Convolution Neural Network 와 Recurrent Neural Network 를 활용한 네트워크 패킷 분류

임현교*, 김주봉**, 한연희**†
*한국기술교육대학교 창의융합공학협동과정
**한국기술교육대학교 컴퓨터공학과
e-mail : {glenn89, rlawnqhd, yhhan}@koreatech.ac.kr

Network Packet Classification Using Convolution Neural Network and Recurrent Neural Network

Hyun-Kyo Lim*, Ju-Bong Kim**, Youn-Hee Han**
Korea University of Technology and Education, Republic of Korea

요 약

최근 네트워크 상에 새롭고 다양한 어플리케이션들이 생겨나면서 이에 따른 적절한 어플리케이션별 서비스 제공을 위한 패킷 분류 방법이 요구되고 있다. 이로 인하여 딥 러닝 기술이 발전 하면서 이를 이용한 네트워크 트래픽 분류 방법들이 제안되고 있다. 따라서, 본 논문에서는 딥 러닝 기술 중 Convolution Neural Network 와 Recurrent Neural Network 를 동시에 활용한 네트워크 패킷 분류 방법을 제안한다.

1. 서론

최근 새로운 웹 및 모바일의 어플리케이션이 생성되면서 서로 각기 다른 네트워크 트래픽 패킷들이 발생한다. 이로 인하여 네트워크 사업자에서는 각 어플리케이션 별 적절한 서비스를 제공하기 위해 네트워크 상에서의 패킷 분류 기술에 대한 요구가 증가하고 있다.

또한 딥 러닝 기술과 같이 다양한 머신 러닝 기술들이 나타나면서 이를 활용한 패킷 분류에 대한 관심이 증가하고 있다. 따라서 Convolution Neural Network (CNN) 을 이용하여 패킷을 분류하는 방법 [1, 2]이 등장 하였으며, Recurrent Neural Network (RNN) 을 활용하여 패킷의 시간적 흐름을 나타내는 플로우를 분류하는 방법 [3]이 등장하였다. 하지만 해당 기술들은 단순히 패킷을 이미지화 하여 단일 패킷만을 분류 하거나, 플로우만을 분류하기 때문에 실시간으로 제공되는 데이터의 분류에 적합하지 못하다.

본 논문에서는 실시간으로 생성 되는 네트워크 트래픽들에 대하여 CNN 과 RNN 의 모델을 학습하는 부분과 네트워크 트래픽을 분류하는 부분을 나눔으로써 실시간으로 네트워크 트래픽을 분류하는 방법을 제안한다.

2. CNN 및 RNN 모델 학습

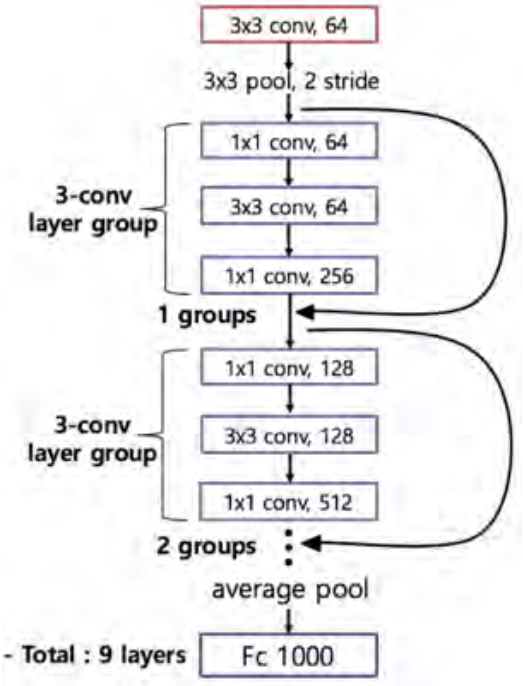
본 논문에서 제안하는 네트워크 패킷 분류 방법은 CNN 과 RNN 학습 모델을 동시에 이용한다. 이를 위하여 각 CNN 과 RNN 모델의 학습이 사전에 이루어져야 한다. 또한, 각 학습 모델은 네트워크 트래픽에 대한 분류이기 때문에 CNN 이용하여 네트워크 트래픽의 패킷단위에서의 분류를 수행하게 된다. 동시에 RNN 의 경우 네트워크 트래픽의 플로우 단위의 분류를 수행하게 된다.

