICIDS2017 데이터 셋을 활용한 검증을 수행하였다. Convolution layer, Pooling layer, Fully connected layer구성의 변화에 따른 CNN기반 네트워크 트래픽 분류기법의 성능을 평가하였고, 기존의 머신러닝 (Naive Bayes, KNN, SVM) 기반의 방법과의 비교를 수행하였다. 분류기법의 Accuracy의 정도뿐만 아니라, 각 분류모델을 도출하기 위해 필요한 학습 시간도 함께 평가하였다.

2장에서는 기계학습 기법과 컨볼루션 신경망 및 기계학습 기반 트래픽 분류 연구에 대해 소개하고, 3장에서는 제안하는 컨볼루션 신경망 기반 유해 네트워크 트래픽 탐지 기법을 소개한다. 4장에서는 KDD 1999 데이터 셋과 CICIDS 2017 데이터 셋을 기반한 제안기법의 성능 검증 결과를 기술하고, 5장에서 본 논문의 결론 및 향후 연구 내용에 대해 기술한다.

Ⅱ. 관련연구

1. Naive Bayes

Naive Bayes 분류 기법은 Bayes 이론을 적용하는 확률적 분류기법으로, 특징 벡터들 간의 독립성이 강할수록 그 성능이 좋아진다. Naive Bayes 분류 알고리즘은 대부분의 알고리즘만큼 복잡하지 않으므로 빠르고 쉽게 적용할 수 있다. 비록, 분류가 복잡한 알고리즘보다 정확하지 않지만 유사한 결과를 얻을 수 있다. 속성 값의 조건부 클래스 확률을 계산하기 위해 Naive Bayes 분류 알고리즘은 Bayes 정리를 사용한다.

이러한 확률은 데이터에서 직접 추측할 수 있으며 클래스의 속성은 조건부로 독립적이라는 가정 하에 계산할 수 있다. 실제로, 이 가정은 사실이 아닐 수도 있다. 이 조건을 지키지 않는다면 잘못된 확률 계산이 발생하지만 이러한 위반은 예측 정확도에 영향을 미치지 않을 수 있다. 계산에 잘못된 확률이 사용되더라도 예측은 정확할 수 있다. 이 확률 모델의 특성에 따라 지도학습 환경에서 이 알고리즘을 사용하여 매우 효율적으로 학습할 수 있다.[5]

2. SVM (Support Vector Machine)

SVM은 주어진 특징 벡터들 간의 마진을 최대로 하는 방법을 이용해 서로 다른 특징을 가지는 데이터 집합을 분류하는 기법으로, 다수의 특징벡터가 주어지더라도 집합간의 Support Vector를 구함으로써 분류기법을 안정적으로 운용할 수 있는 장점을 가진다.

SVM은 보이지 않는 데이터를 정확하게 분류하는 일반화를 최적화하는 것이다. 이 최적화는 오버피팅과 같은 다른 학습 알고리즘에서 나타나는 문제를 해결한다. 인공 신경망을 학습해야하는 것처럼 SVM을 학습해야한다. 입력 공간의 교육 데이터를 고차원의 기능 공간으로 맵핑한다.

클래스를 구별하는 최적의 초평면을 구성하여 특징 공간에서 선형 결정 경계를 결정한다. 이를 통해 SVM은 입력 공간에서 비선형 경계를 얻을 수 있다. 지원 벡터는 클래스 사이의 경계를 가장 잘 정의하는 입력 공간의 점이다. 입력 공간에서 계산을 수행할 수 있게 해주는 커널 함수를 사용하면 특징 공간에서 잠재적으로 어려운 계산을 피할 수 있다. 통계적 학습 이론의 개념은 좋은 일반화를 위해 어떤 요인을 통제해야 하는지 설명하는데 사용된다.[6]

3. KNN (K-Nearest Neighbor)

KNN은 임의의 데이터를 입력으로 이용하였을 때, 해당 데이터의 특징 벡터와 다른 데이터들의 특징 벡터와 유사도를 계산하여, 가장 유사도가 높은 K개의 데이터를 이웃으로 선택하는 기법이다. 만약 K = 1이고, KNN을 분류 기법으로 이용한다면, 입력된 데이터는 가장 유사도가 높은 하나의 그룹으로 분류된다.

KNN 알고리즘은 교육 데이터 세트에서 데이터를 수집하고 이 데이터를 새 레코드에 대한 예측을 위해 사용한다. 각각의 새 레코드에 대해 학습 데이터 세트 중 가장 가까운 레코드가 결정된다. 가장 가까운 레코드의 대상 속성 값에 따라 새 레코드에 대한 예측이 수행된다.

이 알고리즘은 임의의 인스턴스에 대한 분류 예측 또는 회귀 예측을 만든다. 이를 위해, 임의의 인스턴스에 가장 가까운 학습 인스턴스를 식별한다. 이후, 학습 인스턴스의 클래스 레이블을 반환한다.

