실제 네트워크 모니터링 환경에서의 ML 알고리즘을 이용한 트래픽 분류

준회원 정 광 본*, 정회원 최 미 정*, 김 명 섭**, 원 영 준*, 홍 원 기*

Traffic Classification Using Machine Learning Algorithms in Practical Network Monitoring Environments

Kwang Bon Jung* *Associate Member*,
Mi Jung Choi*, Myung Sup Kim**, Young J. Won*, James W. Hong* *Regular Members*

요 약

Traffic classification의 방법은 동적으로 변하는 application의 변화에 대처하기 위하여 페이로드나 port를 기반으로 하는 것에서 ML 알고리즘을 기반으로 하는 것으로 변하여 가고 있다. 그러나 현재의 ML 알고리즘을 이용한 traffic classification 연구는 offline 환경에 맞추어 진행되고 있다. 특히, 현재의 기존 연구들은 testing 방법으로 cross validation을 이용하여 traffic classification을 수행하고 있으며, traffic flow를 기반으로 classification 결과를 제시하고 있다. 본 논문에서는 testing방법으로 cross validation과 split validation을 이용했을 때, traffic classification의 정확도 결과를 비교한다. 또한 바이트를 기반으로 한 classification의 결과와 flow를 기반으로 한 classification의 결과를 비교해 본다. 본 논문에서는 J48, REPTree, RBFNetwork, Multilayer perceptron, BayesNet, NaiveBayes와 같은 ML 알고리즘과 다양한 feature set을 이용하여 트래픽을 분류한다. 그리고 split validation을 이용한 traffic classification에 적합한 최적의 ML 알고리즘과 feature set을 제시한다.

Key Words : 트래픽 분류 (Traffic classification), ML 알고리즘 (algorithm), 애플리케이션 분류

ABSTRACT

The methodology of classifying traffics is changing from payload based or port based to machine learning based in order to overcome the dynamic changes of application's characteristics. However, current state of traffic classification using machine learning (ML) algorithms is ongoing under the offline environment. Specifically, most of the current works provide results of traffic classification using cross validation as a test method. Also, they show classification results based on traffic flows. However, these traffic classification results are not useful for practical environments of the network traffic monitoring. This paper compares the classification results using cross validation with those of using split validation as the test method. Also, this paper compares the classification results based on flow to those based on bytes. We classify network traffics by using various feature sets and machine learning algorithms such as J48, REPTree, RBFNetwork, Multilayer perceptron, BayesNet, and NaiveBayes. In this paper, we find the best feature sets and the best ML algorithm for classifying traffics using the split validation.

※ 본 연구는 두뇌한국 21 연구결과로 수행되었음
* 포항공과대학교 컴퓨터공학과 ({jkbon, mjchoi, yjwon, jwkhong}@postech.ac.kr)
** 고려대학교 컴퓨터정보학과 (tmskim@korea.ac.kr)
논문번호 : KICS2007-10-458, 접수일자 : 2007년 10월 8일, 최종논문접수일자 : 2008년 7월 21일

Ⅰ. 서 론

인터넷이 발전하면서 멀티미디어 서비스와 대용량 데이터 전송 등 다양한 서비스를 제공하기 위한 애플리케이션이 등장하고, 웜 바이러스와 같이 네트워크에 문제를 일으키는 트래픽이 등장하고 있다. 네트워크 관리자는 다양한 애플리케이션에서 발생하는 네트워크 트래픽을 원활히 지원할 수 있도록 네트워크의 용량을 계획 (planning)하기 위해서, 또한 웜 바이러스 같이 네트워크와 서비스에 문제를 야기하는 트래픽을 구분하기 위해서 각 애플리케이션별 네트워크 트래픽을 분류 (traffic classification) 할 필요가 있다.

현재 traffic classification은 주로 포트 번호를 바탕으로 이루어지고 있으며, 페이로드를 통한 분석도 행해지고 있다. 최근의 애플리케이션은 동적인 포트 번호를 할당하여 패킷을 발생하며, 심지어 IANA에 고정적으로 할당되어 있는 포트 번호까지도 할당해서 패킷을 발생하는 애플리케이션이 늘어나고 있다. 뿐만 아니라 페이로드를 통한 분석에 있어서도 애플리케이션에서 생성한 패킷의 페이로드가 암호화되어 전송되는 경우 페이로드 분석을 통한 트래픽 분류를 어렵게 하고 있다. 이러한 추세는 페이로드나 포트 번호를 이용한 traffic classification의 정확도를 떨어뜨린다. 이러한 문제의 해결책으로 포트 번호 또는 페이로드에 의존하지 않고 트래픽 특징에서 얻은 트래픽의 통계적 feature들을 이용한 traffic classification이 제시되었고, 이러한 경향에 따라 Machine Learning (ML) 알고리즘을 이용한 traffic classification이 분류의 대안으로 제시되고 있다 [4,6,7,16].

기존 ML 알고리즘을 이용한 traffic classification 연구에서는 source IP, destination IP, source port, destination port, protocol의 5가지 정보가 동일하며 1분 동안(처음 발생한 packet과 최후에 발생한 packet의 시간 차이)에 발생한 모든 packet을 양방향의 트래픽 흐름으로 정의하는 flow를 기반으로 네트워크 트래픽의 feature set을 구성하여 트래픽 분류를 수행하고 있다. 또한, 기존 연구에서 제시된 traffic classification 방법론의 성능을 평가함에 있어 대부분의 연구는 동일한 시간대에 수집된 하나의 데이터 set 안에서 training set과 testing set을 구성하는 cross validation을 채택하고 있지만, 이는 실제 네트워크를 운영하는 운영자의 입장에서 현실적으로 적용가능성이 적다. 이를 보완하기 위해서 ML 알고리즘을 적용하여 training을 시키는 training set과 성능 평가에 사용되는 testing set을 분리하는 split validation 기법이 적용되어야 한다. 또한 classification의 정확도 표현에 있어서도 flow 기반으로 수집한 데이터에 대해서 성능을 나타내는 것이 아니라 각 패킷의 바이트 양을 기준으로 하는 바이트 기반으로 표현하는 것이 실제 네트워크 모니터링과 관리에 필요하다. 본 논문에서는 기존 연구에서 진행되어 온 cross validation과 flow 기반의 traffic 분류 방법에 대한 문제점을 찾아보고 split validation과 바이트 기반 traffic classification의 성능 평가 필요성을 제시한다. 이러한 필요성을 기반으로 split validation과 바이트 기반 정확도 관점에서 traffic classification을 위한 최적의 ML 알고리즘과 feature set을 실험을 통하여 제시하고자 한다.

