https://doi.org/10.22670/knom.2023.26.1.61

앙상블 모델에 임계값을 적용한 3단계 순차적 응용 트래픽 분류 시스템

이 민 성◆, 백 의 준*, 박 지 태*, 최 정 우*, 김 명 섭°

3-Step Sequential Application Traffic Classification System Applying Threshold to Ensemble Model

Min-Seong Lee◆, Ui-Jun Baek*, Jee-Tae Park*, Jeong-Woo Choi*, Myung-Sup Kim°

요 약

효율적인 네트워크 관리 및 운용, 서비스 품질 개선, 네트워크 보안 향상을 위하여 응용 트래픽 분류가 필수적이다. 암호화된 트래픽의 발생으로 응용 트래픽을 분류하기 위한 머신러닝 및 딥러닝을 사용한 응용 트래픽 분류에 관한 연구가 이루어지고 있다. 하지만 높은 정확도를 도출하기 위하여 데이터 전처리를 위한 새로운 방법론을 추가하거나 모델의 구조를 반복하는 등, 전체 처리 시간이 증가하는 방법들이 사용되고 있다. 본 논문에서는 간단한 3가지 모델들을 순차적으로 사용하여 전체 처리 시간을 단축하는 방법을 제안한다. 제안하는 분류 시스템에 2가지 앙상블 모델과 1가지 딥러닝 모델을 적용하여 데이터를 분류하였다. 처리 속도가 빠른 앙상블 모델에서의 적절한 임계값을 적용하여 분류가 가능한 데이터를 먼저 분류하고, 남은 데이터를 처리 속도가 느린 딥러닝 모델에 사용하여 전체 처리 속도를 개선한다. 제안한 방법을 적용한 결과, CNN 단일 모델을 사용한 결과보다 6% 높은 정확도를 도출하였으며, 전체 처리 시간은 0.21초 단축되는 결과를 도출하였다.

Key Words : Application Traffic, Traffic Classification, Machine Learning, Deep Learning, Processing Speed

ABSTRACT

Application traffic classification is essential for efficient network management and operation, service quality improvement, and network security improvement. Research is being conducted on application traffic classification using machine learning and deep learning to classify application traffic due to the generation of encrypted traffic. However, in order to derive high accuracy, methods that increase the total processing time, such as adding a new methodology for data prepossessing or repeating the structure of a model, are being used. In this paper, we propose a method to reduce the total processing time by sequentially using three simple models. Data were classified by applying two ensemble models and one deep learning model to the proposed classification system. Data that can be classified by applying an appropriate threshold in the ensemble model with high processing speed is first classified, and the remaining data is used in the deep learning model with slow processing speed to improve overall processing speed. As a result of applying the proposed method, 6% higher accuracy was derived than the result using the CNN single model, and the total processing time was shortened by 0.21 seconds.

※ 이 논문은 국토교통부 AI기반 스마트하우징 기술개발사업(20SHTD-B157018-01), 2023년도 교육부의 재원으로 한국연구재단의 지원을 받아 수행된 지자체-대학협력 기반 지역혁신 사업의 결과로 수행되었음(2021RIS-004).
◆ First Author : Korea University Department of Computer and Information Science, min0764@korea.ac.kr
° Corresponding Author : Korea University Department of Computer and Information Science, tmskim@korea.ac.kr
* Korea University of Department of Computer and Information Science, {pb1069, pjj5846, choigoya97}@korea.ac.kr
논문번호 : KNOM2023-01-11 Received July 15, 2023; Revised August 10, 2023; Accepted August 20, 2023

Ⅰ. 서 론

응용 트래픽 분류는 네트워크에 존재하는 다양한 응용 프로그램들과 프로토콜을 식별하기 위하여 중요한 역할을 한다. 네트워크 성능을 높이기 위해 분류된 트래픽에 대해 모니터링 및 최적화 등의 작업을 수행하여 효율적인 네트워크 관리 및 운용을 가능하게 한다. 기업에서는 직원들이 사용하는 애플리케이션과 서비스를 모니터링하여, 사용되는 응용 트래픽들의 분류를 통해 보안 정책을 마련할 수 있다. 서비스 측면에서는 응용 트래픽의 분류를 통해 특정 서비스의 성능을 모니터링하고, 사용량이나 패턴을 통해 서비스 품질 개선에 도움을 줄 수 있고, 서비스 제공 및 새로운 서비스를 개발할 수 있다. 네트워크 보안 측면에서 비정상적인 트래픽을 감지하고 빠른 조치를 취하기 위해 빠르고 정확한 응용 트래픽 분류가 필요하다. [1.2]