KNN 알고리즘은 하나의 인스턴스만 사용하는 대신 가장 가까운 학습 인스턴스의 지정된 번호 K를 사용하여 이 프로세스를 확장한다. 학습 데이터 내의 노이즈 영향을 크게 받지 않으며 학습 데이터 수가 많을수록 효과적인 장점이 있다.[7]

4. CNN(Convolutional Neural Network)

CNN은 영상 이미지 분류를 위한 최신의 분류 모델로, 다수의 필터를 영상 이미지의 픽셀 데이터에 적용하여 고차원 특징을 추출하여 분류기를 학습하는 모델이다. 이때, 추출되는 고차원 특징들은 Convolution layer, Pooling layer, Fully connected layer로 구성되는 Hidden layer 내부에 존재하게 되어 각 특징들에 대한 자세한 정보를 확인하기 어렵고, 특별한 의미를 부여하기 힘들다.

4.1 Convolution Layer

컨볼루션 레이어는 CNN의 핵심 빌딩 블록이다. 레이어의 매개 변수는 수신 필드가 작지만 입력 볼륨의 깊이까지 확장되는 일련의 커널로 구성된다. 순방향으로 통과하는 동안 각 필터는 입력 볼륨의 너비와 높이에 따라 컨볼루션되어 필터 항목과 입력 항목 사이의 내적을 계산하고 해당 필터의 2차원 활성화 맵을 생성한다. 결과적으로 네트워크는 데이터 엔트리가 일정한 데이터 셋에서 특정 유형의 기능을 감지할 때 활성화되는 필터를 학습한다. 깊이 치수를 따라 모든 필터에 대한 활성화 맵을 쌓으면 컨볼루션 레이어의 전체 출력 볼륨이 형성된다.

4.2 Max-Pooling Layer

풀링은 비선형 다운 샘플링의 한 형태이다. Max Pooling이 가장 일반적으로 사용되며 입력을 겹치지 않는 사각형의 집합으로 분할하고 각 하위 영역에 대해 최대 값을 출력한다. 직관적으로 특징의 정확한 위치는 다른 특징과 비교하여 거친 위치보다 덜 중요하기 때문에 커널의 공간 크기를 점차 감소 시켜 네트워크의 매개 변수 수 및 계산량을 줄이고 오버 피팅을 제어하는 역할을 한다.

4.3 Flattening and Fully Connected layer

여러 가지 Convolution, Max-Pooling 레이어가 끝나면 신경망의 높은 수준의 추론은 Fully-connected 레이어를 통해 수행된다. Fully-connected 레이어의 뉴런은 (Non-Convolutional)일반 인공 신경 네트워크에서 볼 수 있듯이 이전 레이어의 모든 활성화 값들과 연결된다. 하지만, 이전의 여러 가지 레이어에서 이러한 Fully-connected 레이어에 2차원 자원을 넘겨주기 위해선 1차원 자료로 바꿔줘야 한다. 이때 사용되는 것이 Flatten 레이어이고 하나의 벡터를 만들어 낸다.

4.4 Softmax

Fully-connected layer의 벡터 내부에는 classes에 대한 점수가 담겨있다. Softmax는 네트워크의 비정규화된 출력을 확률로 변환하고 이 확률들의 합을 1이 된다. 다시 말해, Softmax는 예측 출력 클래스에 대한 확률 분포에 맵핑하기 위해 신경망에 사용된다.

5. 기계학습 기반 트래픽 분류 기법

일반적인 네트워크 트래픽 분류는 프로토콜 변수, 패킷의 크기 등을 이용하는 룰 기반 분류 기법을 활용해 수행되었다. 최근, 기계학습 기법이 성숙해짐에 따라, 기계학습 기반 네트워크 트래픽 분류 기법에 대한 연구가 각광을 받고 있다. 이러한 기법들은 사전에 제공되는 네트워크 트래픽 데이터를 통해 학습된 기계학습 기반 분류기(Classifier)를 이용해 여러 종류의 네트워크 트래픽을 분류한다. 이때, 분류기를 학습시키기 위한 데이터로 패킷 지속 시간, 패킷 길이, 시간, 프로토콜 변수 등의 네트워크 트래픽 특성을 사용할 수 있으며, Decision Tree,

SVM, KNN[2], Random Forest[11]와 같은 다양한 분류기를 활용할 수 있다. 또한, 최근 Deep learning 기술이 보편화 되면서, 네트워크 트래픽 분류에 Deep learning을 활용하는 연구가 진행되고 있다. [10]

Ⅲ. CNN기반 유해 네트워크 트래픽 탐지 기법

본 논문에서는 네트워크 트래픽 정보를 이미지로 표현하고, 이를 컨볼루션 신경망을 통해 학습시켜 네트워크 공격 트래픽을 분류하는 모델을 제공하는 기법을 제안한다. 컨볼루션 신경망은 이미지 데이터의 특징을 추출하지 않고, 데이터 패턴을 직접 학습하여 직관적인 분류를 가능하게 하는 Deep learning 기술 중 하나이다. 네트워크 트래픽 관련 정보를 이미지화 하여 컨볼루션 신경망으로 학습할 경우, 별도의 주요 특징 추출 없이 직관적 분류를 빠르게 수행할 수 있는 장점이 있다. 이러한 직관적 학습은 네트워크 트래픽 정보 뿐 만 아니라 네트워크 헤더 및 페이로드 단위에 적용가능하다. 이 논문에서는 네트워크 트래픽 관련 정보를 이용한 컨볼루션 신경망 기반의 트래픽 분류로 그 범위를 한정한다.