2.1. CNN 모델 학습

CNN 모델의 학습을 위하여 수집된 네트워크 데이터의 전처리 과정이 필요하다. 전처리 과정은 수집된 네트워크 데이터의 어플리케이션별로 5-tuple (source/destination ip, source/destination port, protocol) 이 동일한 패킷들과 3600 초 이내에 발생한 패킷들을 하나의 플로우로 정의하여 나누었다. 나누어진 플로우 데이터는 학습을 위하여 각 플로우 패킷들의 어플리케이션 레이어의 페이로드를 추출하여 이미지로 변환

† 교신 저자: 한연희 (한국기술교육대학교)

이 논문은 2016 년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임 (No. NRF-2016R1D1A3B03933355)



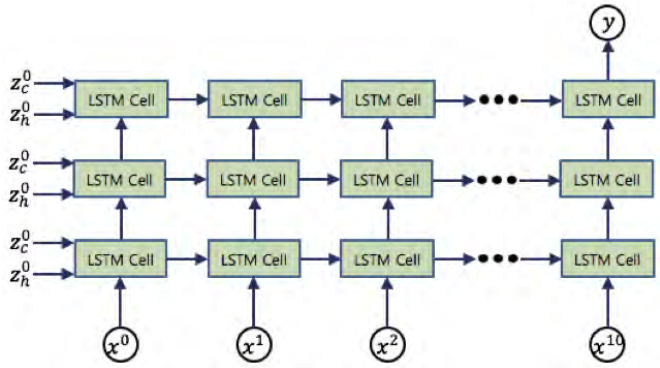
(그림 1) CNN 모델 구조도

하는 과정을 거친다. 이미지 형태로 변환된 어플리케이션 별 플로우의 패킷 데이터는 CNN 모델 학습의 train, validation, test 의 각 입력데이터로 사용된다. 또한, 어플리케이션 별로 one-hot vector 로 라벨링 하여 입력데이터의 정답 데이터로 사용된다. 하지만 학습 데이터로 사용하기에 어플리케이션 별 데이터의 개수가 적은 경우 적은 데이터들을 모아 'Etc.' 라는 라벨링을 하여 학습 모델의 입력데이터로 사용한다.

CNN 의 모델 구조는 (그림 1) 과 같이 Resnet [4]의 주요 구조인 Bottleneck Residual Block 을 3 층으로 만들어 총 11 개의 히든 레이어를 갖는 뉴럴 네트워크로 구성하였다. 구성된 뉴럴 네트워크에 각 패킷의 페이로드 데이터를 입력데이터로 사용하여 학습을 수행하게 된다. 학습이 수행 된 후 학습된 모델을 저장하여 본 논문에서 제안하는 네트워크 패킷 분류에 사용하게 된다.

2.2. RNN 모델 학습

RNN 모델 구조는 CNN 보다 순차적인 데이터 학습



(그림 2) RNN 모델 구조도

	Src. IP	Dst. IP	Src. Port	Dst. Port	Protocol	Pkt-In Count	Lifetime	Label
Flow #1	1.1.1.1	2.2.2.2	2170	8080	TCP	1	3411	-
Flow #2	3.3.3.3	4.4.4.4	1170	2000	UDP	8	122	-
Flow #3	5.5.5.5	6.6.6.6	1099	3211	TCP	17	3600	-
				Flow #1: Packet-In...				
Flow #1	1.1.1.1	2.2.2.2	2170	8080	TCP	19	3600	-
				Flow #1: T^{th} Packet-In → Classify by RNN (Per-Flow-Basis) → 'Foo'				
Flow #1	1.1.1.1	2.2.2.2	2170	8080	TCP	T	3600	'Foo'
				Flow #2: Packet-In...				
Flow #2	3.3.3.3	4.4.4.4	1170	2000	UDP	19	3600	-
				Flow #2: T^{th} Packet-In → Classify by RNN (Per-Flow-Basis) → 'Etc.'				
Flow #2	3.3.3.3	4.4.4.4	1170	2000	UDP	T	3600	'Etc.'
				Flow #3: No Packet-In since 3600 sec. ago				
~~Flow #3~~	~~5.5.5.5~~	~~6.6.6.6~~	~~1099~~	~~3211~~	~~TCP~~	~~17~~	~~0~~	