암호화된 트래픽의 발생으로 응용 트래픽을 분류하기 위한 포트 기반 분류[3], 페이로드 기반 분류[4], 그리고 통계정보를 사용한 분류[5] 방법의 사용이 어려워지고 있다. 포트 기반 분류 방법은 임의의 포트를 사용하거나 포트 번호를 위장하여 표준 등록 포트 번호를 사용하지 않는 등의 문제가 발생한다. 페이로드 기반 분류 방법으로 대표적인 데이터 패킷 검사(DPI) 방법은 데이터 패킷에서 패턴이나 키워드를 찾아낸다. 하지만 암호화되지 않은 트래픽에서만 적용할 수 있다. 통계정보 기반 분류 방법은 암호화된 트래픽에서도 사용할 수 있지만, 실용성이 떨어지는 단점이 있다. 이에 따라 머신러닝 및 딥러닝을 사용한 응용 트래픽 분류에 관한 연구가 이루어지고 있다. 대부분의 머신러닝 및 딥러닝 기반의 연구는 높은 정확도를 도출하는 연구가 진행되고 있다. 트래픽 분류의 정확도를 높이기 위하여 데이터를 전처리할 때 새로운 방법론을 추가하거나, 모델의 구조 반복, 추가적인 연산 등을 통해서 성능 향상을 이루어내고 있다. 이러한 방법들은 정확도는 높아지는 장점이 있지만, 모델의 크기가 커져 처리 속도가 느려지는 단점이 있다.

본 논문에서는 기존에 사용되던 모델들을 사용하여 모델의 구조나 추가적인 연산 없이 데이터 분류 측면에서 효율성을 고려한 방법을 제안한다. 3가지 모델을 순차적으로 사용하여 데이터를 분류하는 방법으로, 먼저 사용되는 2가지 모델은 처리 속도가 빠른 앙상블 모델을 사용하고 마지막에 사용되는 모델은 처리 속도가 느리지만, 정확도가 높은 딥러닝 모델을 사용한다. 분류가 쉬운 데이터들은 머신러닝 단계에서 임계값을 적용하여 분류가 완료된다. 분류가 완료되고 남은 데이터들은 딥러닝 단계에서 분류를 진행한다. 3가지 모델을 사용하여 전체적인 처리 속도를 개선한다.

1장의 서론에 이어 2장에서는 관련 연구에 대하여 설명한다. 3장에서는 처리 속도를 개선하기 위해 제안하는 방법에 대하여 설명한다. 4장에서는 실험 결과를 통해 제안하는 방법론의 결론을 도출한다. 마지막으로 5장에서는 결론 및 향후 연구에 대하여 설명한다.

Ⅱ. 관련 연구

본 장에서는 응용 트래픽 분류 분야에서 사용되는 인공지능 기반의 알고리즘들에 대하여 설명한다. 더불어 인공지능 기반의 알고리즘을 사용하면서 트래픽 분류속도 개선을 위한 논문을 소개한다.

2.1 앙상블 기법

랜덤 포레스트(Random Forest)는 머신러닝 분야에서 앙상블 분류 기법 배깅의 대표적인 모델로서 많은 분야에서 사용되고 있다. 랜덤 포레스트는 여러 의사 결정 트리를 병렬로 연결하고 최종 결과에 대한 다수결이나 평균을 사용하는 방법이다. 과적합의 문제를 최소화하고 예측 정확도를 높일 수 있는 것이 장점이다[6].

그래디언트 부스팅 분류(Gradient Boosting Machine) 모델[7]은 앙상블 모델에서 부스팅의 대표적인 모델로서 사용되고 있다. 이전 학습의 결과에서 나온 오차를 보완하는 방식을 통해 순차적으로 트리를 생성하는 방법이다. 차례대로 트리를 생성하기 때문에 트리를 병렬로 연결하여 처리하는 랜덤 포레스트보다 속도는 느리지만 높은 정확도를 도출하는 모델이다.

2.2 딥러닝 기법

딥러닝은 머신러닝에서 발전되어 컴퓨터 비전 분야에서 활발하게 연구가 진행되고 있다. 컴퓨터 비전 분야에서의 연구된 내용으로 다양한 분야에서 딥러닝이 적용되고 있으며. 응용 트래픽 분류 분야에서도 적용되고 있다. 응용 트래픽 분류 분야에서는 암호화된 트래픽을 분류하기 위하여 CNN (Convolution Neural Network)을 적용하여 트래픽을 분류하는 분류 방법을 제안하였다[8]. 응용 트래

픽 데이터를 학습하여 딥러닝 모델에 적용하고, 트래픽의 특징을 추출 단계에서 SAE(Stacked Auto encoder)를 사용하여 특징을 추출하고 CNN을 사용하여 응용 트래픽 분류를 연구한 논문이 있다[9].

