# 애플리케이션 트래픽 분류를 위한 머신러닝 알고리즘 성능 분석*

김성윤, 김명섭
고려대학교 컴퓨터정보학과
e-mail:{adayslife, tmskim}@korea.ac.kr

# Performance Analysis of Machine Learning Algorithms for Application Traffic Classification

Sung-Yun Kim, Myung-Sup Kim
Dept. of Computer and Information Science, Korea University

요 약

기존에 트래픽 분류 방법으로 payload 분석이나 well-known port를 이용한 방법을 많이 사용했다. 하지만 동적으로 변하는 애플리케이션이 늘어남에 따라 기존 방법으로 애플리케이션 트래픽 분류가 어렵다. 이러한 문제의 대안으로 Machine Learning(ML) 알고리즘을 이용한 애플리케이션 트래픽 분류 방법이 연구되고 있다. 기존의 논문에서는 일정 시간동안 수집한 data set을 사용하기 때문에 적게 발생한 애플리케이션은 제대로 분류하지 못하여도 전체적으로는 좋은 성능을 보일 수 있다. 본 논문에서는 이러한 문제를 해결하기 위해 각 애플리케이션마다 동일한 수의 data set을 수집하여 애플리케이션 트래픽을 분류하는 방법을 제시한다. ML 알고리즘 중 J48, REPTree, BayesNet, NaiveBayes, Multilayer Perceptron 알고리즘을 이용하여 애플리케이션 트래픽 분류의 정확도를 비교한다.

## 1. 서론

다양한 애플리케이션이 등장하고 이로인한 트래픽이 증가됨에 따라 네트워크를 효율적으로 관리하는 것이 중요해졌다. 특히 적합한 QoS(Quality of Service) 및 안전한 네트워크 환경을 제공하기 위해서 정확한 애플리케이션 트래픽 분류는 필요하다. 예전에는 IANA에서 지정한 port를 기반으로 한 well-known port를 이용한 방법과 애플리케이션의 payload 분석을 통해 특정 signature를 추출하는 방법을 사용하였다. 하지만 최근에는 동적 port를 할당하여 사용하거나 payload를 암호화하여 사용하는 애플리케이션이 많아지고 있어서 기존의 방법을 통한 애플리케이션 트래픽 분류의 정확도가 떨어지고 있다. 이러한 문제점을 해결하기 위해서 Machine Learning(ML) 알고리즘을 적용하여 트래픽을 분류하는 것이 대안으로 제시되고 있다.

기존 ML 알고리즘을 이용한 연구에서는 동일한 시간대의 data set을 가지고 트래픽을 분류하였다. 하지만 이 방법은 동일한 시간대에 애플리케이션 트래픽 양이 다르므로 비중을 많이 차지하는 애플리케이션을 잘 분류하고 비중이 적은 애플리케이션을 잘 분류하지 못해도 전체적으로 좋은 성능이 된다. 전체적으로는 좋은 성능을 보이지만 각 애플리케이션을 정확하게 분류하였다고는 할 수가 없다. 이를 보완하기 위해서 ML알고리즘을 적용하는 data set의 수를 동일하게 하여 애플리케이션 트래픽 분류 성능을 평가해야 한다.

기존 대부분의 논문에서는 feature를 source IP, destination IP, source port number, destination port number이 네 가지로 선택하여 분류를 수행하고 있다. 그러나 Williams는 feature의 개수가 많을수록 분류의 Overall accuracy가 높아짐을 보여준다.[4] 본 논문에서는 성능 향상을 위해서 더 추가적인 feature 들을 선택하여 애플리케이션 트래픽 분류 성능을 평가한다.

ML 알고리즘을 이용한 애플리케이션 트래픽 분류에서 고려할 평가 방법은 cross validaton과 split validaton 이 있다. cross validaton은 동일한 data set 안에서 trainnig set과 testing set을 구성한다. 반면에 split validaton 은 독립적인 trainnig set과 testing set으로 이루어진다. 본 논문에서는 실제 네트워크에 적합한 split validaton 방법 [2, 3] 을 적용한다.

본 논문의 구성은 다음과 같다. 2장에서는 본 연구에서 사용된 data set과 Feature Set 대해 설명한다. 3장은 실험한 내용과 결과에 대해 설명한다. 마지막으로 4장에서는 결론과 향후 연구로 본 논문을 맺는다.

