https://doi.org/10.7840/kics.2023.48.5.581

CNN 기반 HTTPS 트래픽의 서비스 및 응용 단위 다중 레이블 분류

김 보 선•, 최 정 우*, 박 지 태*, 백 의 준*, 김 명 섭°

Multi-Label Classification of CNN-Based HTTPS Traffic by Service and Application Level

Boseon Kim•, Jung-Woo Choi*, Jee-Tae Park*, Ui-Jun Baek*, Myung-Sup Kim°

요 약

웹 응용 프로그램은 HTTP 프로토콜의 보안 문제로 HTTPS 프로토콜을 사용한다. HTTPS 응용 트래픽 분류를 위해 다양한 연구가 진행되고 있으나, 암호화로 인해 분류하기 어렵다. 이를 해결하기 위한 방법들은 multi-class 분류이며 서비스 단위로 응용 트래픽을 분류한다. 웹 서비스는 여러 응용 프로그램의 조합으로 구성되며, 트래픽 또한 여러 응용 프로그램의 조합으로 하나의 서비스를 이룬다. 기존 연구와 같이 서비스 단위로만 분류할 경우, 서비스 내 여러 응용 프로그램의 트래픽이 섞여 미탐 혹은 오탐할 수 있다. 따라서 우리는 multi-label 분류를 사용하여 플로우 별로 서비스 단위인 주라벨과 응용 프로그램 단위인 부라벨을 정의하고, 합성 곱 신경망을 설계하여 6개의 서비스와 서비스 내 14개의 응용 프로그램을 분류한다. 6개의 서비스 분류 정확도는 100%, 14개의 응용 프로그램 분류 정확도는 85% 성능으로 서비스 단위와 응용 프로그램 단위로 트래픽을 분류하였다.

키워드 : 트래픽 분류, HTTPS 트래픽 분류, 응용 트래픽 분류, 합성 곱 신경망, 다중 레이블 분류

Key Words : Traffic Classification, HTTPS Traffic Classification, Application, Traffic Classification, Convolutional Neural Network, Multi-label Classification

ABSTRACT

Various studies are being conducted to classify HTTPS-applied traffic, but it is difficult to classify due to encryption. Methods to solve this problem are multi-class classification and application traffic is classified by service level. A web service consists of a combination of several applications, and traffic also forms a service with a combination of several applications. As in previous studies, if classified only as a service level, traffic from various applications in the service can be mixed in the service and misdetected or falsely detected. Therefore, we use multi-label classification to define the main label, which is the service level, and the sub label, which is the application level, for each flow, and to design a synthetic product neural network to classify six services and 14 applications within the service. Traffic was classified by service level and application level with 6 service classification accuracy of 100% and 14 application classification accuracy of 85% performance.

※ 이 논문은 2020년도 산업통상자원부 및 한국산업기술평가관리원(KEIT) 연구비 지원에 의한 연구(No.20008902, IT비용 최소화를 위한 5채널 탐지기술 기반 SaaS SW Management Platform(SMP) 개발)이고, 2021년도 교육부의 재원으로 한국연구재단의 지원을 받아 수행된 지자체-대학 협력기반 지역혁신 사업의 결과로 수행되었음(2021RIS-004)

• First Author : Department of Computer and Information Science, Korea University, boseon12@korea.ac.kr, 학생회원

° Corresponding Author : Department of Computer and Information Science, Korea University, tmskim@korea.ac.kr, 종신회원

* Department of Computer and Information Science, Korea University, choigoya97@korea.ac.kr; pjj5846@korea.ac.kr, 학생회원; pb1069@korea.ac.kr, 학생회원

논문번호 : 202209-207-B-RN, Received September 8, 2022; Revised December 15, 2022; Accepted January 30, 2023

Ⅰ. 서 론

IT 기술이 발전함에 따라 사용자들은 웹 사이트를 통해 여러 정보를 얻고 다른 사용자들과 소통하고 있다. 인터넷은 일상생활에서 매우 중요한 서비스이며, 코로나-19의 영향으로 인해 인터넷 이용 시간이 꾸준히 증가하고 있다. 대표적으로 사회적 거리두기 기간의 장기화로 인해 인터넷을 통한 온라인 소통 및 정서적 교류 활동으로 메신저 및 SNS 이용률이 지속적으로 증가하는 추세를 보이고 있다. 또한 실내 활동 시간이 길어지면서 동영상 서비스 이용 또한 큰 폭의 증가세를 유지하고 있다. 또한 소비와 경제 활동의 온라인 및 비대면화 흐름이 반영되어 인터넷 쇼핑과 인터넷 뱅킹 이용이 급격히 증가하였다[1]. 과학기술정보통신부와 한국지능정보사회진흥원에서 조사한 국내 인터넷 이용률 통계 자료는 그림 1과 같다. 이와 같이 인터넷 이용률이 증가 하면서 SNS, 콘텐츠, 쇼핑, 뱅킹과 같은 여러 산업의 응용 프로그램 종류 및 수 또한 증가하고 있다.