2.3 분류 속도 개선

응용 트래픽 분류에서 분류 정확도를 높이기 위한 연구들이 대부분이다. 트래픽 분류속도와 관련된 논문들은 트래픽 분류의 병렬 처리나 실시간으로 트래픽을 분류하는 논문들이 있다. 2014년 응용 트래픽의 실시간 처리를 FPGA(Field Programmable Gate Array) 하드웨어 환경에서 SVM을 사용한 트래픽 고속 처리 연구가 있다[10]. 또한, 2018년 트래픽 분류를 위해 GPU 가속 SVM(Support Vector Machine)을 제안한 연구가 있다. 이 연구 또한 병렬 트래픽 분류를 적용하여 학습 속도를 높이고 분류 속도를 높인 연구이다[11].

Ⅲ. 본 론

본 장에서는 제안하는 방법의 전체 구조와 제안하는 방법에서 사용되는 용어 정의, 그리고 임계값을 결정하는 방법에 대하여 설명한다. 전체 구조에서 학습 모델 생성 과정과 테스트 과정에 관해서 설명한다. 제안하는 방법에서 사용되는 용어들을 정리하여 모델 구조의 이해를 도운다. 마지막으로 2가지 앙상블 모델에서 임계값을 결정하는 방법에 대하여 설명한다.

3.1 전체 모델 구조

본 절에서는 제안하는 방법의 전체 구조에 대하여 설명한다. 전체 구조는 그림1과 같다. 제안하는 방법에서 사용되는 3가지 모델은 2개의 앙상블 모델과 1개의 딥러닝 알고리즘을 사용한다. 학습 과정에서 학습 데이터로 3가지 모델에 대한 학습이 모두 이루어진다. 3가지 모델에서 학습이 완료된 후 각 모델에 대한 학습 모델이 생성된다. 학습된 모델이 완성되면서 검증 결과를 알 수 있으며, 검증 결과를 기반으로 각 모델에서 데이터를 분류하기 위한 임계값과 신뢰도를 결정해야 한다.

3가지 모델이 학습 과정을 거쳐 학습 모델이 완성되고, 검증 결과를 통해 첫 번째, 두 번째 모델에서 임계값이 결정된다. 정해진 임계값을 기반으로 테스트 데이터를 통해 3가지 모델에서 데이터를 순차적으로 분류한다. 첫 번째 모델에서 임계값을 적용하여 테스트 데이터에서 정답이 되는 데이터를 먼저 분류하고 남은 데이터는 두 번째 모델로 넘겨진다. 두 번째 모델에서는 적용된 임계값을 만족하면서 정답이 되는 데이터를 분류하게 되고 남은 데이터는 마지막으로 세 번째 모델에서 분류가 진행된다.

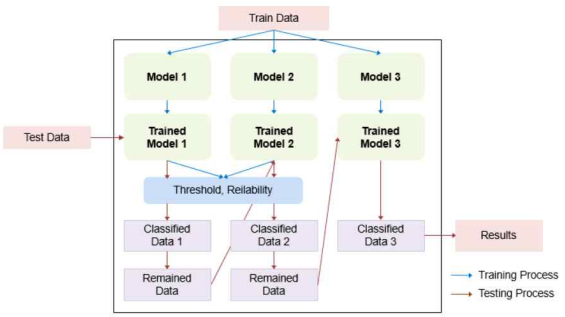

그림 1. 전체 모델 구조
Fig. 1. Overall Model Architecture

3.2 용어 정의

본 절에서는 제안하는 방법에서 사용되는 용어를 정의한다. 데이터 세트에 대한 학습이 완료되면 검증 과정에서 데이터에 대한 클래스별 분류 결과가 값으로 나타나게 된다. 클래스별 분류된 값들에 대하여 임계값이 사용되는데, 첫 번째 모델에서 임계값을 만족하지 못하는 데이터들은 첫 번째 모델에서 분류하지 못하는 것으로 판단하고 두 번째 학습 모델에서 분류된다. 마찬가지로 세 번째 모델에서의 임계값을 결정한 뒤 남은 데이터들은 세 번째 모델에서 최종적으로 분류된다. 제안하는 방법에서 사용되는 용어는 다음과 같다.