1. 네트워크 트래픽 정보 이미지 변환

제안하는 기법은 이미지 처리에 강점을 가지는 CNN 기반의 트래픽 탐지 기법으로, 우선적으로 네트워크 트래픽 정보를 이미지로 변환하는 것이 필요하다. 이를 위해, 우선 네트워크 트래픽 데이터의 각 필드정보를 이미지의 하나의 픽셀로 표현하는 데이터 변환을 수행한다. 이 논문에서는 하나의 픽셀이 8bit 정보를 표현하도록 하여 각 픽셀은 0부터 255까지의 값을 가지도록 하였다.

Algorithm 1은 네트워크 트래픽 정보를 8bit 픽셀정보로 바꾸는 알고리즘이다. 기본적으로 트래픽 정보가 숫자일 경우에는 각 필드정보(column)의 최소값과 최대값을 구한 후, line 10과 같이 현재의 값(r)에서 최소값(min)을 뺀 후, 해당 필드의 변화 구간의 길이(range)로 나눈 후 255로 정규화 한다. 만약 트래픽 정보가 문자일 경우에는 StringHash255 함수를 통해 임의의 문자열을 255로 정규화 한다.

변환되는 이미지는 임의의 길이와 높이가 h인 정사각형 형태를 가지도록 하였다. 만약 트래픽 데이터 필드의 개수가 h x h와 다를 경우, 0의 값을 추가하여 정사각형 이미지를 만들 수 있도록 이미지 변환작업을 수행한다.

```
Algorithm 1 : Dataset Formalization
Requre : Network Traffic Dataset
Ensure : New formalized dataset with range from
0 to 255
data=load(NetworkTrafficDataset)
foreach c in Columns(data)
  min = findMinimum(c)
  max = findMaximum(c)
  range = max - min
  foreach r in Rows(c)
    if(r is String) then
        new_val_r = StringHash255(r)
    else then
     new_val_r = (r – min)/range * 255
   r = new_val_r
```

변환되는 이미지는 임의의 길이와 높이가 h인 정사각형 형태를 가지도록 하였다. 만약 트래픽 데이터 필드의 개수가 h x h와 다를 경우, 0의 값을 추가하여 정사각형 이미지를 만들 수 있도록 이미지 변환작업을 수행한다. 이 논문에서 활용한 두 가지 트래픽 데이터셋의 경우, KDD 1999 데이터셋[8]은 TCP connection 특징 9가지, Domain knowledge 관련 connection 특징 13가지, 2초간의 connection traffic 특징 9가지, 그리고 2초 이상에 해당하는 공격 특징 10가지를 포함하는 41개의 필드를 가지고 있고, CICIDS2017 데이터셋[9]은 Duration, Number of packets, Number of bytes, Length of packets, FlowID, SourceIP, DestinationIP, SourcePort, Destination Port, Protocol등 트래픽의 forward 및 backward direction에 대한 78개의 필드를 가지고 있다. 이 논문에서는 KDD 1999 데이터셋의 경우 41개의 필드에 8번의 zero-padding을 추가해 7x7 이미지로 변환하였고, CICIDS 2017 데이터셋의 경우 78개의 필드에 3번의 zero-padding을 추가해 9x9 이미지로 변환하였다. 이미지 변환 시 수행되는 zero-padding의 경우 임의의 위치에 무작위로 삽입이 가능하다. 그림 1은 트래픽 데이터의 이미지 변환 과정을 나타낸다.