이에 네트워크의 효과적인 운용과 관리를 위해 네트워크 트래픽 분석이 매우 필요하며, 네트워크 관리에 있어 중요성이 강조되고 있다. 네트워크 분석 중 네트워크 트래픽 분류는 서비스 품질 제어, 리소스 사용 계획, 침입 탐지와 같은 분야에서 매우 중요하다. 분류의 중요한 전제는 서로 구별하는데 사용할 수 있는 다양한 응용 트래픽에 대한 특징이 있다는 것이다. 트래픽이 암호화 되면서 암호화로 인해 사용 가능한 유효한 특징이 변경됨에 따라 분류 방법 또한 변하고 있다[2,3]. 다양한 트래픽 분류 연구들이 진행되고 있고[4], 이를 관련 연구에 기술한다. 새로운 응용 프로그램의 등장과 업데이트등 응용 프로그램의 변화로 인해 응용 트래픽 분류가 어려워지고 있다. 이에 따라 기존 응용 트래픽 분류 연구의 한계점을 극복하고 보다 높은 정확도를 보이는 분석 방법이 필요하다.

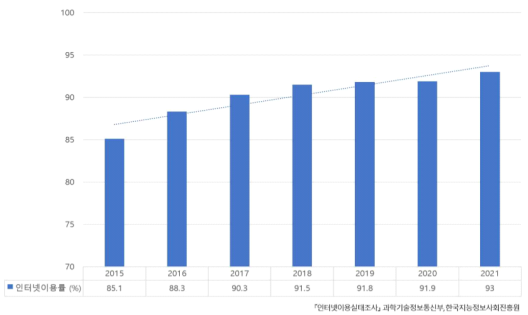


그림 1. 년도 별 인터넷 이용률
Fig. 1. Internet usage rate by year

HTTPS 통신 기반 웹 서비스는 여러 응용 프로그램의 조합으로 구성되어 있다. 따라서 하나의 서비스 트래픽 내에 하나 이상의 응용 프로그램 트래픽이 혼합되어있다. 예를 들어 사용자가 Naver 웹 서비스에 접속하여 Naver 서비스 내 googlefonts나 google-analytics를 사용하는 것은 사용자가 Naver 웹 서비스에 접속하여, Naver 서비스 내 Google 응용 프로그램과 통신함을 의미한다. 이는 Naver 서비스 내 Google 하위 응용 프로그램이 있음을 나타낸다. 기존 HTTPS 트래픽 분류 연구는 HTTPS 트래픽을 오직 서비스 단위로 분류한다[5,6]. 대부분의 연구는 서비스 단위와 응용 프로그램단위로 개별 분류하지 않기 때문에 서비스 내 여러 응용프로그램 트래픽이 혼합되어 오분류 가능성이 높다. 또한 서비스 단위만을 통해 정의할 경우, 트래픽 내 속해있는 또 다른 응용 프로그램에 의해 차단 가능성이 존재한다. 따라서 본 논문은 서비스 단위인 주라벨과 응용프로그램 단위인 부라벨을 이용한 HTTPS 트래픽 분류방법을 제안한다. 기존 연구에서는 Naver 서비스 내 Google 하위 응용 프로그램과 통신하는 플로우를 Naver 서비스로 분류한다. 하지만 제안하는 방법은 해당 플로우를 Naver 서비스와 Google 응용 프로그램으로 분류한다. 우리는 HTTPS 트래픽의 플로우별로 서비스 단위인 주라벨과 응용 프로그램 단위인 부라벨을 정의하고, 합성곱 신경망을 설계하여 6개의 주라벨과 14개의 부라벨로 응용 트래픽을 다중 레이블 분류를 수행한다.

본 논문의 구성은 다음과 같다. 서론에 이어, 2장에서 기존 트래픽 분류 방법들에 대해 기술하고 3장에서 해결하고자 하는 문제 정의 및 방법론에 대해 설명한다. 4 장에서 제안하는 방법을 통한 응용 트래픽 분류 실험 및 결과를 통해 제안하는 방법의 타당성을 증명한다. 마지막으로 5장에서 결론 및 향후 연구에 대해 기술한다.

Ⅱ. 관련 연구

첫 번째는 포트 기반 방법으로 가장 기초적인 응용 트래픽 분류 방법이다. 포트 번호를 통해 응용 트래픽을 분류하는 방법이며, IANA(Internet Assigned Numbers Authority, 인터넷 할당 번호 관리 기관)를 통해 잘 알려진 포트(well-known port)를 확인하여 어떤 응용 프로그램 트래픽인지 확인하는 방법이다[7]. 단순하고 쉬운 방법이지만, 응용 프로그램이 정해진 포트 번호를 사용하지 않거나 동적 포트 번호를 사용

할 경우 트래픽을 오분류 할 가능성이 존재한다.