- D : 전체 데이터 세트의 개수
- M : 사용되는 학습 모델
- Th : 학습 모델에서 각 클래스를 분류하는 분류 결과에 대한 임계값
- $P_{M(Th)}$: 분류 결과에서 임계값을 적용하였을 때, 임계값을 만족하는 데이터의 개수
- $TP_{M(Th)}$: 임계값을 만족하는 데이터 중, 정답인 데이터의 개수
- $R_{M(Th)}$: 학습 모델에서 임계값에 대한 신뢰도

$$R_{M(Th)} = \frac{P_{M(Th)}}{TP_{M(Th)}} \quad (1)$$

- $CDR_{M(Th)}$: 학습 모델에서 임계값에 따른 데이터 세트의 분류 비율

$$CDR_{M(Th)} = \frac{P_{M(Th)}}{D} \quad (2)$$

3.3 임계값의 결정

본 절에서는 2가지 모델에서의 임계값을 결정하는 방법에 대해서 설명한다. 학습 모델이 완성되면 첫 번째, 두 번째에서 사용되는 앙상블 모델의 검증 결과를 기반으로 임계값과 신뢰도를 결정해야 한다. 첫 번째 모델에서 결정되는 임계값은 임계값을 변경하면서 나타나는 신뢰도의 기울기를 통해 결정할 수 있다. 기울기의 값이 가장 큰 지점의 임계값으로 첫 번째 모델의 임계값을 결정한다.

$$Max(Th_{M1}) = \frac{R_{Th} - R_{Th-0.1}}{0.1} \qquad (3)$$

두 번째 모델에서의 임계값을 결정하는 방법은 두 번째 모델에서의 신뢰도 값과 첫 번째 모델을 사용하여 데이터를 분류하고 남은 데이터로 두 번째 모델을 통해 분류하였을 때의 전체 데이터 분류 비율을 곱하여 가장 큰 값이 나오는 지점의 임계값을 사용한다. 단일 모델로서의 신뢰도 값이 중요하고, 속도 처리 개선을 위해 2가지 모델을 사용했을 때 전체 데이터를 분류하는 비율이 중요하기 때문에 두 가지 지표를 사용하여 두 번째 모델의 임계값을 결정한다. 두 가지 지표를 곱해 가장 큰 값이 나오는 지점의 임계값을 두 번째 모델의 임계값으로 결정한다.

$$Max(Th_{M2}) = R_{M2} * CDR_{M1(Th)+M2} \qquad (4)$$

Ⅳ. 실험 결과

본 장에서는 처리 속도를 개선하기 위해 제안한 방법을 실험한 결과에 대하여 설명한다. 실험에서는 3가지 모델에 데이터를 순차적으로 학습 모델에 사용하여 전체 처리 속도를 개선하는 것을 목표로 한다. 첫 번째 모델로는 랜덤 포레스트(Random Forest)를 사용하고 두 번째 모델로는 그래디언트 부스팅 분류(Gradient Boosting Machine)를 사용한다. 마지막으로 세 번째 모델로는 간단한 1D CNN 모델을 사용한다.

4.1 데이터 세트

본 절에서는 실험에 사용된 데이터 세트에 대하여 설명한다. 실험에 사용된 데이터 세트는 ISCXVPN2016으로 암호화된 응용 트래픽으로 이루어진 데이터 세트다. 해당 데이터 세트는 VPN, NoN-VPN을 분류하는 이진 분류 방법과 6가지 카테고리에 대한 분류가 가능한 공개 데이터 세트다. 본 논문에서는 6가지 카테고리의 분류를 진행한다. 데이터 세트에 포함되는 6개 카테고리에 대한 정보는 표1이며, 사용되는 플로우 정보는 표2와 같다. 데이터 세트에서 하나의 플로우 내 5개 패킷의 페이로드가 사용되며, 길이는 784바이트로 통일한다. 5개 패킷의 페이로드 길이가 784바이트보다 적으면 0으로 패딩하여 길이를 맞춰주고, 784바이트보다 많으면 784바이트에 맞춰 잘라내서 사용한다.

표 1. ISCXVPN2016 데이터 세트 정보
Table 1. ISCXVPN2016 Dataset Information

Traffic	Content
CHAT	ICQ, AIM, Skype, Facebook and Hangouts
Email	SMPTS, POP3S, and IMAPS
File_Transfer	Skype, FTPS and SFTP using Filezilla and an external service
P2P	uTorrent and Transmission (Bittorrent)
STREAMNG	Vimeo and Youtube
VOIP	Facebook, Skype and Hangouts voice calls(1h duration)

표 2. 학습 및 테스트 데이터 세트 정보
Table 2. Training and Test Dataset Information

	Train	Test	Total
CHAT	1,931	483	2,414
Email	1,426	356	1,782
File_Transfer	5,962	1,490	7,452
P2P	290	73	363
Streaming	635	159	794
VoIP	13,304	3,327	16,631
Total	23,548	5,888	29,436

4.2 데이터 학습 및 학습 모델 생성

본 절에서는 속도 비교를 위해 3가지 모델을 단일 모델로서 학습하였을 때 나오는 결과에 대하여 설명한다. 2가지 앙상블 모델과 1가지 딥러닝 모델을 학습하여 나온 결과와 학습된 모델에 테스트 데이터를 사용하여 속도 비교를 진행하였을 때 결과는 표3과 같다.