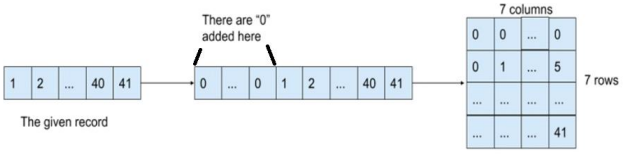

그림 1. KDD 1999 데이터 이미지 변환 과정
Fig. 1. Image Transform Process of KDD 1999 dataset

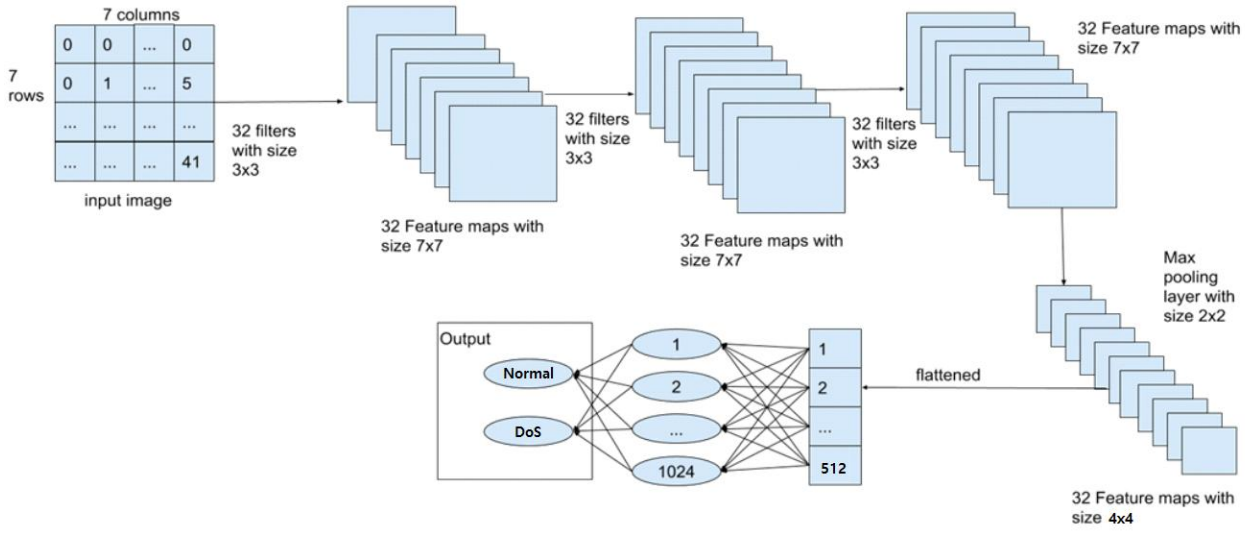

그림 2. KDD 1999 데이터를 위한 CNN기반 네트워크 트래픽 분류기 구조 예시
Fig. 2. Example of a structure of CNN based network traffic classifier for KDD 1999 dataset

2. CNN 기반 네트워크 트래픽 분류

변환된 이미지는 CNN 기반 네트워크 트래픽 분류기의 입력으로 사용된다. CNN 기반의 분류기는 Convolution Layer, Max-Pooling Layer, Fully Connected Layer로 구성되는데, 제안하는 기법에서 사용되는 이미지의 크기가 작고 하나의 픽셀이 8bit를 사용하는 점을 고려해서 각 Layer의 구성을 고려할 수 있다.

일반적인 영상처리에서 사용되는 CNN 구조는 각 Convolution Layer에서 Pooling을 수행하여 입력되는 이미지의 특징을 줄이게 된다. 하지만, 제안하는 기법에서 사용되는 이미지는 7x7 또는 9x9 와 같이 매우 작은 이미지를 사용하게 되어 Convolution Layer에서 Pooling을 수행하여 입력 feature의 수를 줄일 필요성이 크지 않다. 또한 Filter를 이용해 Convolution을 수행할 때도, zero-padding을 통해 입력 feature의 사이즈를 유지할 수 있도록 하였다. 다만, 다양한 feature의 특징을 확보하기 위해 Convolution시에 32개의 필터를 사용하였다.

32개의 필터를 이용해 Convolution을 수행한 이후, Fully Connected Layer의 입력 노드의 수를 줄이기 위해 Max-Pooling Layer를 사용한다. 이 논문에서는 2x2, 3x3, 4x4 필터를 이용해 pooling을 수행하였다.

Fully Connected Layer에서는 512 또는 1024개의 Hidden Unit을 이용해 각 Layer를 구성하고, 하나 또는 두 개의 Layer를 이용해 Fully Connected Layer를 구성하였다. 다만, Hidden Unit의 개수는 Fully Connected Layer의 입력 노드 수보다는 많도록 설정하였다. Fully Connected Layer 출력은 각 데이터셋에서 사용한 트래픽 라벨 개수와 같도록 하고, SoftMax기법을 활용해 네트워크를 학습시켰다.

그림 2에서, KDD 1999 데이터를 위한 CNN기반 네트워크 트래픽 분류기 구조에 대한 하나의 예를 나타낸다. 이 예에서는 Convolution을 위해 3x3 filter 32개를 사용하고, 3번의 Convolution을 수행하고, 2x2 filter를 이용해 Max-pooling을 수행한다. 이 후, 512의 입력노드와 1024개의 hidden unit을 가지고, Normal과 DoS의 두 개의 출력을 가지는 1 level Fully Connected Layer을 학습시켜 입력되는 네트워크 트래픽 변환 이미지의 Label을 분류한다.