두 번째는 페이로드 기반 방법으로 특정 응용 프로그램에서 발생한 트래픽을 분석해 다른 응용 프로그램과 구분 할 수 있는 시그니처라고 하는 특정 응용 프로그램만의 특징을 추출하여 트래픽을 분류하는 방법이다[8,9]. 시그니처는 수작업으로 추출하기 때문에 효율성 부분에서 좋지 않으며, 암호화 트래픽이 증가하며 시그니처를 추출하기 어려워졌다. 또한 시그니처가 정의되어 있는 응용 프로그램에 대해 정확한 트래픽 분류 방법이지만, 시그니처가 정의되어 있지 않은 응용 프로그램에 대해서는 트래픽을 구분하기 어렵다. 따라서 새로운 응용 프로그램의 경우 그에 해당하는 시그니처를 정의하기 전까지 분류가 불가능하다.

세 번째는 통계 정보 기반 방법으로 응용 트래픽의 통계 정보를 이용한 분류 방법이다. 패킷 간의 시간, 패킷 크기와 같은 통계적 분석을 통 해 특징을 생성하여 응용 프로그램을 분류하는 방법이다.

네 번째는 머신러닝 기반 방법으로 응용 프로그램별 특징을 추출하여 머신러닝 알고리즘을 통해 응용 프로그램을 분류하는 방법이다[10-13]. 패킷 크기, 패킷 간의 시간 등 통계 정보를 추출하여 이를 학습하여 분류한다. 이는 응용 트래픽을 분류할 때, 다른 분석 방법에 비해 높은 분석률을 제공하지만, 트래픽이 학습된 응용들로 분류가 진행되기 때문에 새로운 응용 프로그램이 나올 경우 유연하게 대처하지 못한다. 또한 통계 정보는 사람이 패킷 단위 혹은 플로우 단위에서 추출하기 때문에, 이 과정에서 어떠한 특징이 응용 프로그램을 잘 나타내는지 알 수 없다.

Ⅲ. 문제 정의

본 장에서 기존 응용 트래픽 분류 방법에 대해 문제를 정의하고 이를 해결하기 위한 방법론에 대해 기술한다. 현재 다양한 응용 트래픽 분류 방법이 연구되고 있지만, 여러 문제점들이 존재한다. 첫째, 머신러닝 기반분류 방법은 다른 분석 방법들에 비해 높은 분류 정확도를 보이지만 새로운 응용 프로그램에 대해 유연하게 대처하지 못하며, 사람이 직접 통계 정보를 추출하기 때문에 사람에 의존적이다[14]. 둘째, SNI (Server Name Indication) 기반 분류 방법은 시각적으로 확인 가능한 분석 방법이지만 SNI는 암호화가 적용되기 전 단계로 SNI를 변경하여 접근 제한 웹 사이트를 우회할 경우 SNI 기반 분류 방법엔 효과적이지 않다. 또한 SNI는 보안상의 문제로 ESNI(Encrypted Server Name Indication), ECH(Encrypted ClientHello)와 같이 SNI 암호화 기술이 확대되고 있기 때문에 SNI 기반 트래픽 분류로는 응용 트래픽을 분류하기 점점 어려워진다[13]. 셋째, 웹 서비스는 여러 응용 프로그램의 조합으로 구성되어 있다. 현재 연구들은 서비스 내 응용 프로그램 유무를 고려하지 않고, 하나의 라벨을 사용하여 서비스 단위로만 분류한다. 하지만 알라딘 서비스 트래픽 내 SNI를 추출하면 그림 2와 같이 알라딘 외에도 구글, 페이스북, 네이버, 광고 회사와 같은 다른 응용 프로그램의 트래픽이 발생한 것을 알 수 있다. 알라딘 웹 서비스를 이용하여 트래픽을 수집하였으나 그림 2와 같이 로그인 과정에서 혹은 광고에 의해 다른 응용 프로그램의 트래픽 또한 알라딘 웹 서비스에 접속했을 때 발생하였다. 그림 3은 응용 별 패킷 흐름을 나타낸다. 알라딘 서비스에서 발생한 트래픽 중 SNI가 www.aladin.co.kr인 플로우는 알라딘 응용 프로그램이다. 네이버 서비스에서 발생한 트래픽 중 SNI가 www.naver.com인 플로우는 네이버 응용 프로그램이다. 하지만 알라딘 서비스에서 발생한 트래픽

Server Name
stats.g.doubleclick.net
tracking.crazyegg.com
update.googleapis.com
vars.hotjar.com
vars.hotjar.com
vars.hotjar.com
wcs.naver.com
wcs.naver.net
www.aladin.co.kr
www.aladin.co.kr
www.facebook.com
www.google-analytics.com
www.google-analytics.com
www.google.co.kr
www.google.com
www.googleadservices.com
www.googletagmanager.com
www.googletagmanager.com
www.gstatic.com