본 논문의 구성은 다음과 같다. Ⅱ장에서는 traffic classification에 적용되고 있는 여러 가지 ML 알고리즘들을 살펴보고, 본 논문과 관련 있는 기존 연구에 대해서 정리한다. Ⅲ장에서는 본 논문에서 다루고자 하는 문제에 대해서 정리하고, Ⅳ장에서는 본 연구에서 사용한 traffic data set에 대해 설명하도록 한다. Ⅴ장에서는 본 연구에서 실행한 실험에 대한 내용과 실험 결과에 대해서 설명하고 실험 결과의 의미를 요약한다. 마지막으로 Ⅵ장에서 결론과 향후 연구로 본 논문을 맺는다.

Ⅱ. 관련 연구

이 장에서는 traffic classification에 관련된 ML 기법에 대해서 간략히 살펴보고, ML 알고리즘을 적용하여 traffic classification을 수행한 기존 연구에 대해서 정리한다.

2.1 Machine Learning 알고리즘

Traffic classification에 사용되는 ML 알고리즘은 크게 Supervised ML, Unsupervised ML, Neural Network 기반의 ML로 나누어 볼 수 있고, 선택된 ML 알고리즘의 분류의 성능을 높이기 위하여 Feature Reduction이 추가되어 사용된다.

Supervised ML 알고리즘은 데이터를 이미 알려진 그룹으로 분류하는 방법이다. 즉, 분류해야 하는 데이터의 종류를 미리 알려준 다음, 새로운 데이터를 이미 알려진 그룹 중 하나로 분류하는 것이다. 이러한 ML 알고리즘으로는 Bayesian Network, Bayesian Network Tree, Naïve Bayes, Naïve Bayes Tree,

Decision Tree (J48, C4.5) 등이 있다.

Unsupervised ML 알고리즘은 데이터들의 유사성을 기반으로 그룹핑을 함으로써 분류하는 알고리즘을 말한다. Supervised ML 알고리즘은 알려진 그룹으로 분류가 되지만, Unsupervised ML 알고리즘은 알려진 그룹 외에 전혀 다른 그룹으로도 분류가 된다. 여기서 전혀 다른 그룹은 machine에 의해 나누어진 것이므로 사람이 그 그룹의 정체성을 파악하기 힘들다. 즉, 새로운 데이터에 대해서 새로운 그룹을 만듦으로써 기존에 그룹핑 되지 않은 데이터에 대해서도 분류가 가능해진다. 단, Unsupervised ML 알고리즘의 경우 새로운 그룹에 대해서는 같은 특성이 있다고 말할 수 있을 뿐 새로운 그룹이 명확히 어떤 것인지는 말 할 수 없다. 이러한 ML 알고리즘으로는 Expectation Maximization (EM), AutoClass, Nearest Neighbor (NN), K Means, DBSCAN 등이 있다.

Neural Network은 생물학에서의 neural network를 수학적이며 계산학적인 모델로 표현한 것이다. 이것들은 인공적인 뉴런들의 상호 연결된 것들로 이루어져 있다. 대부분의 경우에 artificial neural network는 학습 기간 동안에 네트워크를 통해서 흘러 다니는 외부 또는 내부 정보들을 기반으로 네트워크의 구조를 바꿔가며 적응하는 시스템이라 할 수 있다. Neural network는 input node, output node, hidden node의 관계로 이루어져 있으며, 결론적으로 데이터의 패턴을 찾아내는 기능을 한다고 볼 수 있다. Neural network에서 자주 사용하는 알고리즘으로는 Radial Base Function (RBF) Network와 Multilayer Perceptron (MLP) 등이 있다[14].

Feature Reduction은 feature set을 구성할 때에 feature들을 최적화하여 선택하는 방법론이다. Feature들을 많이 넣으면 복잡도가 증가하고, feature들이 적으면 정확도가 감소하게 되는 trade off 관계를 갖는 문제에 대한 해결 방법이다. 이러한 기법으로는 Fast Correlation Based Filter (FCBF) [15]와 Genetic Algorithm (GA) [9] 등이 있다.

2.2 기존 연구 (State of Art)

이 장에서는 ML 알고리즘을 이용한 기존의 traffic

표 1. ML 알고리즘을 이용한 Traffic Classification 연구
Table 1. Research Work of Traffic Classification using ML Algorithms

		ML 알고리즘	Feature set	Description
1	Erman et al. [2]	K-Means, DBSCAN, AutoClass	total number of packets, packet size, payload size, number of transferred bytes, inter-packet arrival time	애플리케이션별 트래픽 분류, 각 ML 알고리즘을 통한 traffic classification 결과 비교 분석
	Zander et al. [16]	Expectation Maximization (EM)	IP address, port number, inter-packet arrival time, packet length, flow size and duration	애플리케이션별 traffic 분류, 다양한 데이터 set을 통해 제안한 분류의 효율성을 측정하여 optimal feature set을 추출
2	Nguyen et al. [5]	Naive Bayes	IP address, port number, protocol, inter-packet arrival time, inter-packet length variation, IP packet length	25개의 패킷을 이용한 sub-flow들의 적절한 조합을 통한 training을 통해 애플리케이션 별 classification 수행
	Park et al. [8]	Naive Bayes, NBKE REPTree	duration of flow, number of packets, initial advertised-window bytes, number of data packets, number of packets with 'PUSH' option, packet size, advertised-window bytes, inter-packet arrival time, number and size of total burst packets	애플리케이션을 QoS provision에 따라 구별하고 QoS level별로 traffic classification이 이루어지는 scheme을 소개
	Andrew et al. [7]	Naive Bayes, NBKE, FCBF	flow duration, TCP port, inter-packet arrival time, payload size, effective bandwidth based upon entropy, Fourier transform of packet	Naïve Bayes estimator를 이용한 애플리케이션 별 traffic classification을 보임
3	Erman et al. [4]	EM, Naive Bayes classifier	total number of packets, packet size, flow duration, inter-packet arrival time	Unsupervised ML 알고리즘을 이용한 애플리케이션 별 traffic classification의 가능성을 보임
	Williams et al. [6]	Naive Bayes, C4.5, Bayesian Network, Naive Bayes Tree	protocol, flow duration, flow volume in bytes and packets, packet length, inter-packet arrival time	Feature reduction이 가지는 효과를 보여줌, 여러 가지 알고리즘의 비교를 통하여 C4.5가 가장 성능이 좋음을 보임
4	Park et al. [9]	GA, J48, NBKE, REPTree		Feature reduction 기법으로 GA를 사용
	Moore et al. [11]			Feature set 249개를 제안함.

classification 연구에 대해서 살펴본다. ML 알고리즘을 이용한 traffic classification 연구는 약 1990년대 말부터 이루어졌으며, 현재에도 많은 연구가 진행 중이다. 이 연구들은 크게 Unsupervised ML 알고리즘을 이용한 경우, Supervised ML 알고리즘을 이용한 경우, Unsupervised와 Supervised ML 알고리즘을 모두 이용한 경우, 그리고 Feature Reduction 방법을 이용한 경우, 이렇게 4가지 경우로 나누어 볼 수 있다.