Ⅳ. 성능 검증

제안된 기법의 성능 검증을 위해, 제안되는 기법을 Tensorflow를 이용해 구현하고 KDD 1999 데이터셋 및 CICIDS 2017 데이터셋을 이용해 구현된 기법의 Accuracy 및 False Positive를 측정하였다.

1. Dataset

1.1 KDD 1999 데이터셋

KDD 1999는 비정상 탐지 방법의 평가를 위해 만들어진 데이터셋이다. 2주간 수집한 KDD 학습 데이터셋은 약 4,900,000개의 단일 연결 벡터로 구성되며 각 연결 벡터는 TCP connection 특징 9가

지, Domain knowledge 관련 connection 특징 13가지, 2초간의 connection traffic 특징 9가지, 그리고 2초 이상에 해당하는 공격 특징 10가지를 포함하는 41개의 특징으로 구성되며, 정상 또는 공격 라벨을 가진다. 공격은 다음 네 가지 범주 중 하나에 속한다[8]:

a) DoS (Denial of Service Attack): 시스템을 악의적인 공격으로 해당 시스템의 자원을 부족하게 하여 의도된 용도로 사용하지 못하게 하는 공격 유형이다.

b) User to Root Attack (U2R): 시스템 일반 사용자 계정의 취약점을 악용하여 루트 엑세스를 얻는 공격 유형이다.

c) Remote to Local Attack (R2L): 해당 컴퓨터의 계정이 없는 공격자가 일부 취약점을 악용하여 해당 컴퓨터 사용자를 통해 로컬 액세스를 얻어 발생하는 공격 유형이다.

d) Probing Attack: 보안 통제를 우회하여 컴퓨터 네트워크에 대한 정보를 수집하려는 공격 유형이다.

이 논문에서는 KDD1999 전체 데이터셋 중에서 1,000,000개의 데이터를 이용해 제안하는 기법의 성능을 검증하였다. 사용된 데이터셋의 트래픽 종류의 분포는 표 1과 같다.

1.2 CICIDS 2017 데이터셋

CICIDS2017 데이터셋은 5일에 걸쳐 캐나다 사이버 보안 연구소의 정상 트래픽과 공격 트래픽 데이터를 8가지 파일로 수집했다. 각 파일에 저장된 트래픽 데이터는 Duration, Number of packets, Number of bytes, Length of packets, FlowID, SourceIP, DestinationIP, SourcePort, Destination Port, Protocol등 트래픽의 forward 및 backward direction에 대한 78개의 필드를 가지고 있으며, Benign과 다양한 공격으로 라벨링 되어있다. 5일 동안 이루어진 공격은 요일마다 종류가 다르다. 이 논문에서는 다수의 DoS 공격이 이루어진 수요일 근무시간의 트래픽을 데이터 셋으로 사용하고, 사용된 데이터 셋의 구성은 표 2와 같다.

표2 에서 볼 수 있듯이 Benign의 전체 트래픽의 63.52%이며 Heartbleed 공격의 경우 전체 트래픽의 0.00017%에 해당한다. 이와 같은 차이에 의해, 데이터 학습을 통한 탐지기가 양성으로 기울 여지가 높다. 또한, 검출기가 이 데이터 세트의 샘플을 무작위로 추출할 경우, "Heartbleed"와 같은 특정 공격 레이블의 인스턴스가 교육 집합에서 발견되지 않을 가능성이 크다. 결과적으로 탐지기는 이러한 유형의 공격 인스턴스가 도착할 때 공격을 탐지하지 못할 수 있다.

표 1. KDD 1999 테스트 데이터셋 구성
Table 1. Composition of KDD 1999 test set

Class Labels	Number of instances
normal	558,227
dos	419,436
r2l	1,125
probe	21,188
u2r	24

표 2. CICIDC 2017 테스트 데이터셋 구성
Table 2. Composition of CICIDC 2017 test set

Class Labels	Number of instances
BENIGN	440,032
DoS GoldenEye	10,294
DoS Hulk	231,074
DoS Slowhttptest	5,500
DoS slowloris	5,797
Heartbleed	12

이것은 CICIDS2017 데이터 세트에서 발견된 주요 단점이다. 이 논문에서는 이러한 단점을 해결하기 위해 무작위 샘플링을 사용하지 않고, 모든 종류의 데이터를 학습 집합으로 사용하도록 설정 하였다.

2. CNN기반 트래픽 분류기 성능 평가

제안된 기법의 성능 평가를 위해 상기 기술한 두 개의 데이터셋을 이용해 3-fold cross validation을 수행하여 분류기의 Accuracy를 측정하였다. 제안된 기법은 Tensorflow 1.14를 이용해 구현하였다.