그림 2. 알라딘 서비스 내 응용 프로그램 리스트
Fig. 2. List of applications in Aladin service

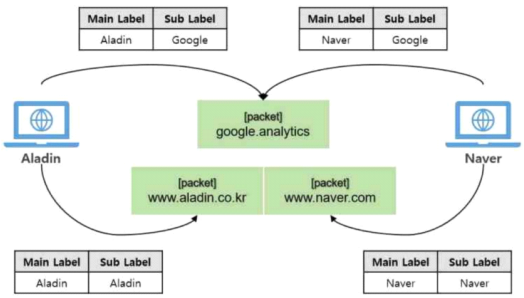


그림 3. 응용 별 패킷 흐름
Fig. 3. Packet flow by application

표 1. 응용 트래픽 분류를 위한 기존 라벨링 방법과 제안하는 라벨링 방법
Table 1. Comparison of the previous labeling method and the proposed labeling method for application traffic classification

Service	Application (by SNI)	Previous Label (for Service)	Proposed Label	
			Main Label (for Service)	Sub Label (for Application)
Aladin	www.aladin.co.kr	Aladin	Aladin	Aladin
	image.aladin.co.kr		Aladin	Aladin
	fonts.gstatic.com		Aladin	Google
Google	www.google.co.kr	Google	Google	Google
	fonts.gstatic.com		Google	Google
	i.ytimg.com		Google	Youtube
Youtube	youtube.com	Youtube	Youtube	Youtube
	i.ytimg.com		Youtube	Youtube
	accounts.google.com		Youtube	Google

중 SNI가 google-analytics인 플로우는 알라딘 서비스로 정의하기 어렵다. 알라딘에서 발생한 패킷이지만 알라딘 고유의 패킷이 아니기 때문이다. 하지만 구글 서비스로 정의하기에는 알라딘 웹 서비스에 접속하면 항상 나오는 트래픽의 일부이기 때문에 온전한 구글 서비스로 정의하기 어렵다. 따라서 우리는 이러한 문제점을 해결하기 위해 주라벨과 부라벨을 사용하여 서비스 단위와 응용 프로그램 단위로 트래픽을 분류한다. 알라딘 서비스에서 발생한 구글 응용 프로그램 플로 우의 주라벨을 알라딘, 부라벨을 구글로 정의한다. 주라벨과 부라벨 총 두 가지 라벨을 사용하여 분류한 이유는 두 가지가 있다. 첫째, 서비스 단위로 분류할 경우, 분류 성능이 저하된다. 예를 들어 알라딘 웹 서비스를 통해 수집한 트래픽 전체를 표 1의 기존과 같이 알라딘으로 분류하여 학습할 경우, 광고나 구글 분석과 같은 여러 응용 프로그램의 특징과 섞여 분류 성능이 저하된다. 둘째, 세부적으로 응용 프로그램 차단이 불가능하다. 예를 들어 구글 서비스를 차단한다고 가정하며, 표 1에서 fonts.gstatic.com은 구글 api이다. 알라딘 웹 서비스를 통해 수집한 트래픽 전체를 표 1의 기존과 같이 알라딘으로 분류하여 학습할 경우, 알라딘 서비스에 섞인 fonts.gstatic 패킷을 구글 서비스로 인식하여 알라딘 서비스를 차단할 가능성이 높다. 하지만 표 1의 제안하는 라벨과 같이 주라벨과 부라벨을 사용할 경우, 구글 서비스를 차단해야하면 주라벨과 부라벨이 모두 구글인 경우를 차단하면 된다. 알라딘 서비스의 구글 응용 프로그램은 주라벨이 알라딘이고 부라벨이 구글이기 때문에 차단 범위에 속하지 않게 된다. 따라서 본 논문은 응용트래픽의 플로우 별로 서비스 단위인 주라벨과 응용프로그램 단위인 부라벨을 정의하고, 합성 곱 신경망을 설계하여 서비스 단위와 응용 프로그램 단위로 응용 트래픽을 다중 레이블 분류한다.

Ⅳ. CNN 기반 다중 레이블 분류 방법

본 장에서 제안하는 응용 트래픽 분류 방법에 대해 설명한다.