표 1은 기존 연구들을 4가지 경우 (1: Unsupervised, 2: Supervised, 3: Both, 4: Feature Reduction)로 나누어 분류하고 각 연구 별로 사용한 ML 알고리즘과 분류에 적용된 feature set, ML 알고리즘을 적용하여 분류한 결과가 무엇인지 나누어 정리한 것이다. 표 1을 보면 Protocol, 바이트 양, connection duration, packet size statistics와 inter-packet arrival statistics를 기본적인 feature set으로 선택하여 필요에 의해 새로운 feature를 추가하여 트래픽을 분류하고 있다. ML 알고리즘을 이용한 traffic classification에 관련해서 진행된 많은 연구들이 offline 환경에서 training과 testing 트래픽의 구분 없이 트래픽을 분류하는 것에 초점을 두고 있지만, 본 논문에서는 training과 testing 트래픽이 다른 환경인 실제 네트워크 모니터링 환경에서 네트워크 트래픽을 분류하는 것에 초점을 두고 있다.

또한 기존 연구에서는 traffic classification을 하기 위한 데이터를 backbone에서 얻고 있는데 이 데이터가 어떤 애플리케이션에서 발생된 것인지 정확히 알 수 있는 방법이 없어 포트 번호에 따른 트래픽 분류 결과를 제시하고 있다. 즉, 기존 연구에서는 애플리케이션별로 분류한 결과가 정확하다는 보장이 없다. 본 논문에서는 데이터를 데스크톱에서 ethereal [10]을 이용하여 수집함으로써 각 애플리케이션에서 생성된 트래픽별로 데이터를 수집하고 분류를 수행함으로써 그 결과에 있어서 정확성이 보장된다. 또한 기존의 traffic classification 연구에서 많이 사용되지 않은 Neural Network 기반의 ML 알고리즘들로 적용해 봄으로써 기존 연구에서 많이 사용된 ML 알고리즘과의 분류 정확도 결과도 비교할 수 있다.

Ⅲ. 문제 정의

이 장에서는 2.2장의 ML 알고리즘을 적용한 traffic classification의 기존 연구 방법을 실제 네트워크상에서의 트래픽 분류에 적용할 때 고려해야 할 점을 정리한다. 기존 연구를 살펴보면 두 가지 문제점을 생각해 볼 수 있다. 첫째, 대부분의 논문 [5, 6, 7, 8, 9]은 flow 기반으로 트래픽 분류의 정확도를 측정한 반면에 [7]은 바이트를 기반으로 분류의 정확도를 측정하고 있다. 대부분의 기존 연구에서는 feature set을 구성하는 정보가 flow 기반에서 얻을 수 있는 것들로 구성되어 있고, 이러한 feature set을 통해서 training과 testing을 수행하고 있다. 따라서 분류의 정확도 측정도 flow 기반으로 이루어지고 있다.

그러나 flow 기반으로 정확도를 측정하는 경우에, 여러 가지 문제가 발생할 수 있다. 첫째, 각 flow가 전체 트래픽 양에서 차지하는 비중은 flow마다 다르다. 즉, flow가 가지고 있는 packet의 수가 다르고 전체 바이트 양도 다를 수 있다. 예를 들면, 어떤 flow는 10,000 바이트로 이루어져 있고, 다른 flow는 1,000 바이트 크기라면, 각각이 1개의 flow지만 바이트 기준으로 보면 이 2개의 flow가 제대로 분류 되었는지 정확도를 측정함에 있어서 그 비중이 분명히 다르다는 것이다. 이러한 문제는 대용량 바이트를 많이 발생시키는 P2P 애플리케이션 또는 Web disk, FTP 애플리케이션에서 많이 찾아볼 수 있다. 이러한 문제를 Class Imbalance Problem이라고 한다 [12]. 여기서 class는 하나의 애플리케이션으로 볼 수 있는데, 대용량의 바이트를 발생시키는 애플리케이션과 MSN messenger, Google talk와 같은 작은 양의 바이트를 발생시키는 chatting 애플리케이션은 모든 애플리케이션에서 발생하는 바이트에서 차지하는 비중 측면에서 볼 때, 분명 그 양에서 상대적인 차이를 보이게 된다.

또한 flow를 기반으로 traffic classification을 수행하는 것은 네트워크 모니터링에서 중요한 부분을 차지하는 traffic shaping이나 usage billing policy에 적용하기 위한 실제적인 트래픽 데이터양에 관련한 정확한 데이터를 제공하지 못하기 때문에 [12] 이러한 문제를 해결하기 위해서라도 분류 결과의 정확도를 바이트 기반으로 고려하는 것이 필요하다.

기존 연구의 또 다른 문제로는 성능평가를 위한 testing 기법과 관련된 문제이다. 기존 연구[2,4,5,6,16]를 살펴보면 training과 testing을 위한 기법으로 cross validation 기법을 선택하고 있다. cross validation 기법을 선택하는 것은 의도하지 않은 문제점을 안고 있다. 예를 들어, 포트 번호와 같은 feature에서 문제점을 찾아볼 수 있는데, 대부분의 애플리케이션은 더 이상 고정적인 포트 번호를 사용하지 않는다. 특히 애플리케이션을 사용하는 client에서 사용하는 포트 번호는 유동성의 정도가 더 심하다. Client 애플리케이션의 포트 번호는 특

정한 seed 값에서 1씩 증가하면서 할당되는 특징을 가지고 있다. 특정 시간에 특정한 애플리케이션 'A'만을 사용한 데스크톱에서 얻은 데이터를 살펴보니 client에서 할당한 포트 번호가 1000번부터 3000번까지 1씩 증가하는 것을 볼 수 있었다. 이 데이터를 cross validation 기법에 대입해 보면, 그 데이터 set 안에서 training을 위한 데이터와 testing을 위한 데이터가 형성되므로, 이렇게 형성된 데이터들을 가지고 분류를 하게 되면 1~3000번에 있는 포트 번호를 가지는 데이터는 'A'라는 애플리케이션에 의해서 발생된 것이라고 분류되기 쉽고 그 외의 포트 번호를 가지는 데이터는 'A'라는 애플리케이션에 의해서 발생된 것이라고 분류되기 어렵다. 만약 split validation을 이용하여 분류를 할 때, 앞에서 말한 데이터를 training을 위한 데이터로 사용하고 포트 번호가 6000번부터 9000번까지 형성된 데이터를 testing을 위한 데이터로 사용하여 분류를 수행하게 되면 cross validation을 이용하여 분류를 한 결과만큼 분류의 정확도가 높게 나오지 않는다. 따라서 split validation의 경우는 client의 포트 번호는 적절한 feature set이 될 수 없다. 즉, cross validation을 이용한 분류를 기반으로 하는 기존의 연구 결과가 실제 네트워크 모니터링 환경에서의 네트워크 애플리케이션을 분류하는데 적용하기에 부적절한 면이 있다는 것을 알 수 있다.

Ⅳ. 데이터 트레이스 및 Feature Set 정의

이 장에서는 네트워크 트래픽을 수집한 환경과 본 논문에서 ML 알고리즘에 적용하기 위한 feature set의 종류 및 분류 결과의 정확성을 측정하기 위한 방법에 대해서 살펴본다.