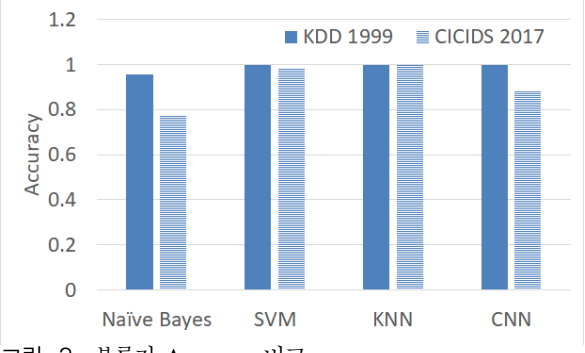

그림 3. 분류기 Accuracy 비교
Fig. 3. Comparison of accuracy of Classifiers

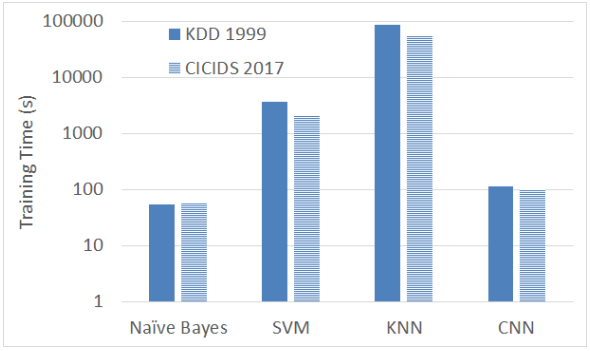

그림 4. 분류기 Training Time 비교
Fig. 4. Comparison of Training Time of Classifiers

그림 3은 제안된 CNN기반 트래픽 분류기와 Naive Bayes, SVM, KNN의 Accuracy를 비교결과를 나타낸다. Naive Bayes, SVM, KNN 기반 트래픽 분류는 Weka 3.8.1을 활용해 수행하였다. 수행결과, Naive Bayes는 두 개의 데이터 셋에 대해 가장 낮은 성능을 보였다. 반면, SVM과 KNN은 두 데이터 셋에서 모두 98%이상의 Accuracy를 가지는 것을 확인하였다. CNN의 경우 KDD 1999 데이터 셋의 경우 99.7%의 Accuracy를 달성하였으나, CICIDS 2017 데이터 셋의 경우 88.6%의 Accuracy를 달성하였다.

그림 4는 서로 다른 분류기의 학습 시간을 나타내고 있다. Naive Bayes와 CNN의 경우 학습시간이 모두 약 100초정도 걸린 것을 확인할 수 있다.

반면, SVM은 1000초 이상, KNN은 100,000초 정도 걸리는 것을 확인할 수 있었다. 즉, Naive Bayes와 CNN이 SVM과 KNN에 비해 10배에서 100배정도 빠르게 동작한다는 것을 확인하였다.

이러한 결과를 종합하면, 데이터 간의 distance를 비교하는 SVM과 KNN의 경우 성능은 우수하나 학습 및 검사에 드는 비용이 매우 크다는 단점이 있다. CCN의 경우 Naive Bayes와 같이 학습에 이용되는 비용이 적지만, SVM과 KNN에 근접하는 Accuracy를 달성할 수 있다는 점을 확인하였다.

3. CNN 구성에 따른 트래픽 분류기 성능 평가

제안된 CNN 기반 트래픽 분류기는 CNN 모델의 구성에 따라 그 성능이 달라질 수 있다. 이 논문에서는 Convolution Layer의 수, Max-Pooling Filter Type, Depth of Fully Connected Layer, Number of Hidden Unit를 변화시키며 CNN기반 분류기의 Accuracy를 측정하였다. 표 3과 표4는 다양한 CNN 구조 구성에 대한 KDD 1999 데이터셋과 CICIDS 2017 데이터셋을 사용하였을 때의 CNN 기반 트래픽 분류기의 Accuracy와 Training time측정 결과를 나타낸다.

표 3. 다양한 CNN 구성에 따른 CNN기반 트래픽 분류기 정확도 (KDD 1999)
Table 3. Accuracy of CNN based Traffic Classifier with various CNN configuration (KDD 1999)

# of Conv	Type of Pooling Filter	Depth of Fully Con. Layer	# of Hidden Units	Accuracy	training time (s)
1	-	1	512	**0.987218**	44.4
			1024	**0.987111**	84.1
		2	512	0.986629	54.2
			1024	0.985939	132.0
	2x2	1	512	0.983915	17.6
			1024	**0.987078**	22.2
		2	512	0.986846	25.9
			1024	**0.987414**	67.2
	4x4	1	512	0.98649	25.5
			1024	0.986547	38.6
		2	512	0.986308	35.0
			1024	0.986084	87.0
2	-	1	512	0.985449	73.0
			1024	0.983912	110.1
		2	512	0.986659	82.4
			1024	0.983407	161.4
	2x2	1	512	0.986875	45.6
			1024	0.985642	50.1
		2	512	0.986477	55.8
			1024	0.985349	95.5
	4x4	1	512	0.979749	54.9
			1024	0.973917	68.5
		2	512	0.979627	65.3
			1024	0.97961	116.4
3	-	1	512	0.975236	99.8
			1024	0.974967	138.8
		2	512	0.97749	110.2
			1024	0.974661	188.5
	2x2	1	512	0.977068	75.4
			1024	0.974787	80.3
		2	512	0.985392	86.4
			1024	0.981334	126.9
	4x4	1	512	0.980674	86.2
			1024	0.979657	99.6
		2	512	0.980622	96.6
			1024	0.977139	149.5