4.1 분류 방법 구조도

본 논문에서 제안하는 응용 트래픽 분류 모델의 설계 과정은 그림 4와 같다. 전처리 단계, 분류 단계로 구성 되어 있다. 플로우마다 서비스 단위로 주라벨을 정의하고 응용 프로그램 단위로 부라벨을 정의하여,

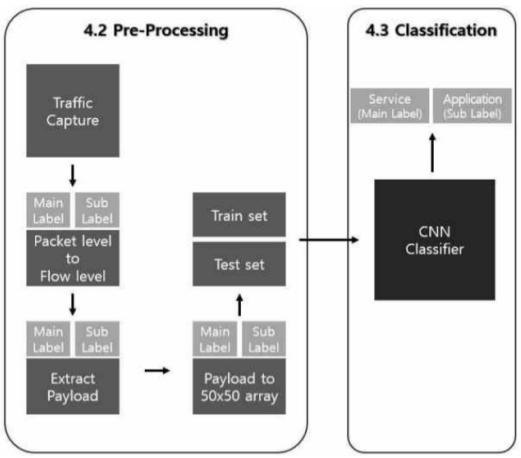


그림 4. 응용 트래픽 분류 모델 설계 과정
Fig. 4. Application traffic classification model design process

트래픽을 서비스 단위와 응용 프로그램 단위로 분류한다.

4.2 전처리

응용 트래픽 분류 모델 설계 과정은 그림 4와 같다. Microsoft의 Network Monitor 프로그램의 Traffic Capture 기능을 이용해 웹 서비스 트래픽을 수집한다. 수집한 트래픽은 패킷 단위로 구성된 cap 파일 형태로 저장된다. 트래픽 수집 단계에서 나온 cap 파일의 패킷 단위를 플로우 단위(Flow with Packet)로 변환한다. Flow with Packet 형태는 Client PC와 Application Server 간의 하나의 세션에서 발생하는 패킷들의 집합 이다. 주라벨은 서비스 이름(A)으로 정의한다. 서비스 (A) 트래픽에서 서비스 고유 응용 프로그램의 부라벨을 고유 응용 프로그램(A)의 이름으로 정의하고, 서비스 (A) 트래픽에서 또 다른 응용 프로그램(B)의 플로우는 부라벨을 또 다른 응용 프로그램(B)의 이름으로 라벨링 한다. Flow with Packet에서 페이로드를 추출한다. 플로우 간 처음 연결 시 발생하는 패킷에서 중요한 정보를 포함하기 때문에 플로우마다 2500의 크기만큼 페이로드를 자른다. 페이로드 크기를 일정하게 맞추기 위해 2500보다 작은 경우 0으로 채운다. 추출된 데이터는 0 ~ 255 사이의 값을 가지며, 모든 전처리 단계를 거치면 2500 크기의 페이로드를 50x50 크기의 2D CNN 입력값 으로 만든다.

4.3 분류 모델

제안하는 분류 모델은 그림 5와 그림 6과 같다. 그림 5와 그림 6과 같은 분류 모델 1과 분류 모델 2는 주라 벨과 부라벨을 적용하여 서비스와 응용 프로그램으로 트래픽을 분류하는 모델이다. 두 모델은 공통적으로 InputLayer에 50x50 배열인 입력 값을 갖는다. 분류 모델 1은 2D CNN에서 주로 사용하는 일반적인 직렬 모델이며, 분류 모델 2는 2D CNN을 응용한 병렬 모델이 다. 두 가지 CNN 모델로 주라벨과 부라벨을 통해 서비스 단위와 응용 프로그램 단위로 트래픽을 분류한다. 분류 모델 1은 2개의 Convolutional layer를 거쳐 서비스단위 Dense와 응용 프로그램 단위 Dense를 통해 트래픽을 분류한다. 서비스 단위 Dense는 6개의 서비스로 트래픽을 분류하며, 응용 프로그램 단위 Dense는 14개의 응용 프로그램으로 트래픽을 분류한다. 분류 모델2 는 병렬 모델로 각각 2개의 Convolutional layer를 거쳐 나온 결과값을 Concatenate를 통해 합치고 다시 2개의 Dense layer를 통해 결과를 도출한다. 서비스 단위 Dense는 6개