4.1 트래픽 트레이스 (Traffic Trace)

본 논문에서는 2.2장에서 언급한 기존 연구들의 backbone에서 데이터를 수집하여 포트 번호를 기반으로 애플리케이션을 분류한 방법의 정확성에 대한 문제 때문에 하나의 데스크톱에서 ethereal [10]을 이용하여서 7가지의 대표 애플리케이션의 data trace를 수집하였다. 이 대표 애플리케이션은 POSTECH의 네트워크 현황을 모니터링 하고 있는 NG-MON [3]을 참조하여 사람들이 많이 사용하고 있다고 판단되는 애플리케이션들 중에 다양한 종류를 선택한 것이다. 7개의 대표 애플리케이션으로 online으로 음악방송을 제공해주는 'alsong', online으로 방송을 제공받거나 제공할 수 있는 'afreeca', Web disk인 'clubbox', ftp를 이용해서 파일을 주고받을 수 있는 'alftp', Microsoft에서 제공하는 chatting 애플리케이션인 'MSN messenger', 여러 가지 contents를 실시간으로 제공해주는 'Gom', Web browser인 'iexplore'를 선택하였다.

표 2. 각 애플리션별 Data size 및 Flow 개수
Table 2. Data size & # of the Flow of Each Application

애플리케이션	Training		Testing	
	Size (MB)	Flow (개수)	Size (MB)	Flow (개수)
MSN	454.07	377	454.07	65
Afreeca	946.97	483	946.97	223
Clubbox	904.244	4490	904.244	835
Gom	168.654	1633	168.654	756
Alftp	1574.528	501	1574.528	234
Iexplore	73.542	1386	73.542	258
Alsong	16.3975	85	16.3975	49

데스크톱에서 ethereal을 써서 각 애플리케이션 별 packet을 캡처하여 필요한 feature 데이터를 추출하였다. 각 애플리케이션마다 training과 testing을 위한 데이터가 각각 필요하기 때문에 packet 데이터도 두 번을 수집했다. 각각 training을 위한 packet은 4시간 정도 모으고, testing을 위한 packet 데이터는 1시간 30분 동안 모았다. Split validation을 위해 두 개의 데이터를 수집한 시간은 서로 다르게 하였다. Testing을 위한 데이터를 1시간 30분 동안 모은 까닭은 적어도 1시간 이상은 모아야 각 애플리케이션의 특성을 반영하는 데이터를 모을 수 있기 때문이고, training을 위한 데이터를 4시간 정도는 모아서 훈련을 시켜야 testing을 위한 1시간 30분 데이터의 애플리케이션의 데이터 특성을 모두 반영할 수 있을 것이라고 생각하기 때문이다. 표 2는 packet 데이터의 용량을 MB로 나타낸 것과 이렇게 모은 데이터들을 flow 데이터로 가공한 값을 나타낸 것이다.

4.2 Feature 정의

Traffic classification을 위해 수집하는 정보인 feature를 선정하는 기준은 데이터를 aggregation하는 기준에 따라서 달라질 수 있다. 본 논문에서는 flow를 기준으로 데이터를 aggregation하였으며 네트워크 트래픽을 분류하기 위해 가장 많이 사용되는 feature 정보들 [5]은 아래와 같다.

- IP address (source, destination)
- Port number (source, destination)
- 바이트 양

- Connection duration
- Packet size statistics (minimum, maximum, mean, standard deviation)
- Inter packet arrival time statistics (minimum, maximum, mean, standard deviation)

IP address와 포트 번호는 packet header에서 얻을 수 있는 정보이며, 데이터가 암호화 된다 하더라도 이 정보들은 라우팅 정보로 남아 있기 때문에 값을 추출할 수 있다. 바이트 양은 하나의 flow에 속하는 데이터의 양이 얼마나 되는지를 나타내는 feature이며, connection duration은 하나의 flow가 지속되는 시간을 나타내는 feature이다. Packet size에 대한 다양한 통계자료와 inter-packet arrival time에 대한 다양한 통계자료도 각 flow가 가지는 특징을 나타내는 feature라고 볼 수 있다. 대부분의 기존 연구 [2, 4, 5, 6, 16]에서는 source IP, destination IP, source port number, destination port number 이 네 가지를 feature들을 선택하여 분류를 수행하고 있다. 본 논문에서는 feature set을 형성할 때에, feature중 IP address와 포트 번호를 선택하거나 혹은 선택하지 않은 다양한 set을 형성하여서 최적의 feature set을 구하려 한다. 따라서 다음과 같이 5가지 종류의 feature set을 정의했으며, 이들 중 최적의 feature set을 찾고자 한다.

(1) all: 모든 feature가 선택된 경우
(2) without port: 모든 feature에서 source와 destination의 port number를 제외한 경우
(3) without IP: 모든 feature에서 source와 destination의 IP address를 제외한 경우
(4) without IP&port: 모든 feature에서 source와 destination의 IP address, port number를 제외한 경우
(5) without src IP&src port: 모든 feature에서 source IP address와 port number가 빠진 경우

4.3 평가

ML 알고리즘을 통한 분류가 잘 되었는가 평가하기 위해서 precision과 recall 그리고 overall accuracy 세 가지 평가 기준이 있으며, 식1~3과 같이 True Positive (TP)와 False Positive (FP), False Negative (FN) 등으로 나타낼 수 있다.

Overall accuracy : 전체 데이터를 하나로 놓고 제대로 분류가 된 양이 얼마나 되는지에 대해서 알아보기 위한 평가 기준이다. 반면에 precision과 recall은 각 애플리케이션을 기준으로 분류가 얼마나 정확하게 되었는지에 대해서 알아보기 위한 평가 기준이다.

Overall accuracy

$$= \frac{\Sigma\, TP of each application}{total element} \quad \text{(식 1)}$$

Recall과 precision을 예를 들어 설명하면 다음과 같다. 실제 A라는 그룹에 속하는 30개 데이터를 ML 알고리즘 이용해서 분류 한 결과 그 중에서 A 그룹에 속했다고 분류한 결과 값이 40개이고 이중 실제 A 그룹에 속하는 데이터가 20개였다면, 전체 A 그룹의 30개중 20개만 A 그룹에 속했다고 분석한 비율인 20/30이 recall의 값이다. 그리고 분류한 결과 A라는 그룹에 속한다고 나온 결과 40개 중에는 A 뿐 아니라 B나 C 그룹 내의 값도 포함된 것이다. 제대로 분류한 A 값은 20개 이므로 A 그룹으로 분류된 데이터 중 실제로 A에 속하는 데이터의 비율은 20/40, 즉 제대로 분류한 비율 값이 precision이다.