표 4. 다양한 CNN 구성에 따른 CNN기반 트래픽 분류기 정확도 (CICIDS 2017)
Table 4. Accuracy of CNN based Traffic Classifier with various CNN configuration (CICIDS 2017)

# of Conv	Type of Pooling Filter	Depth of Fully Con. Layer	# of Hidden Units	Accuracy	training time (s)
1	2x2	1	1024	**0.886356**	53.6
		2	1024	**0.886473**	53.0
	3x3	1	1024	**0.886072**	29.7
		2	1024	0.885704	30.3
1	2x2	1	1024	0.88585	93.9
		2	1024	**0.886165**	94.6
	3x3	1	1024	0.884439	71.6
		2	1024	0.883632	73.2

실험 결과, 대부분의 구성에서 대해 CNN 기반 트래픽 분류기의 Accuracy는 비슷한 결과가 나오는 것을 확인하였다. 또한, Convolution Layer를 추가하지 않는 경우에 미세하게나마 높은 Accuracy를 얻을 수 있었다. 또한, Pooling filter를 사용하지 않거나 더 작은 크기의 Pooling filter를 사용하는 경우 또한 높은 Accuracy를 얻을 수 있었다.

이러한 결과는 제안하는 CNN 기반 트래픽 분류기에 사용되는 이미지의 크기와 연관이 있는 것으로 분석된다. 이 검증에서 사용되는 이미지의 크기는 가 7x7 또는 9x9로 일반적인 CNN 응용에서 사용되는 영상보다 매우 작은 크기이다. 따라서 Convolution을 여러 번 거치거나 큰 사이트의 필터로 Pooling을 수행할 경우, 각 픽셀정보가 가지는 데이터의 정보의 분류기에 미치는 영향이 약해져서 성능이 미세하게 떨어지는 것으로 분석된다.

CNN기반 트래픽 분류기의 학습 시간은 Convolution Layer의 개수가 늘어나거나, Fully Connected Layer가 늘어나거나, Hidden Unit의 개수가 늘어남에 따라 증가하는 반면, Pooling filter의 크기가 커질수록 학습시간은 줄어드는 것을 확인하였다. 이 검증에서는 1번의 Convolution후, 2x2 필터를 사용한 풀링을 수행하고, 2014개 hidden unit을 가지는 2개의 layer로 구성된 fully connected layer를 사용하는 경우, 상대적으로 적은 학습 비용으로 높은 정확도를 달성할 수 있음을 확인하였다.

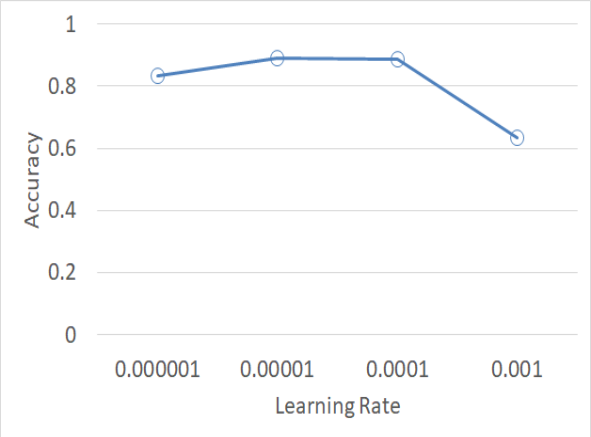

그림 5. Learning Rate에 따른 CNN 기반 트래픽 분류기의 Accuracy
Fig. 5. Accuracy of CNN based traffic classifier with differenct learning rate

CICIDS 2017 데이터셋의 경우, 데이터의 쏠림에 의해 분류기가 학습이 제대로 되지 않는 경우를 발견할 수 있었다. 특히 CNN 학습에 사용되는 Adam Optimizer의 Learning Rate에 분류기 학습 정도가 영향을 받는 것을 확인 할 수 있었다. 그림 5는 CICIDS 2017 데이터 셋에서 Learning Rate가 Accuracy에 미치는 영향을 나타내고 있다. 여기서, Learning Rate를 0.00001로 설정하였을 때 안정적으로 높은 정확도를 확보할 수 있음을 확인하였다.