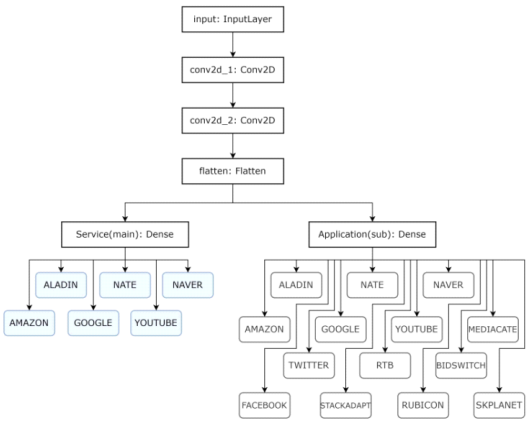


그림 5. 제안하는 응용 트래픽 분류 모델 1
Fig. 5. Application traffic classification model 1

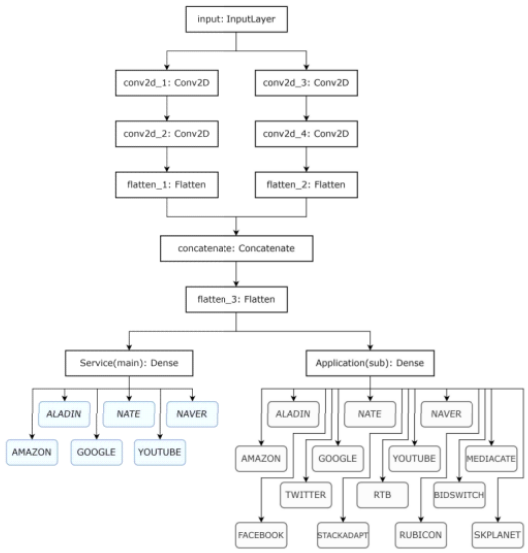


그림 6. 제안하는 응용 트래픽 분류 모델 2
Fig. 6. Application traffic classification model 2

의 서비스로 트래픽을 분류하며, 응용 프로그램 단위 Dense는 14개의 응용 프로그램으로 트래픽을 분류한다. 우리는 제안하는 두 가지 모델을 통해 6개의 서비스와 14개의 응용 프로그램을 분류하는데 적합하다는 것을 검증하기 위해 5장 실험을 통해 검증한다.

V. 실험 및 결과

본 장에서 제안하는 방법을 검증하기 위한 실험을 진행한다. 6개 응용 트래픽에 대한 6개 서비스(주라벨)와 14개 응용 프로그램(부라벨)을 대상으로 CNN 기반 응용 트래픽 분류 실험을 진행하여 제안하는 방

법론의 적합성을 검증한다.

5.1 데이터 셋

Microsoft의 Network Monitor 프로그램을 통해 수집 한 6개의 웹 서비스 트래픽의 정보는 표 2와 같다. 알라딘, 네이트, 네이버, 아마존, 구글, 유튜브에서 수집한모든 데이터 셋의 주라벨과 부라벨은 표 2와 같다. 알라딘 서비스에서 나온 구글 응용 프로그램 플로우의 주라벨은 0, 부라벨은 1이고 구글 서비스에서 나온 고유의 구글 응용 프로그램 플로우의 주라벨은 4, 부라벨은 1이다.

표 2. 수집한 웹 응용 트래픽 데이터 셋 정보
Table 2. Dataset information of web service traffic

Web Application Traffic				# Flow
Service	Application	Main Label	Sub Label	
Aladin	Aladin	0	0	997
	Google		1	
	Facebook		2	
	Naver		3	
Nate	Nate	1	4	1202
	Mediacategory		5	
	stackadapt		6	
	rtb		7	
	rubicon		8	
	skplanet		9	
Naver	Naver	2	3	933
	Google		1	
Amazon	Amazon	3	10	951
	Twitter		11	
	Bidswitch		12	
Google	Google	4	1	784
	Youtube		13	
Youtube	Youtube	5	13	897
	Google		1	

5.2 실험

6개의 서비스 트래픽 데이터 셋 중 60%는 학습용 데이터로 사용하였고 20%의 데이터 셋은 테스트용 데이터로 사용하였으며, 나머지 20%의 데이터 셋은 검증용 데이터로 사용하였다. 과적합을 방지하기 위해 EarlyStopping 콜백 함수를 사용하여 적절한 시점에 학습을 조기 종료하도록 하였다. 각 실험은 10번 반복하였고, 학습 정확도와 테스트 정확도는 최대 정확도를 평균 낸 값이다. 제안하는 CNN 분류 모델 1과 분류 모델 2의 검증을 위해 그림 7 모델인 분류 모델 3과 비교한다. 분류 모델3은 응용 프로그램 단위로 트래픽을 분류하는 모델로, 부라벨 하나를 사용하여 분류를 수행하는 모델이다. 분류 모델 3은 2D CNN에서 일반적으로 사용하는 직렬모델로 2개의 Convolutional layer를 거쳐 Dense를 통해 응용 프로그램 단위로 14개의 응용 프로그램을 분류한다. 비교를 위해 세 가지 모델을 대상으로 학습을 수행하고, 이를 기반으로 분류 검증 실험을 수행한다. 세 가지 모델을 통해 응용 트래픽 분류 실험을 진행하였고, 학습 및 검증 결과는 표 3에 기술하였다. 제안하는 모델의 적합성을 검증하기 위한 비교 모델로 분류 모델 3은 14개의 응용 프로그램을 83.87% 정확도로 분류한다. 제안하는 분류 모델 1은 6개의 서비스를 100% 정확도로 분류하며, 14개의 응용 프로그램을 84.92% 정확도로 분류한다. 제안하는 분류 모델 2는 6개의 서비스를 100% 정확도로 분류하며, 14개의 응용 프로그램을 85.12% 정확도로 분류한다. 이를 통해 응용 프로그램 단위로 트래픽을 분류하는 분류 모델 3에 비해 서비스 단위와 응용 프로그램 단위로 트래픽을 분류하는 분류 모델 1과 분류 모델 2의 분류 실험 정확도보다 주라벨과 부라벨을 통해 응용 트래픽을 분류한 모델 2와 모델 3의