즉, recall과 precision을 정의하면 다음과 같다.

recall : A 그룹으로 분류된 원소 중 실제 A 그룹에 속한 개수 / 실제 데이터 set에서 A 그룹에 속하는 원소 개수

$$Recall = \frac{TP}{FP + EN} \quad \text{(식 2)}$$

precision : A 그룹으로 분류된 원소 중 실제 A 그룹에 속하는 원소 개수 / A 그룹으로 분류된 전체 원소 개수

$$\text{Precision} = \frac{TP}{TP + FP} \quad \text{(식 3)}$$

최적의 ML 알고리즘과 feature set을 찾기 위해 식 1을 이용하여 overall 정확도 값이 가장 높은 것을 선택한다. 또한 최적의 ML 알고리즘과 feature set의 match에 대해서는 각 애플리케이션의 precision과 recall도 살펴보아 분류의 정확도를 알아보고자 한다.

V. 실험 결과

이 장에서는 5가지 feature set과 다양한 ML 알고리즘 중에서 주어진 traffic trace에 대하여 overall 정확도를 가장 높이는 최적의 feature set과 ML 알고리즘을 찾고자 한다. 본 논문에서는 다음과 같은 3가지 실험을 수행하였다. Cross validation과 split validation의 2가지 testing 기법에 따른 traffic classification 결과를 비교하는 실험과, 바이트 기반으로 결과를 구한 것과, flow 기반으로 측정한 분류 결

과를 비교하는 실험이다 마지막으로, split validation에서 최적의 알고리즘과 최적의 feature set을 구하는 실험을 수행하였다. 본 논문에서는 ML tool 중 하나인 Weka [1]를 이용하여 실험하였다.

5.1 Testing 기법에 따른 트래픽 분류 분석

이 장에서는 먼저 cross validation과 split validation의 testing 기법에 따른 분류 결과를 비교하는 실험을 수행한다. 4.1장에서 설명한 방법으로 training과 testing을 수행하였다. 여기에서 사용한 ML 알고리즘은 Decision tree인 J48 알고리즘이다. 이 알고리즘을 선택한 이유는 기존 연구 [6, 8, 9]에서 cross validation 방법으로 실험했을 때, J48 알고리즘이 traffic classification에 있어서 뛰어난 성능을 보였기 때문이다. 이 실험에서 사용하는 feature set은 cross validation과 split validation의 분류 정도를 비교하기 위해 대표로 모든 feature (1: all)를 사용했을 때와 전체 feature에서 IP address를 사용하지 않았을 경우 (3: without IP) 2가지를 살펴보았다.

그림 1은 cross validation과 split validation을 testing 방법으로 사용하였을 때의 traffic classification을 한 결과의 overall accuray (식 1)를 보여주고 있다. 먼저 모든 feature로 구성된 feature set을 적용할 때, cross validation의 경우 overall 정확도가 약 95.55%로 측정된 반면에 split validation을 testing method로 사용하면 약 62.64%의 overall accuracy 값을 보인다. IP address를 제외한 feature들로 구성된 feature set을 적용할 때에도 cross validation을 testing method로 사용하면 overall accuracy가 약 95.76%값을 보였고, split validation의 경우는 overall accuracy가 약 63.56%로 측정되었다.

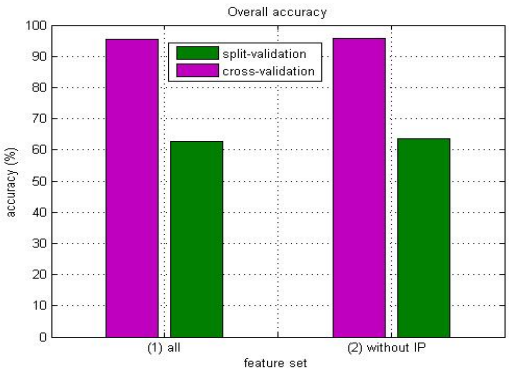

그림 1. Cross & Split Validation의 Overall 정확도
Fig. 1. Overall 정확도 of Cross & Split Validation

두 가지 feature set 모두 cross validation을 적용하면 기존 논문과 같이 traffic classification에 있어서 높은 overall accuracy를 가지고 있지만, split validation을 하게 되면 그리 높지 않은 overall accuracy를 보임을 알 수 있다. 즉, 실제 네트워크상의 traffic classification에서는 split validation이 이루어져야 함으로 기존의 cross validation의 traffic classification의 정확도 값을 그대로 받아들이기 어렵다. 또한 split validation 상에서의 최적의 ML 알고리즘과 feature set들 역시 기존의 cross validation 방법과 다를 수 있다. 따라서 본 논문에서는 실험을 통해 split validation에서의 최적의 ML 알고리즘과 feature set을 찾는 것은 의미가 있다.

5.2 바이트/Flow 기반에 따른 트래픽 분류 분석

이 장에서는 traffic classification을 바이트 기반으로 수행한 것과 flow 기반으로 수행했을 때의 overall accuracy를 비교한다. 본 논문에서 제시한 6개의 알고리즘을 모두 적용해 보았으며 사용한 feature set은 4.2에서 언급한 모든 feature들로 구성된 feature set (1: all)이다. 그리고 여기서 사용하는 알고리즘은 J48, REPTree, NaiveBayes, BayesNet, MLP, RBFNetwork이다.

그림 2는 바이트와 flow 기반으로 traffic classification을 수행했을 때, overall accuracy를 각 ML 알고리즘 별로 나타낸 것이다. 이 결과를 살펴보면, J48은 바이트를 기반으로 분류를 수행하면 overall accuracy가 약 85% 정도 이지만 flow 기반일 경우에는 약 63% 정도의 정확도를 보인다. 대부분의 알고리즘의 차이가 약 6%에서 많으면 약 20%까지 보이면서 바이트를 기반으로 했을 때에 정확도 결과가 더 높음을 알 수 있다. 그러나 RBFNetwork 알고리즘은 바이트를 기반으로 했을 때에는 약 78%이지만 flow를 기반으로 하

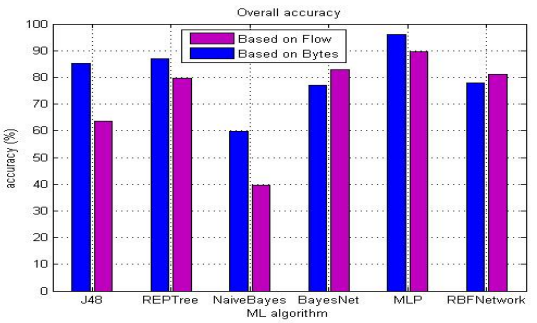

그림 2. 바이트 & Flow 기반의 Overall 정확도
Fig. 2. Overall 정확도 based on 바이트 & Flow

면 약 81%의 정확도가 나온다. 또한 BayesNet 알고리즘도 바이트를 기반으로 했을 때에는 정확도 값이 약 77%였지만 flow를 기반으로 하면 약 82%의 정확도가 나온다. 이와 같이 근소한 차이를 보이면서 flow 기반으로 분류했을 때에 overall 정확도 값이 더 높은 결과가 나오는 ML 알고리즘도 있다.