## 2. Data set과 Feature Set

이 장에서는 본 논문에서 선택한 애플리케이션과 정의한 Feature Set에 대해 기술한다.

* 이 논문은 2007년 정부(교육인적자원부)의 재원으로 한국학술진흥재단의 지원을 받아 수행된 연구임(KRF-2007-331-D00387)

### 2.1 Data set

본 논문에서는 고려대학교 세종캠퍼스에서 많이 사용된다고 판단되는 애플리케이션과 외국에서 많이 사용되고 있는 애플리케이션들로 선택하였다. 최근 p2p 트래픽의 비중이 높아짐에 따라 p2p 트래픽을 중심으로 선택하였다. <표 1>은 선택된 애플리케이션을 보여준다.

<표 1> 선택한 애플리케이션

| 종 류 | | 애플리케이션 | Flow 개수 | |
|---|---|---|---|---|
| | | | Train | Test |
| P2P | Filesharing | Bittorrent | 500 | 500 |
| | | Edonkey | 500 | 500 |
| | | Fileguri | 500 | 500 |
| | | Limewire | 500 | 500 |
| | | Morpheus | 500 | 500 |
| | Messenger | MSN | 500 | 500 |
| | | Nateon | 500 | 500 |
| | | Skype | 500 | 500 |
| | P2P-TV | AfreecaPlayer | 500 | 500 |
| | | Joost | 500 | 500 |
| | | PPLive | 500 | 500 |
| Non - P2P | | Gom TV | 500 | 500 |
| | | Grealradio | 500 | 500 |
| | | Iexplore | 500 | 500 |
| | | POP3 | 500 | 500 |
| | | Coreftp | 500 | 500 |

우선 크게 p2p와 non-p2p 2개의 데이터로 나누고 p2p는 Filesharing, Messenger와 p2p-TV로 3개의 데이터로 나누었다. Filesharing는 국내에서 많이 사용하고 있는 Fileguri, 국외에서 많이 사용되는 Bittorrent, Limewire, Morpheus, 국내외에서 많이 사용되는 Edonkey를 선택하였다. Messenger는 국내에서 많이 사용되는 Nateon과 국외에서 많이 사용되는 Skype, 국내외에서 많이 사용되는 MSN을 선택하였다. p2p-TV는 국내에서 많이 보는 아프리카와 국외에서 많이 보는 PPLive, Joost를 선택하였다. non-p2p의 경우는 Webbrowser인 Iexplore와 라디오와 동영상을 방송하는 Gom TV, Grealradio와 메일 서비스인 POP3와 파일을 주고 받을 수 있는 Coreftp로 선택하였다. 각 애플리케이션은 500개의 training set과 testing set 으로 구성하였다.

### 2.2 Feature Set

본 논문에서는 애플리케이션 트래픽을 flow 단위로 구분하고 <표 2> 에서 정의된 바와 같이 각 flow로부터 feature set을 추출하였다. 본 논문에서는 flow의 정의를 패킷의 5-tuple 정보를 공유하는 양방향 패킷들의 집합으로 정의한다. 즉 5-tuple 정보가 동일한 단방향 패킷들과 이의 역방향 패킷들의 합이다. TCP의 경우 SYN 패킷에 의한 연결시작에서부터 FIN 패킷이 발생하는 연결의 끝 사이에 발생한 모든 양방향 패킷들이 하나의 flow가 되고, UDP의 경우는 최소 패킷 발생 시점에서 마지막 패킷이 발생한 사이의 모든 양방향 패킷들이 하나의 flow가 된다.

<표 2> 선정한 Feature

| |
|---|
| Source IP Address/ Port |
| Destination IP Address/ Port |
| IP protocol |
| In / Out Packets sent in Duration |
| In / Out Octets sent in Duration |
| In / Out # of SYN packet in flow |
| In / Out # of ACK packet in flow |
| In / Out # of RST packet in flow |
| In / Out # of FIN packet in flow |
| In / Out Packets (minimum, maximum, average, standard deviation) |
| In / Out window (minimum, maximum, average, standard deviation) |
| In / Out jitter (minimum, maximum, average, standard deviation) |

선정된 Feature는 IP adress, port번호, IP protocol의 종류, flow안의 패킷의 수와 size의 총합, TCP flow에 각 flag(syn, ack, rst, fin)가 있는 패킷 수, 각 패킷의 size와 window 와 시간차의 minimum, maximum, average, standard deviation 이다.