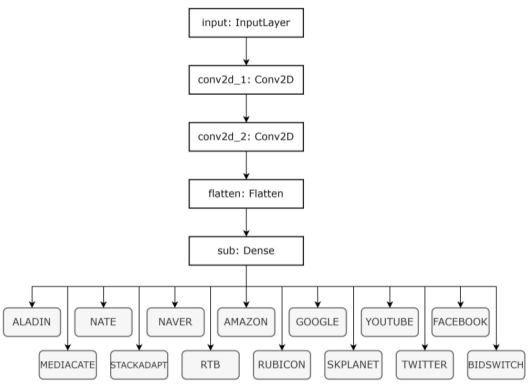


그림 7. 응용 트래픽 분류 모델 3 (비교모델)
Fig. 7. Application traffic classification model 3

표 3. 모델 별 응용 트래픽 분류 실험 결과
Table 3. Results of application traffic classification experiment by model

Model No.	Train Acc		Test Acc	
	Main	Sub	Main	Sub
1	100	85.81	100	84.92
2	100	86.74	100	85.12
3	-	85.07	-	83.87

분류 실험 정확도가 더 높은 것을 알 수 있다. 트래픽을 서비스 단위와 응용 프로그램 단위로 분류함으로써 서비스 고유의 응용 프로그램이 아닌 또 다른 응용 프로그램 또한 분류가 가능해졌고, 여러 응용 프로그램의 특징과 섞여 분류 성능이 저하되는 문제점을 해결하였다. 제안하는 분류 모델 1과 분류 모델 2를 통해 특정 웹 서비스 뿐 아니라 웹 서비스의 특정 응용 프로그램을 세부적으로 관리할 수 있게 되었다. 서비스 내 여러 응용 프로그램에 의해 차단되거나 오탐 혹은 미탐할 가능성 또한 저하되었다. 트래픽을 서비스와 응용 프로그램 단위로 분류함으로써 웹 서비스가 여러 응용 프로그램의 트래픽으로 혼합된 점에 대한 문제점이 해결되었다.

Ⅵ. 결 론

본 논문에서는 기존 HTTPS 응용 트래픽 분류 방법의 한계점에 대해 분석하여 해결하고자 합성 곱 신경망 기반 서비스 및 응용 단위별 트래픽 분류 방법을 제안하였다. 기존 연구에서는 하나의 서비스에서 나오는 모든 응용 프로그램을 하나의 서비스로 정의하였지만 트래픽 차단 복잡성 및 분류 성능 저하에 대한 가능성을 고려하여, 플로우마다 서비스(주라벨)와 응용 프로그램(부라벨)을 정의하여 분류 실험을 진행하였다. 다중 레이블을 적용하여 합성 곱 신경망을 설계하였고 6개의 서비스와 14개의 응용 프로그램을 분류하는데 있어 적절하다는 것을 검증하였다. 향후 SNI 정보가 암호화되어도 해당 분류 모델을 통해 정확한 트래픽 분류가 이뤄질 수 있을 것으로 기대하며, 주라벨과 부라벨을 사용하여 특정 서비스에서 발생하는 여러 응용 프로그램 또한 분류가 가능해졌다.

향후 연구로는 응용 프로그램 분류 또한 높은 성능을 보이는 모델 설계를 목적으로 여러 매개변수를 사용하여 분류를 수행하고자 한다. 입력 크기 혹은 입력 형태와 같은 여러 매개변수를 변경하고, 패킷의 시계열성을 고려한 모델과 결합하여 실험할 계획이다. 또한 크롬에서 수집한 데이터뿐 아니라 엣지나 파이어폭스와 같은 여러 브라우저에서 같은 응용 트래픽을 수집하여 브라우저 또한 분류하고자 한다.

References

[1] Ministry of Science and ICT, National Information society agency, *2021 Survey on the Internet Usage* (2022), Retrieved Aug., 31, 2022. (https://www.nia.or.kr/site/nia_kor/ex/bbs/List.do?cbIdx=99870)

[2] Y. Xue, D. Wang, and L. Zhang, “Traffic classification: Issues and challenges,” in *2013 IEEE ICNC*, pp. 545-549, 2013. (https://doi.org/10.1109/ICCNC.2013.6504144)

[3] P. Velan, M. Čermák, P. Čeleda, and M. Drašar, “A survey of methods for encrypted traffic classification and analysis,” *Int. J. Netw. Manag.*, vol. 25, no. 5, pp. 355-374, 2015. (https://doi.org/10.1002/nem.1901)

