논문 21-46-02-17 The Journal of Korean Institute of Communications and Information Sciences '21-02 Vol.46 No.02
https://doi.org/10.7840/kics.2021.46.2.324

PDCCH 채널 학습을 통한 모바일 서비스 컨텐츠 예측

문 성 우◆, 김 성 현*, 신 홍 기**, 천 경 열***, 윤 현 구****, 최 용 훈°

Mobile Service Content Prediction Method through PDCCH Channel Learning

Sung-woo Moon◆, Sung-hyun Kim*, Hong-gi Shin**, Kyung-yul Cheon***,
Hyungoo Yoon****, Yong-Hoon Choi°

요 약

트래픽의 응용 별 분류 기법들이 다양하게 연구되었으나 대부분의 기술들은 패킷 레벨에서 헤더들을 분석하거나, 페이로드의 시그니쳐를 분석하여 트래픽을 분류하고 있다. 본 논문에서는 물리 계층의 PDCCH (Physical Downlink Control Channel) 정보만을 학습하여 모바일 서비스 컨텐츠를 분류하는 방법을 제시한다. 제안하는 방법은 물리계층의 정보만을 이용하므로 암호화된 패킷에도 적용할 수 있다. 실험을 위하여, 현재 운용중인 LTE(Long Term Evolution) 기지국으로부터 PDCCH 채널 정보들을 수집하였으며, 분류 기법으로 랜덤 포레스트, SVM, AutoEncoder, 심층 신경망, 합성곱 신경망 기법들을 이용하여 모바일 서비스 컨텐츠 예측을 수행하였다. 예측 정확도는 최대 99% 임을 확인하였다.

Key Words : Service Content Prediction, KNN, Decision Tree, SVM, Random Forest, Ensemble, Deep Learning

ABSTRACT

Although various traffic classification techniques have been studied for each application, most of the technologies classify traffic by analyzing headers or payload signatures at the packet level. In this paper, we propose a method for classifying mobile service contents by learning only the physical downlink control channel (PDCCH) information of the physical layer. Since the proposed method uses only information of the physical layer, it can be applied to encrypted packets. For the experiment, PDCCH channel information was collected from the currently operating LTE (Long Term Evolution) base station, and mobile service content prediction was performed using random forest, SVM, AutoEncoder, deep neural network, and convolutional neural network as classification techniques. The prediction accuracy was observed up to 99%.

※ This work was supported by Institute for Information Communications Technology Promotion(IITP) grant funded by the Korea government(MSIT) (No.2017-0-00109, Development of Frequency Analysis Technology for the Virtuous Circulation of Radio Resource)
※ 본 연구는 2019년도 광운대학교 교내학술연구비의 지원을 받아 수행되었음.
※ This research was results of a study on the "HPC Support" Project, supported by the 'Ministry of Science and ICT' and NIPA.
◆ First Author : Kwangwoon University, Division of Robotics, tjddn3322@naver.com, 학생(석사), 학생회원
° Corresponding Author : Kwangwoon University, Division of Robotics, yhchoi@kw.ac.kr, 정교수, 종신회원
* Kwangwoon University, Division of Robotics, dandyleader@naver.com, 학생(석사)
** (주)네오위즈, ghdrl95@gmail.com, 연구원, 정회원
*** ETRI, kycheon@etri.re.kr, 정회원
**** Myongji College, hgyoon@mjc.ac.kr, 정회원
논문번호 : 202009-240-C-RN, Received September 27, 2020; Revised December 1, 2020 Accepted December 1, 2020

Ⅰ. 서 론

트래픽의 응용 별 분류 기법들이 다양하게 연구되었다. 대부분의 기술들은 패킷 레벨에서 헤더들을 분석하거나, 페이로드의 시그니쳐를 분석하여 트래픽을 분류하므로, 관련 패킷들을 관찰하여야 한다. 인터넷 트래픽의 응용 서비스별 분류에 대한 폭넓은 서베이[1]는 참고문헌 [1]에 자세히 정리되어 있다. 본 논문에서는 물리계층의 PDCCH (Physical Downlink Control Channel) 채널 정보만으로 서비스 컨텐츠를 예측하는 기술을 살펴본다.