본 논문은 5가지의 Feature set을 이용하여 애플리케이션을 트래픽 분류의 정확도를 비교 분석해 보았다.

(1) all : 모든 attribute
(2) whitout ip : source/ destination ip 제외
(3) whitout port : source/ destination port 제외
(4) whitout ip&port : source/ destination ip & port 제외
(5) whitout src ip&port : source ip & port 제외

## 3. 실험결과

본 논문에서는 5가지의 data set을 가지고 p2p와 non-p2p 분류, p2p에서의 Filesharing와 Messenger와 p2p-TV 분류 그리고 16가지의 애플리케이션 분류를 하는 실험을 하였다. 본 논문에서는 Weka tool [1]을 이용하여 ML 알고리즘 중에 기존 논문에서 많이 사용하였던 J48, REPTree, BayesNet, NaiveBayes, MLP(Multilayer Perceptron) 알고리즘를 이용하여 애플리케이션 트래픽 분류를 하였다

### 3.1 평가방법

분류의 정확도를 평가하기 위해서 Precision, Recall Overall accuracy 3가지 평가 기준을 사용하였다. Precision과 Recal은 각 애플리케이션 마다의 분류의 정확도를 나타내는 것이고 Overall accuracy는 전체 애플리케이션의 정확도를 나타내는 것이다. Recall은 'A 그룹으로 분류된 원소 중 실제 A 그룹에 속한 개수 / 분류 전 실제 데이터 set에서 A 그룹의 원소 개수'이고 precision은 'A 그룹으로 분류된 원소 중 실제 A 그룹에 속하는 원소 개수 / A 그룹으로 분류된 전체 원소 개수'이다. 수식은 아래와 같은 식으로 나타낼 수 있다.

<표 3> Confusion Matrix

| | Actual Positive | Actual Negative |
|---|---|---|
| Predicted Positive | TP(True Positive) | FP(False Positive) |
| Predicted Negative | FN(False Negative) | TN(True Negative) |

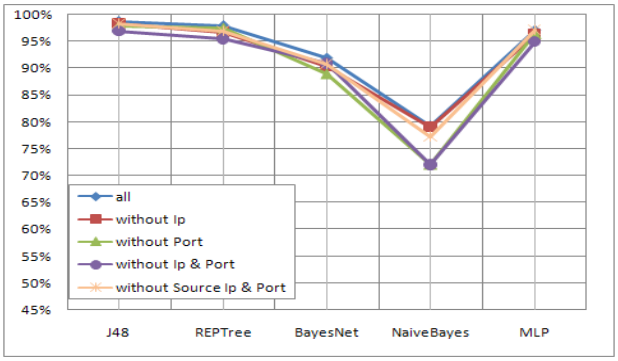
$$Recall = \frac{TP}{TP+FN} \qquad Precision = \frac{TP}{TP+FP}$$

$$Overall\ Accuracy = \frac{summation\ TP\ of\ each\ Application}{total\ element}$$

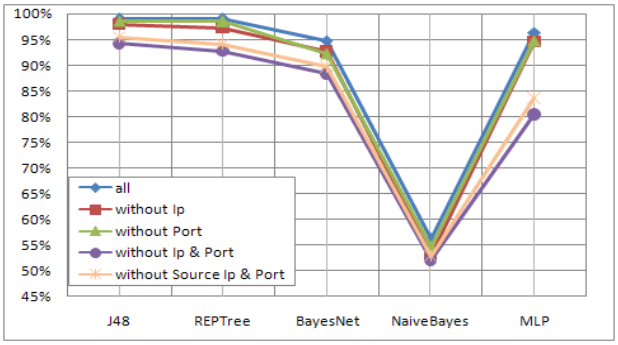
### 3.2 애플리케이션 트래픽 분류

(그림 1, 2, 3) 은 5가지의 data set을 J48, REPTree, BayesNet, NaiveBayes, MLP 알고리즘을 이용하여 (p2p, non-p2p), (Filesharing, Messenger, P2P-TV), (16가지 애플리케이션)으로 총 75회 실험 하였다. 모든 실험에서 J48, REPTree 알고리즘은 95%이상의 Overall accuracy 결과가 나왔다. MLP의 경우 p2p, non-p2p 분류에서는 좋은 성능이 나왔지만 16가지 애플리케이션 분류에서는 성능이 떨어졌다.