표 3. 단일 모델에서의 실험 결과
Table 3. Experimental results in a single model

Train Model	Test Accuracy	Test Time
Random Forest	80.11%	0.09sec
GBM	86.58%	0.28sec
1D CNN	91.75%	2.37sec

4.3 임계값 결정

본 절에서는 앞서 사용된 2가지 모델의 임계값을 결정하는 과정에 대하여 설명한다. 학습 데이터 세트로 학습 모델을 생성하면서 각 데이터에 대한 클래스를 분류하는 분류 결과가 나온다. 학습 모델의 분류 결과에서 임계값을 결정하기 위하여 0.2~0.9의 임계값을 설정하고, 각각의 신뢰도를 비교하여 최적의 임계값을 찾아낸다.

4.3.1 랜덤 포레스트의 임계값 결정

랜덤 포레스트는 첫 번째로 사용되는 모델이며, 3가지 모델 중 분류속도가 가장 빠른 모델이다. 학습된 모델에 테스트 데이터를 사용하고, 임계값을 주어 신뢰도를 확인한다. 랜덤 포레스트로 학습된 모델에서 검증 결과를 통해 임계값을 결정한다. 검증 결과에 대한 임계값을 적용한 결과와 기울기 값은 표4와 같다.

표 4. 랜덤 포레스트에서의 임계값 결정
Table4. Determination of Threshold in Random Forest

Th	P_{RF}	TP_{RF}	R_{RF}	CDR_{RF}	(3)
0.2	5888	4717	80.11%	80.11%	-
0.3	5839	4687	80.27%	79.60%	0.016
0.4	5728	4601	80.32%	78.14%	0.005
0.5	5507	4479	81.33%	76.06%	0.101
0.6	5116	4233	82.74%	71.89%	0.141
0.7	3160	3137	**99.27%**	53.27%	**1.653**
0.8	2694	2694	100%	45.75%	0.073
0.9	270	270	100%	4.58%	0

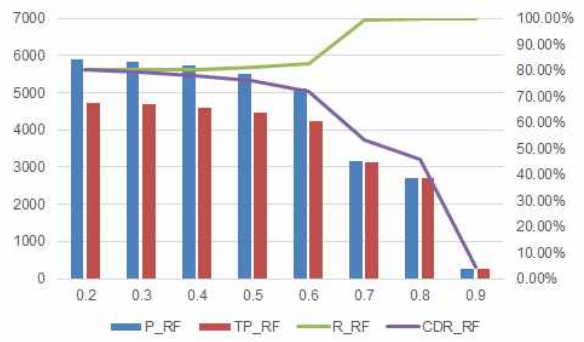

그림 2. Random Forest Result According to Threshold
Fig. 2. 임계값에 따른 랜덤 포레스트 결과

첫 번째 모델에서의 임계값을 결정할 때, 신뢰도의 기울기의 의 값이 가장 큰 지점의 임계값으로 결정한다. 랜덤 포레스트에 임계값을 주어 신뢰도의 기울기를 살펴봤을 때, 임계값이 0.7이 되는 지점의 기울기의 값이 가장 컸다. 따라서 랜덤 포레스트에서는 임계값을 0.7로 설정한다. 임계값을 설정하였을 때, 테스트 데이터 5888개 중에서 3137개의 데이터가 분류되며, 신뢰도는 99.27%로 임계값 이상으로 분류된 데이터의 많은 부분이 먼저 분류될 수 있다는 것을 확인하였다. 이는 전체 데이터의 53.27%에 해당하는 데이터를 첫 번째 분류 모델에서 분류할 수 있다는 것을 보여준다.