이 실험 결과 바이트 기반의 분석에서는 최적의 ML 알고리즘이 MLP, REPTree, J48, RBFNetwork, BayesNet 등의 순서였지만 flow 기반의 분석에 있어서는 최적의 정확도를 보이는 ML 알고리즘이 MLP, BayesNet, RBFNetwork, REPTree, J48의 순서를 보임을 알 수 있다. 또한 바이트 기반의 분석을 통해서 traffic classification에 있어서 전체적으로 overall 정확도가 높은 결과를 얻을 수 있다. 따라서 실제 네트워크 모니터링 환경에서 정확한 traffic classification을 수행하는 환경을 위해서는 바이트 기반의 분석이 필요하며 바이트 기반으로 분석시의 최적의 ML 알고리즘과 feature set을 구해야 한다.

5.3 Split Validation 환경에서의 최적의 ML 알고리즘과 Feature Set

이 장에서는 실험을 통해 split validation 환경에서의 최적의 ML 알고리즘과 feature set을 찾아보고자 한다.

그림 3은 6개의 알고리즘과 5개의 다른 feature set을 적용하였을 때 바이트 기반의 분류 결과로써 overall 정확도를 나타내고 있다. 그림 3의 결과를 살펴보면 바이트 기반으로 traffic classification을 수행했을 때에는 MLP 알고리즘을 이용하여 (3)번, (1)번, (5)번, (2)번 feature set을 이용하였을 경우에 각각 약 95%, 94%, 92%, 91%의 overall accuracy를 나타내었

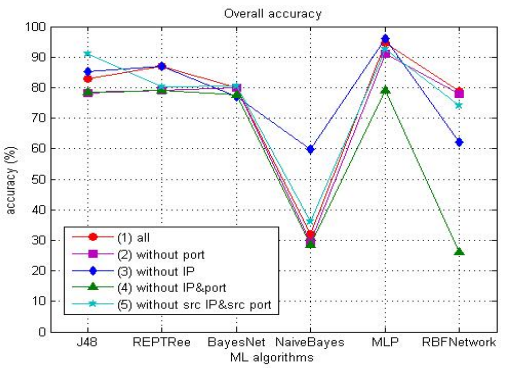

그림 3. Feature set과 ML 알고리즘에 따른 Split Validation을 적용한 바이트 기반의 Overall 정확도
Fig. 3. Overall 정확도 based on 바이트 by Different Feature Sets & ML Algorithms with Split Validation

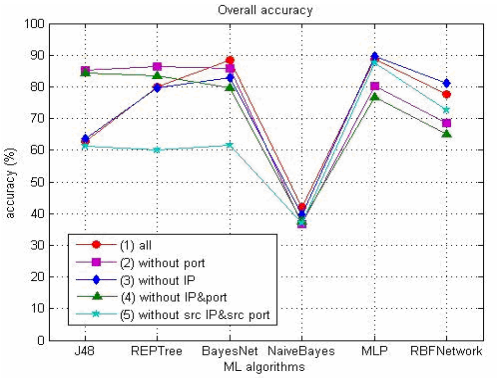

그림 4. Feature set과 ML 알고리즘에 따른 Split Validation을 적용한 Flow 기반의 Overall 정확도
Fig. 4. Overall 정확도 based on Flow by Different Feature Sets & ML Algorithms with Split Validation

다. 그리고 J48 알고리즘을 이용하고 (5)번 feature set을 이용하였을 경우에 약 91%의 overall accuracy가 나옴을 알 수 있다.

그림 4는 6개의 알고리즘과 5개의 다른 feature set을 적용하였을 때의 flow 기반의 분류 결과로써 overall 정확도를 나타내고 있다. Multilayer Perceptron (MLP) 알고리즘에 4.2장의 (3)번 feature set을 적용했을 때가 89.48%로 overall accuracy가 가장 높게 나오는 것을 볼 수 있으며, BayesNet 알고리즘에 (1)번 feature set을 적용했을 때에 88.58%로 두 번째로 높게 나오는 것을 볼 수 있다. 그 다음은 MLP 알고리즘을 이용하여 (5)번 feature set을 사용하면 높은 overall accuracy가 나옴을 볼 수 있다. J48, REPTree 알고리즘을 이용하여 (2)번이나 (4)번 feature set을 사용하였을 때에도 비교적 높은 overall accuracy가 나옴을 알 수 있다. Flow 기반이었을 때에는 BayesNet 알고리즘 같은 경우도 높은 정확도가 나왔지만 오히려 바이트를 기반으로 하였을 때에는 정확도가 높지 못하다.

결국 본 논문에서 가장 높은 overall accuracy를 보이는 ML 알고리즘과 feature set은 MLP 알고리즘과 전체 feature에서 IP address를 사용하지 않았을 경우 (3: without IP)임을 바이트 기반 (95%의 overall accuracy)에서나 flow 기반 (89%의 overall accuracy)에서나 다름이 없음을 알 수 있다. 하지만, 그 다음으로 높거나 세 번째로 높은 ML 알고리즘과 feature set은 바이트를 기반으로 했을 때와 flow를 기반으로 했을 때에 차이를 보였다.

표 3은 MLP 알고리즘과 (3)번 feature set을 이용했을 경우에 각 애플리케이션의 precision과 recall을 바

표 3. MLP 알고리즘과 Feature Set (3: without IP)을 적용하였을 때 Precision & Recall
Table 3. Precision & Recall with MLP Algorithm and Feature Set (3: without IP)

Application	Precision (%)		Recall (%)	
	byte based	Flow based	byte based	Flow based
msn	1	36	27.66	37.5
Iexplore	93	78.4	99.52	91.6
Alsong	0	0	0	0
Gom	99.93	94	74.38	95.6
Afreeca	94.36	87.6	90.73	75.6
ftp	99.96	83.9	99.99	99
Clubbox	84.01	93.7	99.91	96.3

이트 기반으로 보았을 때와 flow 기반으로 보았을 때를 나타낸 것이다. 표 3에서 보면 'alsong' 애플리케이션이 제대로 분류되지 않은 결과를 볼 수 있다. 하지만 이 결과는 MLP 알고리즘과 (3)번 feature set에 따른 결과일 뿐 다른 알고리즘이나 feature set을 적용했을 경우에는 'alsong'도 분류되는 결과를 찾아 볼 수 있다. Feature set 중 (5)번 source IP와 source port를 제외한 feature들로 구성된 feature set과 J48을 이용할 경우 'alsong' 애플리케이션의 precision은 76.2%가 나오고 recall은 38.1%로 나온다. 하지만 MLP를 이용하면 분류가 제대로 되지 않았다. 이는 data set에 의해서 'alsong' 애플리케이션의 결과가 좋지 않은 것이 아니라, MLP 알고리즘의 특징에 의해 좋지 않은 결과가 나왔음을 의미한다.