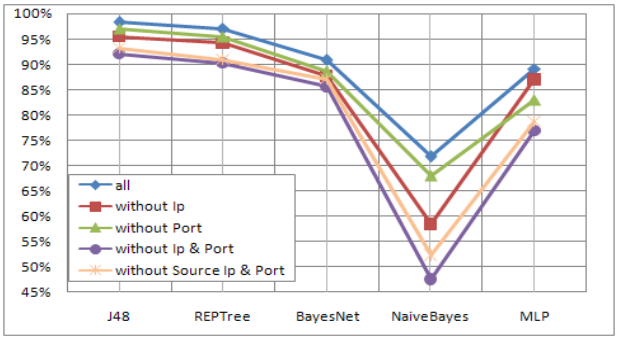
Feature set은 전체적으로 all 일 때 가장 좋은 성능을 보이며 whitout ip&port 일 때 가장 나쁜 성능을 보였다.




(그림 1) P2P , Non-P2P Overall Accuracy




(그림 2) Filesharing, Messenger, P2P-TV Overall Accuracy




(그림 3) 각 애플리케이션 Overall Accuracy

<표 4> 는 각 애플리케이션 분류 중 가장 좋은 성능을 보인 J48 알고리즘 이용하여 all Feature set을 분류한 결과이다. Overall accuracy는 98.29% 로 좋은 성능을 보였다.

<표 4> J48 알고리즘을 이용한 분류 결과 (all)

| 애플리케이션 | Precision | Recall | 애플리케이션 | Precision | Recall |
|---|---|---|---|---|---|
| Bittorrent | 0.986 | 0.972 | Afreeca | 0.978 | 0.97 |
| Edonkey | 0.988 | 1 | Joost | 0.996 | 1 |
| Fileguri | 1 | 1 | PPLive | 1 | 1 |
| Limewire | 0.998 | 0.988 | Gom TV | 0.992 | 0.992 |
| Morpheus | 0.961 | 0.99 | Grealradio | 0.998 | 0.996 |
| MSN | 0.97 | 0.956 | Iexplore | 0.942 | 0.97 |
| Nateon | 0.958 | 0.914 | POP3 | 0.98 | 0.99 |
| Skype | 0.99 | 0.992 | Coreftp | 0.988 | 0.996 |

<표 4> 를 보면 Nateon의 Recall 값이 95% 밑으로 실험 결과 중에서 낮은 결과값을 보였다. 하지만 낮은 Recall값도 90%이상의 결과가 보임으로써 전체적인 분류 성능이 좋았다. Nateon 데이터중 분류가 잘못된 것은 거의 Iexplore트래픽으로 분류 되었다. PPLive, Fileguri는 Precision과 Recall의 결과값이 1로 자신의 트래픽만 모두 완벽하게 분류 하였다.

## 4. 결론 및 향후 과제

본 논문은 같은 수의 data set으로 다양한 ML 알고리즘과 5가지 Feature set으로 애플리케이션을 분류하였다. 일정한 수의 data set을 적용함으로써 좀더 정확한 애플리케이션 트래픽 분류를 하였다.

앞으로 가장 좋은 성능이 보인 J48 알고리즘을 이용하여 더 다양한 p2p 트래픽과 게임 트래픽 등 실제 많이 사용되고 있는 애플리케이션 트래픽 분류가 필요하겠다. 또한 많은 feature를 선택하였는데 feature의 개수가 많을수록 계산 복잡도가 높아지는 단점이 있다. 그러므로 overall accuracy을 높인다고 판단하기 힘든 feature들을 제외한 나머지 feature로 재정의가 필요하겠다.

## 참고문헌

[1] Machine Learning Lab in The University of Waikato, "weka"[Online]Available: http://www.cs.waikato.ac.nz/ml.

[2] 정광본, 최미정, 김명섭, 원영준, 홍원기, "Split Validation 방법에서의 트래픽 분류를 위한 최적의 ML 알고리즘과 Feature Set", 통신학회 추계종합학술발표회, 서울대학교, 서울, Nov. 17, 2007, pp. 177.

[3] 정광본, 최미정, 김명섭, 원영준, 홍원기, "ML 알고리즘을 적용한 인터넷 애플리케이션 트래픽 분류," The Committee on Korean Network Operations and Management (KNOM), Vol. 10, No. 2, Dec. 2007, pp. 39-52.

[4] N. Williams, S. Zander, G. Armitage, "Evaluating Machine Learning Methods for Online Game Traffic Identification," CAIA Technical Report 060410C, April 2006.