(그림 3) 네트워크 패킷 데이터 저장 및 분류 결과

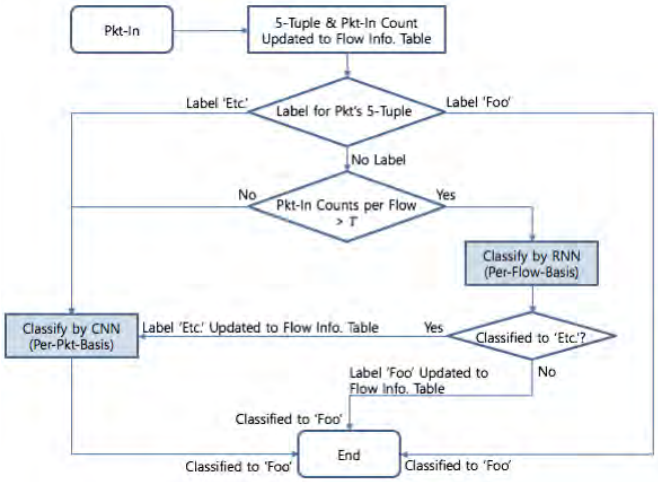
에 있어 뛰어난 성능을 보이고 있다. 따라서, 패킷의 순차적인 정보를 담고 있는 플로우를 RNN 을 통한 네트워크 패킷 분류 학습에 적용 할 것이다.

RNN 모델의 학습의 경우 CNN 과 동일한 네트워크 데이터의 전처리 과정을 거친다. 전처리 과정을 거친 훈련 데이터는 플로우 단위로 다시 나뉘게 된다. 학습을 위하여 선택된 플로우는 기 설정된 플로우당 패킷의 개수에 따라 처음부터 설정된 패킷의 개수만큼 하나의 플로우 데이터 세트로 생성하여 RNN 모델의 입력데이터로 사용된다. 각 플로우는 CNN 과 마찬가지로 one-hot vector 로 라벨링된 train, validation, test 별 정답 데이터를 갖게 된다.

학습을 위한 네트워크 구조는 RNN 의 한 종류인 LSTM (Long Short-Term Memory) 셀[5]을 사용한다. 또한 네트워크는 Multi-Layer 로 구성되어 있으며, (그림 2)는 전체적인 RNN 모델 학습 구조를 나타낸다. 해당 구조를 통해 학습된 모델을 저장하여 제안하는 네트워크 패킷 분류에 사용한다.

3. CNN & RNN 을 이용한 네트워크 패킷 분류 방법

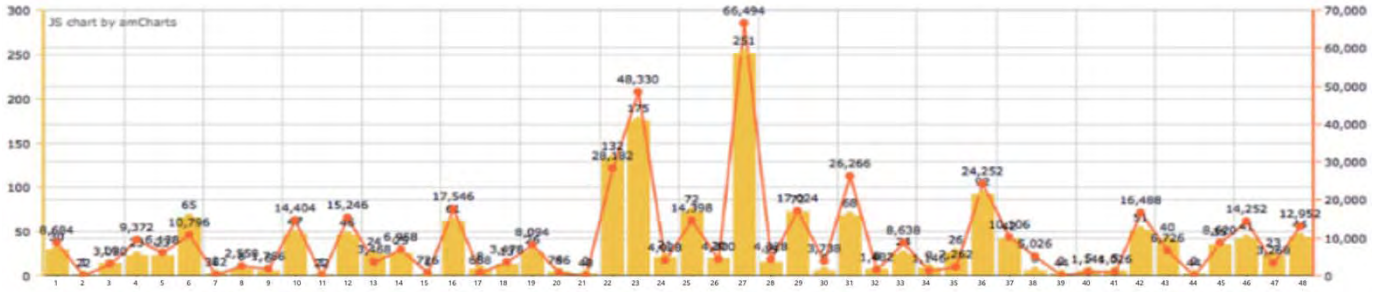
네트워크 패킷 분류를 위하여 2 장에서 수행한 CNN 과 RNN 의 학습된 모델을 이용해 네트워크 트래픽 분류로 로드시켜 각 모델을 동시에 사용한다. (그림 3)



(그림 4) 네트워크 패킷 분류 절차

<표 1> 제안하는 네트워크 패킷 분류를 통해 분류된 어플리케이션 종류

1	wunderlist	7	com.apple	13	googledrivesync.exe	19	vBoxHeadless	25	git	31	Backup and Sync	37	Mail	43	Steam.exe
2	Polaris Office	8	apsd	14	Google Drive	20	vnc	26	n5client.exe	32	CalendarAgent	38	mysql	44	sourcetree.exe
3	vpnkit	9	Docker	15	utorrent	21	dropbox	27	web	33	leagueoflegends	39	VLC	45	skype
4	nosmain	10	helpd	16	Discord.exe	22	teamviewer	28	LINE	34	mysqlworkbench	40	itunes	46	system Idle Process
5	Evernote	11	Notes	17	prl_naptd	23	ssh	29	BitTorrent	35	kakaotalk	41	sourcetree	47	slack
6	TslGame.exe	12	onedrive	18	ClientLoggingService	24	svchost.exe	30	ASDSvc.exe	36	Send Anywhere	42	python	48	com.apple.siri