4.3.2 그래디언트 부스팅 분류의 임계값 결정

그래디언트 부스팅 분류는 두 번째 모델에서 사용되는 모델이며, 앞에 사용된 랜덤 포레스트보다 분류속도가 느리지만, 더 높은 정확도를 보여주고 있다. 단일 모델로 사용하였을 때, 랜덤 포레스트 모델보다 6% 정도 정확도가 더 높으며, 테스트 시간은 0.2초 느린 결과를 나타냈다. 2번째 모델에서 임계값을 결정하는 방법은 단일 모델에서의 신뢰도와 두 가지 모델을 모두 사용하였을 때 분류된 비율을 기반으로 결정한다. 그래디언트 부스팅 분류 모델을 사용한 결과에서의 신뢰도를 통해 단일 모델로서 확실하게 분류할 수 있는 수치를 확인할 수 있다. 또한, 두 가지 모델을 사용하였을 때 분류 비율을 통해 최대한 많은 비율의 데이터를 처리할 수 있도록 해야 한다. 두 가지 수치를 사용하여 적절한 임계값을 설정하고 정답이 되는 분류 데이터를 분류해야 한다. 단일 모델로서 그래디언트 부스팅 분류 모델에 임계값을 적용한 결과는 표5와 같다.

첫 번째 모델인 랜덤 포레스트에서 0.7의 임계값을 설정하고 3137개의 데이터를 먼저 분류하였다. 랜덤 포레스트에서 분류하고 남은 2751개의 데이터를 그래디언트 부스팅 분류 모델에 넣었을 때의 결과는 표6과 같다.

표 5. 임계값에 따른 그래디언트 부스팅 분류 모델 결과
Table5. Gradient Boosting Machine Result According to Threshold

Th	P_{GB}	TP_{GB}	R_{GB}	CDR_{GB}
0.2	5888	5098	86.58%	86.58%
0.3	5888	5098	86.58%	86.58%
0.4	5844	5083	86.97%	86.32%
0.5	5131	4748	92.53%	80.63%
0.6	3237	3158	97.55%	53.63%
0.7	2418	2396	99.09%	40.69%
0.8	2168	2168	100%	36.82%
0.9	1121	1121	100%	19.03%

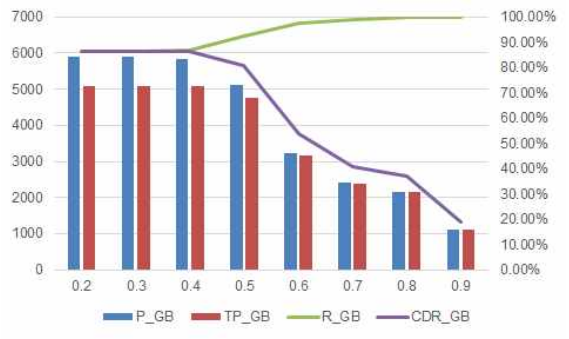

그림 3. Gradient Boosting Machine Result According to Threshold

Fig. 3. 임계값에 따른 그래디언트 부스팅 분류 모델 결과

표 6. 랜덤 포레스트 임계값 설정 후 그래디언트 부스팅 분류 모델 결과

Table6. Result of Gradient Boosting Machine after settings Random Forest Threshold

GB Th	$P_{RF(0.7)+GB}$	$TP_{RF(0.7)+GB}$	$R_{RF(0.7)+GB}$	$CDR_{RF(0.7)+GB}$
0.2	2751	1966	71.46%	86.66%
0.3	2751	1966	71.46%	86.66%
0.4	2707	1951	72.07%	86.41%
0.5	2002	1621	80.96%	80.80%
0.6	732	653	89.20%	64.36%
0.7	291	269	92.43%	57.84%
0.8	141	141	5.12%	55.67%
0.9	0	0	0%	53.27%

임계값을 설정하기 위하여 단일 모델로 사용하였을 때의 신뢰도 값과 2가지 모델을 사용하였을 때의 분류 비율을 곱하여 임계값을 확인한다. 두 번째 모델의 단일 모델 검증 결과의 신뢰도와 전체 데이터 분류 비율을 곱한 결과는 표7과 같다. 실험 결과 그래디언트 부스팅 분류 모델에서는 임계값을 0.4로 주는 것이 적절하다.

표 7. 그래디언트 부스팅 분류 모델에서의 임계값 결정

Table7. Determination of Threshold in Gradient Boosting Machine

GB Th	R_{GB}	$CDR_{RF(0.7)+GB}$	R_{GB} * $CDR_{RF(0.7)+GB}$
0.2	86.58%	86.66%	75.03%
0.3	86.58%	86.66%	75.04%
0.4	86.97%	86.41%	**75.15%**
0.5	92.53%	80.80%	62.78%
0.6	97.55%	64.36%	62.78%
0.7	99.09%	57.84%	57.31%
0.8	100%	55.67%	55.67%
0.9	100%	53.27%	53.27%

4.4 제안한 방법을 사용한 전체 결과

앞선 실험을 통해 랜덤포레스트에서는 0.7의 임계값을 적용하고 그래디언트 부스팅 분류 모델에서는 0.4의 임계값을 적용하는 것으로 결과를 내었다. 본 절에서는 두 번째 모델까지 임계값을 적용하여 데이터를 분류하고 남은 데이터를 마지막 1D CNN 모델까지 사용한 결과에 대해서 설명한다. 실험 결과를 통해 3가지 모델을 순차적으로 사용하였을 때 적용된 임계값이 적절한지 확인한다. 실험 결과는 표8과 같다.