V. 결 론

본 논문에서는 유해 네트워크 트래픽을 탐지하기 위한 컨볼루션 신경망 (CNN) 기반 네트워크 트래픽 분류 기법을 제안하였다. 제안하는 기법은 네트워크 트래픽의 특징 벡터를 이미지로 변환하고 이를 CNN을 적용하여 분류기 학습에 이용함으로써, DoS 공격 네트워크 트래픽과 일반 네트워크 트래픽을 성공적으로 분류한다. 특히, 기존의 머신러닝 기법 기반의 트래픽 분류 기법과 비교하였을 때, 제안하는 CNN기반 네트워크 트래픽 분류 기법은 상대적으로 빠른 시간에 수용가능한 정도의 높은 정확도를 달성할 수 있음을 실제 네트워크 트래픽 데이터를 이용한 실험을 통해 검증하였다. 또한, 다양한 CNN 분류기의 구조를 평가하여, 트래픽 정보와 같이 변환 후 작은 크기의 이미지를 이용하는 CNN 분류기를 위해 적합한 CNN 구조를 도출하였다. 향후, 제안된 기법을 적용하여 실제 네트워크 시스템에서 실시간으로 해당 모델을 학습시키고 공격을 탐지하는 방법에 대한 연구를 진행 하고자 한다.

References

[1] Jintae Choi, Nguyen Sinh-Ngoc Nguyen, Kyungbaek Kim, "Performance Comparison of Traffic Classification Methods for Detecting Malicious Network Traffic," *in Proc. KSII 2017 Fall Conference*, Vol. 19, No. 2, 2017

[2] Jintae Choi, Sinh-Ngoc Nguyen, Jeongnyeo Kim, Guee-Sang Lee, Kyungbaek Kim, "Performance Comparison of Traffic Classification Techniques for Detecting Malicious Network Traffic", *In the proceedings of SMA 2017 conference*, December 2017.

[3] Hee-Gon Kim, Do-Young Lee, Jae-Hyung Yoo, James Won-ki Hong, "A Machine Learning-based Method for Virtual Network Function Resource Demand Prediction," *KNOM Review*, Vol. 21, No. 2, Dec. 2018, pp. 1-9.

[4] Ui-Jun Baek, Mu-Gon Shin, Se-Hyun Jee, Jee-Tae Park, Myung-Sup Kim, "The Method of Feature Selection for Anomaly Detection in Bitcoin Network Transaction," *KNOM Review*, Vol. 21, No. 2, Dec. 2018, pp. 18-25.

[5] https://www.ibm.com/support/knowledgecenter/en/SS6NHC/com.ibm.swg.im.dashdb.analytics.doc/doc/r_naive_bayes_background.html

[6] Sharma, Anand & Sharma, Tanvi & Mansotra, Vibhakar. (2016). Performance Analysis of Data Mining Classification Techniques on Public Health Care Data. International Journal of Innovative Research in Computer and Communication Engineering. 4.

[7] https://www.ibm.com/support/knowledgecenter/en/SS6NHC/com.ibm.swg.im.dashdb.analytics.doc/doc/r_knn.html

[8] KDD Cup 1999 Data. [Online]. Available: http://kdd.ics.uci.edu/database/kddcup99/kddcup99.html, November 2017.

[9] Panigrahi, Ranjit, and Samarjeet Borah. "A detailed analysis of CICIDS2017 dataset for designing Intrusion Detection Systems." *International Journal of Engineering & Technology* 7.3.24 (2018): 479-482

[10] Shu, Jun Hua, Jiang Jiang, and Jing Xuan Sun. "Network Traffic Classification Based on Deep Learning." *Journal of Physics: Conference Series*. Vol. 1087. No. 6. IOP Publishing, 2018.

[11] Jun, Li, et al. "Internet traffic classification using machine learning." *2007 Second International Conference on Communications and Networking in China*. IEEE, 2007.

염 성 웅 (Sungwoong, Yeom)

2019년 2월 : 전남대학교 전자컴퓨터공학과 학사 졸업
2019년 3월~현재 전남대학교 전자컴퓨터공학과 석사과정
<관심 분야> 네트워크, 빅데이터, 데이터 스트리밍

뉘엔 반 퀴엣 (Van-Quyet Nguyen)

2009.08: Hung Yen University of Technology and Education. (B.S.).
2013.02: Hanoi University of Science and Technology (M.S. Degree).
2019.08 : School of Electronics and Computer Engineering, Chonnam National University (Ph.D. Degree).
2009 ~ now: Lecturer in Hung Yen University of Technology and Education.
<관심분야> BigData Platform, Parallel Processing, Recommendation Systems

김 경 백 (Kim Kyungbaek)

1999년 : 한국과학기술원 전기 및 전자공학과 학사 졸업
2001년 : 한국과학기술원 전기 및 전자공학과 석사 졸업
2007년 : 한국과학기술원 전기 및 전자공학과 박사 졸업
2007년~2011년 : University of California Irvine, 박사 후 연구원
2012년~2015년 : 전남대학교 전자컴퓨터공학부 조교수

2016년 ~ 현재 : 전남대학교 전자컴퓨터공학부 부교수

<관심분야> 분산시스템, 소프트웨어 정의 인프라스트럭쳐, 빅데이터 플랫폼, 소셜 네트워킹 시스템, 블락체인, AI기반 CPS