(그림 5) 패킷 분류 결과

은 패킷 분류 시스템에 들어온 패킷은 플로우의 정의에 따라 5-tuple 과 3600 초 이내에 생성된 패킷일 경우 동일한 패킷으로 간주하여 하나의 플로우로 정의하여 분류한 그림이다. 실제 패킷 분류 시스템의 메모리에 해당 플로우당 패킷의 정보와 분류 결과를 저장하여 관리하게 된다.

패킷의 분류 절차는 (그림 4)와 같으며 패킷 분류 시스템에 하나의 패킷이 들어온 경우 5-tuple 이 동일한 패킷에 대해 3600 초 이내의 패킷이 존재하는 경우 하나의 플로우로 정의 한 후 해당 플로우의 분류된 결과를 먼저 살펴본다. 플로우의 분류 결과가 존재하는 경우 해당 패킷은 이미 분류된 결과로 다시 분류되게 된다. 하지만 동일한 플로우의 패킷이 존재하지 않아 패킷이 되지 않는 경우 먼저 CNN 의 학습 결과를 토대로 해당 패킷의 분류를 수행하게 된다. 이후 5-tuple 이 동일한 패킷이 RNN 모델 학습을 위해 기설정된 플로우당 패킷의 개수인 T 개 만큼의 패킷이 한 플로우에 쌓이게 된 경우 RNN 모델의 학습결과를 토대로 다시 분류를 수행하여 나온 결과로 분류된다.

하지만 실제 네트워크에서 제안하는 패킷 분류 방법을 적용할 경우, 새로운 패킷에 대한 분류를 수행하기 어려운 점이 존재한다. 따라서, 제안하는 패킷 분류 방법에서는 위 CNN 과 RNN 의 모델 학습 과정에서 'Etc.' 라고 하는 새로운 라벨링을 추가하여 학습을 수행한다. 'Etc.'로 라벨링된 패킷은 다른 어플리케이션 별 플로우나 패킷의 개수가 현저히 적어 학습의 데이터로 사용하기 적절하지 않은 패킷들을 모아 'Etc.' 로 라벨링을 한 패킷들 이다. 이러한 패킷들을 이용함으로써 제안하는 네트워크 패킷 분류에서는 Etc 로 분류되는 경우 CNN 모델의 학습 결과를 토대로 해당 패킷이 새로운 패킷인지를 결정하게 된다.

(그림 5)는 패킷의 분류 결과를 나타내며, x 축은 분류를 거쳐 나온 결과를 의미한다 (표 1). 막대그래프의 결과는 분류한 패킷의 개수를 어플리케이션 별로 나타낸다. 선 그래프는 분류한 패킷의 각 페이로드 사이즈를 모두 합한 값이며, 단위는 Byte 이다.

4. 결론

본 논문에서 제안하는 네트워크 패킷 분류 시스템은 CNN 과 RNN 모델을 동시에 사용하여 패킷을 분류하는 방법이다. 또한 학습 부분과 패킷 분류 부분을 나눔으로써 모델의 학습이 이루어짐과 동시에 패킷이 분류 되는 효과를 가져올 수 있다.

참고문헌

[1] W. Wang, M. Zhu, X. Zeng, X. Ye, and Y. Sheng, "Malware traffic classification using convolutional neural network for representation learning," 2017 International Conference on Information Networking (ICOIN), Da Nang, pp. 712-717, 2017.

[2] 김주봉, 임현교, 허주성, 한연희, "Convolutional Neural Network 을 활용한 패킷 페이로드 기반 네트워크 트래픽 분류," 2017 년도 한국정보처리학회 춘계학술발표대회, 2017. 04.

[3] 임현교, 김주봉, 허주성, 권도형, 한연희, "Recurrent Neural Network 을 이용한 플로우 기반 네트워크 트래픽 분류, " 한국정보처리학회 2017 년도 추계학술발표대회, 2017. 11.

[4] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, pp. 770-778, 2017.

[5] H. Sak, Andrew Senior, F. Beaufays, "Long Short-Term Memory Recurrent Neural Network Architectures for Large Scale Acoustic Modeling," Proceedings of the Annual Conference of the International Speech Communication Association (INTERSPEECH), pp 338-342, Jan. 2014.