또한, MLP 알고리즘을 적용하여 (3)번 feature set으로 실험했을 때, flow 기반일 때 traffic classification의 overall accuracy는 89%였고 바이트 기반일 때는 95%였다. 즉 바이트 기반일 때에 traffic classification의 결과가 더 좋게 나타남을 알 수 있는데 그 결과가 표 3에도 드러나고 있음을 볼 수 있다. 그러나 msn이나 Gom player와 같은 몇 가지 특정 애플리케이션의 경우는 flow 기반의 precision과 recall의 수치가 더 높게 나오는 것을 알 수 있다. 따라서 전체 overall 정확도를 높이는 최적의 ML 알고리즘과 각 애플리케이션별 최적의 알고리즘에는 차이가 있음을 알 수 있다. 추후 모든 애플리케이션의 recall과 precision을 높일 수 있는 방법에 대한 연구가 더 필요하다.

5.4 Cross Validation 환경에서의 최적의 ML 알고리즘과 Feature Set

본 논문에서는 5.3장의 split validation 방법으로 얻은 결과를 기존 연구인 cross validation 방법을 통해 얻은 결과와 비교하기 위해서 5.3장에서 이용했던 알고리즘들과 feature set을 그대로 cross validation 방법을 통해 수행하였다.

그림 5는 5.3장과 같이 6개의 알고리즘과 5개의 feature set을 cross validation 방법으로 적용하여 바이트 기반의 overall accuracy를 측정한 그래프이다. 공식 1을 이용한 overall accuracy를 측정한 결과, cross validation을 이용한 결과가 그림 3의 split validation을 했을 때보다 대부분 높게 나왔다.

그림 6은 그림 5와 같은 실험을 flow 기반으로 수행한 것이다. 그림 5와 6에서 보듯이 cross validation을 수행했을 때의 최적의 알고리즘은 J48과 REPTree이다. 바이트 기반으로 측정한 값의 overall 정확도는 모든 feature set에 대하여 세 개의 ML 알고리즘인 J48, REPTree, BayesNet 모두 비슷한 overall accuracy를 보여줌을 알 수 있다. 특이한 점은 NaiveBayes 알고리즘의 경우 바이트 기반의 모든 feature 값을 가진 것 (1: all)의 경우 97% 정도의 정확도를 보인 반면

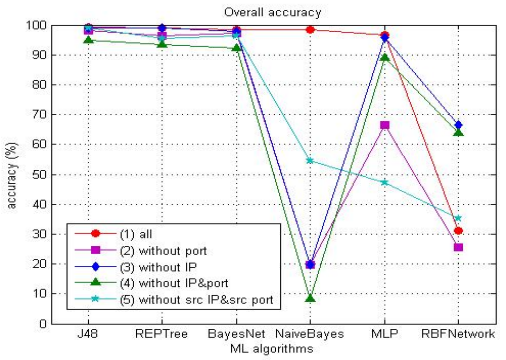

그림 5. Feature set과 ML 알고리즘에 따른 Cross Validation을 적용한 바이트기반의 Overall 정확도
Fig. 5. Overall 정확도 based on 바이트 by Different Feature Set and Algorithms with Cross Validation

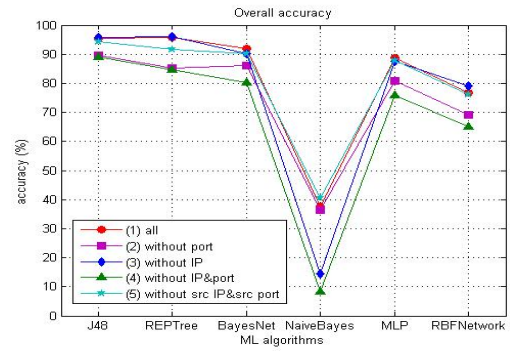

그림 6. Feature set과 ML 알고리즘에 따른 Cross Validation을 적용한 Flow 기반의 Overall 정확도
Fig. 6. Overall 정확도 based on Flow by Different Feature Set and Algorithms with Cross Validation

flow 기반의 경우 38% 정도의 정확도를 보였다.

Cross validation을 이용하여 flow를 기준으로 하였을 때에 가장 높은 overall accuracy를 갖는 알고리즘과 feature set은 REPTree와 (3)번 feature set이고 정확도는 96.12%를 보임을 알 수 있다. 바이트를 기준으로 하였을 때에 가장 높은 overall accuracy를 갖는 알고리즘과 feature set은 J48과 (1)번 feature set이고 정확도는 99.1%임을 알 수 있다. Split validation을 이용하여 flow를 기준으로 하였을 때와 바이트를 기준으로 하였을 때에, 가장 좋은 overall accuracy를 가진 알고리즘과 feature set은 MLP과 IP address를 제외한 (3: without IP) feature set이고 값은 각각 89.48%, 95.97%이다. 즉 cross validation과 split validation에서의 최적의 ML 알고리즘과 feature set이 다름을 알 수 있다.

Split validation에서 MLP 알고리즘을 이용하면, cross validation을 이용해서 얻을 수 있는 overall accuracy 값에 미치지는 않지만, 실제 모니터링 환경상에서 적용 가능한 split validation 방법에서 적어도 MLP 알고리즘을 이용하면 cross validation 수준의 충분히 좋은 결과를 얻을 수 있다는 것이다. 그러나 표 3에서 보았듯이 'alsong'과 같은 애플리케이션의 경우 MLP 알고리즘으로 사용한 경우 제대로 분류가 되지 않음을 볼 수 있다. 즉 split validation에서 overall accuracy 측면에서는 최적의 알고리즘이지만 특정 애플리케이션의 경우 제대로 분류를 하지 못하기 때문에 추후에 특정 애플리케이션을 분류하기 위해서는 각 애플리케이션별 최적의 ML 알고리즘과 feature set을 찾는 연구도 고려되어야 한다. 또한 실제 모니터링 환경에서는 flow가 아닌 바이트 기반의 정보를 수집하여 트래픽 분류의 overall 정확도 값을 높일 수 있다.

Ⅵ. 결론 및 향후 연구

본 논문에서는 기존의 연구 논문에서 찾아 볼 수 있는 cross validation과 flow 기반의 traffic classification의 문제점을 살펴보고 그것을 해결하기 위해 split validation과 바이트 기반의 traffic classification의 필요성을 제시하였다. 실제 네트워크상의 네트워크 트래픽의 분류를 위해서는 바이트 기반의 데이터를 split validation 방법으로 분석해야 한다.

Cross validation과 split validation 방법론의 각 경우에 적합한 최적의 ML 알고리즘과 feature set을 실험을 통하여 제시하였다. 바이트 기반으로 값을 측정하여 classification의 overall accuracy 결과 값과 flow 기반의 결과 값이 어떻게 다른지를 비교하였다. Split validation에서의 최적의 알고리즘과 feature set은 Neural Network 계열의 MLP (Multilayer perceptron)가 최적의 overall accuracy을 갖는 성능을 보였으며, 최적의 feature set은 IP를 제외한 feature set (3: without IP)이 가장 좋은 성능을 보였다. 또한 바이트 기반의 값이 flow 기반의 데이터에 비해 traffic classification에 있어서 더 좋은 overall accuracy를 보임을 알 수 있다.