표 8. 임계값 적용 후 CNN 모델에서의 결과

Table8. Result in CNN model after applying Threshold

GB Th	Test Data	TP_{CNN}	Test Time	Total Test Time	Total ACC
0.2	785	653	2.00sec	2.15sec	97.75%
0.3	785	653	2.00sec	2.15sec	97.75%
0.4	800	666	2.01sec	**2.16sec**	**97.72%**
0.5	1130	926	2.04sec	2.19sec	96.53%
0.6	2098	1722	2.11sec	2.26sec	93.61%
0.7	2482	2052	2.16sec	2.31sec	92.69%
0.8	2610	2163	2.17sec	2.32sec	92.40%
0.9	2751	2303	2.17sec	2.32sec	92.39%

CNN 모델까지 사용하였을 때, 그래디언트 부스팅 모델의 임계값이 높아질수록 CNN에서 처리하는 데이터의 수가 많아지면서 처리 속도가 늘어나게 된다. 앞서 임계값을 적용하는 방법을 통해 결정된 랜덤 포레스트에서의 임계값 0.7, 그래디언트 부스팅 분류 모델에서의 임계값 0.4의 값을 적용하게 되면 전체 테스트 시간이 2.16초로 CNN 단일 모델을 사용하였을 때보다 0.21초 빨라지며, 정확도는 약 6% 정도 높은 정확도를 보여준다. 두 번째 모델에서 임계값이 낮아질수록 전체 분류 정확도는 0.1% 높아지고, 전체 속도는 0.01초 빨라지지만 큰 차이는 없다. 이를 통해 실험 환경에 따라 두 번째 모델의 임계값을 사용자가 적절하게 활용하여 적용할 수 있다.

4.5 실험 결과

본 절에서는 실험한 결과를 설명한다. 본 논문에서의 목표는 3가지 모델을 순차적으로 사용하여 전체 처리 시간을 줄이는 것이 목표이다. 또한, 3가지 모델을 사용하면서 먼저 사용되는 2가지 앙상블 모델에 대한 임계값을 설정하는 방법에 관해 설명하고 실험 결과를 통해 적절한 방법인지 검증하였다. 표9

는 실험 결과에 대한 요약을 나타내며, 표10은 전체 데이터를 3가지 모델에서 분류한 비율에 관한 결과이다.

실험 결과 2가지 앙상블 모델만 비교하였을 때, 전체 정확도는 0.08% 높게 분류되었고, 전체 데이터 처리 시간은 0.04초 빨라졌다. 또한, 딥러닝 모델까지 적용하여 3가지 모델을 비교하였을 때, 전체 정확도는 1D CNN을 사용한 것보다 6% 높은 정확도를 보여주고 있고, 처리 시간은 0.21초 빨라졌다. 1D CNN이 91%의 정확도로 데이터를 처리한다고 해도 분류하지 못하는 데이터들이 있다. 이러한 데이터들을 앞에서 사용되는 두 가지 앙상블 모델에서 분류할 수 있기 때문에 전체 정확도가 높아지는 결과를 얻어낼 수 있다.

표 9 제안한 방법의 실험 결과
Table 9. Experimental Results of the proposed method

Train Model	Test Accuracy	Test Time
Random Forest	80.11%	0.09sec
GBM	86.58%	0.28sec
1D CNN	91.75%	2.37sec
RF+GBM	86.66%	0.24sec
RF+GBM+1D CNN	**97.72%**	**2.16sec**

표 10 제안한 방법의 데이터 분류 비율
Table 10. Data classification rate of the proposed method

Train Model	Input Data	Classified Data Rate
Random Forest	5888	53.27%
GBM	2651	33.13%
1D CNN	800	11.32%