[4] F. Zaki, et al., “GRAIN: Granular multi-label encrypted traffic classification using classifier chain,” *Computer Networks*, vol. 213, no. 109084, Aug. 2022. (https://doi.org/10.1016/j.comnet.2022.109084)

[5] U.-J. Baek, B. Kim, J.-T. Park, J.-W. Choi, and M.-S. Kim, “MISCNN: A novel learning scheme for CNN-Based network traffic classification,” *2022 23rd IEEE APNOMS*, pp. 01-06, 2022. (https://doi.org/10.23919/APNOMS56106.2022.9919961)

[6] W. Wang, et al., “HAST-IDS: Learning hierarchical spatial-temporal features using deep neural networks to improve intrusion detection,” *IEEE Access*, vol. 6, pp. 1792-1806, 2017. (https://doi.org/10.1109/ACCESS.2017.2780250)

[7] IANA, *IANA port number list*, Retrieved Aug., 31, 2022. (https://www.iana.org/)

[8] K.-S. Shim, Y.-H. Goo, S.-H. Lee, and M.-S. Kim, “Automatic payload signature update system for classification of recent network applications,” *J. KICS*, vol. 42, no. 1, pp. 98-107, 2017. (https://doi.org/10.7840/kics.2017.42.1.98)

[9] A. Callado, C. Kamienski, G. Szabo, B. P. Gero, J. Kelner, S. Fernandes, and D. Sadok, “A survey on internet traffic identification,” *IEEE Commun. Surv. & Tuts.*, vol. 11, no. 3, pp. 37-52, 2009. (https://doi.org/10.1109/SURV.2009.090304)

[10] T. T. T. Nguyen and G. Armitage, "A survey of techniques for internet traffic classification using machine learning," *IEEE Commun. Surv. & Tuts.*, vol. 10, no. 4, pp. 56-76, 2008. (https://doi.org/10.1109/SURV.2008.080406)

[11] F. Pacheco, E. Exposito, M. Gineste, C. Baudoin, and J. Aguilar, "Towards the deployment of machine learning solutions in network traffic classification: A systematic survey," *IEEE Commun. Surv. & Tuts.*, vol. 21, no. 2, pp. 1988-2014, 2018. (https://doi.org/10.1109/COMST.2018.2883147)

[12] M. Shafiq, X. Yu, A. A. Laghari, L. Yao, N. K. Karn, and F. Abdessamia, "Network traffic classification techniques and comparative analysis using machine learning algorithms," *2016 2nd IEEE ICCC*, pp. 2451-2455, 2016. (https://doi.org/10.1109/CompComm.2016.7925139)

[13] D. Shamsimukhametov, A. Kurapov, M. Liubogoshchev, and E. Khorov, "Is encrypted clienthello a challenge for traffic classification?," *IEEE Access*, vol. 10, pp. 77883-77897, 2022. (https://doi.org/10.1109/ACCESS.2022.3191431)

[14] S. Rezaei and X. Liu, "Deep learning for encrypted traffic classification: An overview," *IEEE Commun. Mag.*, vol. 57, no. 5, pp. 76-81, 2019. (https://doi.org/10.1109/MCOM.2019.1800819)

김 보 선 (Boseon Kim)

2020년 : 고려대학교 컴퓨터정보학과 학사

2020년~현재 : 고려대학교 컴퓨터정보학과 석사과정

<관심분야> 네트워크 관리 및 보안, 트래픽 모니터링 및 분석

최 정 우 (Jeong-Woo Choi)

2022년 : 고려대학교 컴퓨터정보학과 학사

2022년~현재 : 고려대학교 컴퓨터정보학과 석사과정

<관심분야> 네트워크 관리 및 보안, 트래픽 모니터링 및 분석

박 지 태 (Jee-Tae Park)

2017년 : 고려대학교 컴퓨터정보학과 학사

2017년~현재 : 고려대학교 컴퓨터정보학과 석박사통합과정

<관심분야> 네트워크 관리 및 보안, 트래픽 모니터링 및 분석

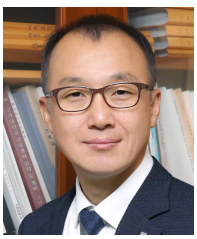
백 의 준 (Ui-Jun Baek)

2018년 : 고려대학교 컴퓨터정보학과 학사

2018년~현재 : 고려대학교 컴퓨터정보학과 석박사통합과정

<관심분야> 네트워크 관리 및 보안, 트래픽 모니터링 및 분석

김 명 섭 (Myung-Sup Kim)



1998년 : 포항공과대학교 전자계산학과 학사

2000년 : 포항공과대학교 전자계산학과 석사

2004년 : 포항공과대학교 전자계산학과 박사

2006년 : Dept. of ECS, Univ of Toronto Canada

2006년~현재 : 고려대학교 컴퓨터정보학과 교수

<관심분야> 네트워크 관리 및 보안, 트래픽 모니터링 및 분석, 멀티미디어 네트워크