추후에 이루어져야 할 연구는, 데이터를 수집하는 것이 하나의 데스크톱이 아닌 다수의 데스크톱에서 얻은 데이터를 가지고 classification을 해봐야 할 것이다. 그리고 좀 더 나은 feature set을 선정하기 위해서, packet의 header 정보를 좀 더 가공하여 최적의 feature를 찾는 연구가 이루어져야 한다. 또한 분류하고자 하는 트래픽의 종류나 목적에 따라 최적의 ML 알고리즘과 feature를 찾는 연구도 진행되어야 한다.

참 고 문 헌

〔1〕 Machine Learning Lab in The University of Waikato, "Weka", 〔Online〕 Available: http://www.cs.waikato.ac.nz/ml.

〔2〕 Jeffrey Erman, Martin Arlitt, Anirban Mahanti, "Traffic Classification Using Clustering Algorithms", SIGCOMM'06 Workshops, Pisa, Italy, Sep. 2006, pp.281-286.

〔3〕 Se Hee Han, Myung Sup Kim, Hong Taek Ju and James W. Hong, "The Architecture of NG MON: A Passive Network Monitoring System", IFIP/IEEE International Workshop on Distributed Systems: Operations and Management, LNCS 2506, Montreal, Canada, Oct. 2002, pp.16-27.

〔4〕 Jeffrey Erman, Anirban Mahanti, Martin Arlitt, "Internet Traffic Identification using Machine Learning", IEEE Global Telecommunications Conference, California, USA、Nov.~Dec. 2006, pp.1-6.

〔5〕 Thuy T. T. Nguyen, Grenville Armitage, "Training on multiple sub flows to optimize the use of Machine Learning classifiers in real world IP networks", IEEE Conference on Local

Computer Networks, Tampa, Florida, USA, Nov. 2006, pp. 369~376.
〔6〕 N. Williams, S. Zander, G. Armitage, “A Preliminary Performance Comparison of Five Machine Learning Algorithms for Practical IP Traffic Flow Classification”, SIGCOMM Computer Communication Review, Oct. 2006, pp.7-15.
〔7〕 Andrew W. Moore, Denis Zuev, “Internet Traffic Classification Using Bayesian Analysis Techniques”, SIGMETRICS’05, Banff, Alberta, Canada, Jun. 2005, pp.50-60.
〔8〕 Junghun Park, Hsiao Rong Tyan, and C. C. Jay Kuo, “Inetnet Traffic Classification For Scalable QoS Provision”, IEEE International Conference on Multimedia and Expo, Jul. 2006, pp.1221~1224.
〔9〕 Junghun Park, Hsiao Rong Tyan, C. C. Jay Kuo, “GA Based Internet Traffic Classification Technique for QoS Provisioning”, International Conference on Intelligent Information Hiding and Multimedia, Pasadena, California, USA, Dec. 2006, pp.251-254.
〔10〕 Etheral, http://www.ethereal.com.
〔11〕 Andrew Moore, Denis Zuev and Michael Crogan, “Discriminators for use in flow based classification”, Technical Report, Intel Research Cambridge, 2005.
〔12〕 Jeffrey Erman, Anirban Mahanti, Martin Arlitt, “Byte Me: A Case for byte accuracy in Traffic Classification”, MineNet’07, J San Diego, California, USA, Jun. 2007, pp.35-37.
〔13〕 Ethem Alpaydin, “Introduction to Machine Learning”, MIT Press, 2004.
〔14〕 Artificial Neural Network, http://en.wikipedia.org/wiki/Artificial_neural_network.
〔15〕 Lei Yu and Huan Liu, “Feature selection for high-dimensional data: A fast correlation-based filter solution”, Proceedings of the International Conference on Machine Learning, Washington, DC, USA, Aug. 2003, pp.856-863.
〔16〕 Sebastian Zander, Thuy Nguyen, Grenville Armitage, “Automated Traffic Classification and Application Identification using Machine Learning”, Proceedings of the IEEE Conference on Local Computer Networks, Sydney, Australia, Nov. 2005, pp.250-257.

정 광 본 (Kwang Bon Jung) 준회원

2006년 전북대학교, 컴퓨터공학과 학사
2006년 3월~2006년 2월 포항공과대학교, 컴퓨터 공학과 석사
<관심분야> 인터넷 트래픽 모니터링 및 분석, 네트워크 관리 네트워크 보안

최 미 정 (Mi Jung Choi) 정회원

1998년 이화여자대학교, 컴퓨터공학과 학사
1998년~2000년 포항공과대학교, 컴퓨터공학과 석사
2000년~2004년 포항공과대학교, 컴퓨터공학과 박사
2004년 Post-Doc., Dept. of computer Science and Engineering, POSTECH, Korea
2004년~2005년 Post-Doc., MADYNES Team, LORIA-INRIA Lorraine, Nancy, France
2005년~2006년 Post-Doc., School of Computer Science, Univ. of Waterloo, Canada
2006~현재 포항공과대학교, 컴퓨터공학과 연구교수
<관심분야> XML 기반의 네트워크 관리, 에이전트 기술, 정책 기반의 네트워크 관리.

김 명 섭 (Myung Sup Kim) 정회원

1998년 포항공과대학교, 전자계산학과 학사
1998년~2000년 포항공과대학교, 컴퓨터공학과 석사
2000년~2004년 포항공과대학교 컴퓨터공학과 박사.
2004년~2006년 Post-Doc., Dept. of ECE, Univ. of Toronto, Canada.
2006년~현재 고려대학교, 컴퓨터정보학과 조교수.
<관심분야> 네트워크 관리 및 보안, 트래픽 모니터링 및 분석, 멀티미디어 네트워크

원 영 준 (Young J. Won) 정회원

2003년 Univ. of Waterloo, BMath in Computer Science
2004년~2006년 포항공과대학교 컴퓨터공학과 석사
2006년~현재 포항공과대학교 컴퓨터공학과 박사과정
<관심분야> 인터넷 트래픽 모니터링 및 분석, 네트워크 운용 및 시스템 관리, 네트워크 보안

홍 원 기 (James W. Hong) 정회원

1983년 Univ. of Western Ontario, BSc in Computer Science
1985년 Univ, of Western Ontario, MS in Computer Science
1985년~1986년 Univ, of Western Ontario, Lecturer
1986년~1991년 Univ, of Waterloo, PhD in Computer Science
1991년~1992년 Univ, of Waterloo, Post-Doc Fellow
1992년~1995년 Univ, of Western Ontario, 연구교수
1995년~현재 포항공과대학교, 컴퓨터공학과 교수
2005년~현재 IEEE Comsoc CNOM Chair
<관심분야> 네트워크 트래픽 모니터링, 네트워크 및 시스템 관리, Network Security