V. 결 론

본 논문에서는 3가지 모델을 사용하여 데이터를 분류하는 방법을 통해 응용 트래픽 분류에서의 전체 처리 속도를 개선하는 방법과 데이터를 분류하기 위해 모델에서 임계값을 설정하는 방법에 대하여 제안하였다. 먼저 사용되는 2가지 모델에는 앙상블 모델을 사용하고 임계값을 적용하여 분류할 수 있는 데이터를 분류하였다. 많은 데이터를 처리할 수 있으면서 정확도를 챙길 수 있는 임계값을 설정하는 방법을 제안하였다. 첫 번째 모델에서는 신뢰도의 변화폭이 커지는 지점의 임계값을 설정하였고, 두 번째 모델에서는 단일 모델에서의 신뢰도 값과 두 가지 모델을 사용하였을 때의 전체 데이터 분류 비율을 곱하여 가장 큰 값이 나오는 지점의 임계값을 사용하였다. 두 가지 앙상블 모델에서의 적절한 임계값을 사용하여 데이터를 분류하고, 앙상블 모델에서 처리할 수 없는 어려운 데이터는 딥러닝 모델인 CNN 모델을 사용하여 전체적인 처리 속도를 개선하였다. 임계값을 적용하여 최대한 많은 데이터를 처리하면서 기존 모델보다 빠른 속도와 높은 정확도를 도출할 수 있었다.

향후 연구로는 딥러닝 모델을 사용하여 처리 시간을 단축하는 방법에 관한 연구를 진행하고, 실제 네트워크 환경에서 발생하는 대용량 트래픽 데이터를 적용하여 전체 처리 시간에 관한 비교 연구를 진행한다.

References

[1] M.-S. Kim, Y. J. Won, and J. W.-K. Hong, “Application-level traffic monitoring and an analysis on IP networks,” ETRI J., vol. 27, pp. 22-42, 2005.

[2] B. Park, Y. Won, J. Chung, M. S. Kim, and J. W. K. Hong, “Fine-grained traffic classification based on functional separation,” Int. J. Network Management, vol. 23, pp. 350-381, Sept. 2013.

[3] IANA port number list. Available: http://www.iana.org/assignments/service-names-prt-numbers/service-names-port-numbers.xml

[4] T. Choi, C. Kim, S. Yoon, J. Park, B. Lee, H. Kim, et al., “Content-aware internet application traffic measurement and analysis,” IEEE/IFIP NOMS 2004, pp. 511-524, 2004.

[5] N. F. Huang, G. Y. Jai, H. C. Chao, Y. J. Tzang, and H. Y. Chang, “Application traffic classification at the early stage by characterizing application rounds,” Inf. Sci., vol. 232, pp. 130-142, May 2013.

[6] Breiman L. Random forests. Mach Learn. 2001;45(1):5 – 32.

[7] Friedman, Jerome H. “Greedy Function Approximation: A Gradient Boosting Machine.” The Annals of Statistics 29, no. 5 (2001): 1189 – 1232. http://www.jstor.org/stable/2699986.

[8] Z. Zou, J. Ge, H. Zheng, Y. Wu, C. Han and Z. Yao, ″Encrypted Traffic Classification with a Convolutional Long Short-Term Memory Neural Network,″ 2018 IEEE 20th International Conference on High Performance Computing and Communications; IEEE 16th International

Conference on Smart City; IEEE 4th International Conference on Data Science and Systems (HPCC/SmartCity/DSS), 2018, pp. 329-334, doi: 10.1109/HPCC/SmartCity/DSS.2018.00074.

[9] Lotfollahi, M., Jafari Siavoshani, M., Shirali Hossein Zade, R. et al. Deep packet: a novel approach for encrypted traffic classification using deep learning. Soft Comput 24, 1999–2012 (2020). https://doi.org/10.1007/s00500-019-04030-2

[10] T. Groléat, S. Vaton, and M. Arzel, "High-speed flow-based classificationon FPGA," Int. J. Netw. Manag., vol. 24, no. 4, pp. 253–271, Jul. 2014.

[11] Guanglu Sun, Xuhang Li, Xiangyu Hou, and Fei Lang. GPU-Accelerated Support Vector Machines for Traffic Classification [J]. Int J Performability Eng, 2018, 14(5): 1088-1098.

이 민 성 (Min-Seong Lee)

2019년 고려대학교 컴퓨터정보학과 학사

<관심분야> 네트워크 관리 및 보안, 트래픽 모니터링 및 분석

백 의 준 (Ui-Jun Baek)

2018 고려대학교 컴퓨터정보학과 학사

2018년 - 현재 고려대학교 컴퓨터정보학과 석사과정

<관심분야> 네트워크 관리 및 보안, 트래픽 모니터링 및 분석

박 지 태 (Jee-Tae Park)

2017 고려대학교 컴퓨터정보학과 학사

2017년 - 현재 고려대학교 컴퓨터정보학과 석사과정

<관심분야> 네트워크 관리 및 보안, 트래픽 모니터링 및 분석

최 정 우 (Jung-Woo Choi)

2022년 : 고려대학교 컴퓨터정보학과 학사

2022년~현재 : 고려대학교 컴퓨터정보학과 석사과정

<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 전자계산 학과 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