트래픽을 분석하는 가장 간단한 방법은 IANA (Internet Assigned Numbers Authority)에서 지정한 포트 번호를 비교해 분류하는 방법이다. 하지만 요즘 사용되는 응용 애플리케이션은 방화벽을 원활히 통과하기 위해 동적 혹은 알려진 포트(Well-Known port)를 사용하기 때문에 70% 미만의 정확도를 가진다[2]. 이 외에 응용 애플리케이션을 예측하는 보편적인 방법으로는 헤더 시그니쳐, 페이로드 시그니쳐, 통계정보 시그니쳐 기반 트래픽 분류 기법이 있다[3]. 헤더 시그니쳐 기반 트래픽 분류 기법은 서버 IP (Internet Protocol) 주소와 포트 번호를 이용해 매우 빠른 속도로 응용 애플리케이션을 분류할 수 있다[4]. 하지만 분류 범위가 넓고, 둘 이상의 포트를 가지거나 임의의 포트를 설정할 수 있는 복잡한 구조를 가진 응용 애플리케이션 같은 경우 적합하지 못하다. 페이로드 시그니쳐 기반 트래픽 분류는 데이터부의 유일한 공통 문자열을 사용한 분류로 정확도가 매우 높은 장점을 가지고 있다[5]. 하지만 시그니쳐 추출 작업에 시간과 인력 소비가 크고 추출과정이 까다로울 뿐만 아니라 암호화된 트래픽을 분류할 수 없다. 통계정보 시그니쳐 기반 트래픽 분류는 패킷 사이즈, 시간과 같은 통계적 정보를 바탕으로 응용 트래픽을 분류하는 방법이다[6]. 암호화된 트래픽을 분류할 수 있는 장점이 있지만, 정확도 면에서 페이로드 시그니쳐에 비해 낮고, 시그니쳐를 추출할 수 있는 응용 애플리케이션이 한정적이다. 이러한 예측 방법들은 패킷레벨 즉 IP 헤더 값들과 시그니쳐를 이용한 예측 방법으로 연구가 다양하게 진행되어 왔다. 하지만 아직까지 물리 계층 (Physical layer)의 데이터를 이용해 응용 애플리케이션을 예측하는 연구는 부족하다.

본 논문에서는 기계학습 기반 물리 계층의 PDCCH 데이터를 이용한 응용 애플리케이션 예측 방법을 제안한다. 머신러닝과 딥러닝은 다양한 분야에 적용되고 있다. 전통적인 알고리즘 기법은 복잡한 분류 문제나 동적으로 변하는 데이터 분류에서 낮은 정확도를 갖는다. 머신러닝은 데이터에 대한 패턴과 규칙을 자율적으로 학습하기 때문에 (PDCCH와 같은) 비정형 데이터에 대한 분류문제에서 기존의 방식보다 높은 정확도를 갖는다. 딥러닝 기법 중 심층 신경망 (Deep Neural Network), 합성곱 신경망 (Convolution Neural Network), 오토인코더 (AutoEncoder) 기법으로 컨텐츠 예측 실험을 진행하고, 머신러닝 기법 중 서포트 벡터 머신 (Support Vector Machine)과 의사결정나무 (Decision tree)를 앙상블 (Ensemble)한 랜덤 포레스트 (RandomForest) 기법을 통해 모바일 서비스 컨텐츠를 예측한다. 물리계층의 정보만을 사용하기 때문에 암호화된 패킷에서도 적용하여 컨텐츠를 예측할 수 있다. 또한 서비스 컨텐츠 예측을 통해 한정된 자원에 프로비저닝 (Provisioning)이 가능해져 자원을 효율적으로 사용할 수 있게 된다.

본 논문은 서론에 이어 2장에서는 다양한 학습 기법을 이용하여 컨텐츠를 예측한 방법에 대해 기술한다. 3장에서는 실험에 사용한 데이터를 포함한 실험환경과 실험결과에 대해 기술하고, 중요한 입력 컬럼에 대해 확인한 후, 적합성을 검증한다. 마지막으로 4장에서는 결론 및 향후 연구에 대해 기술한다.

Ⅱ. 학습 기법

본 장에서는 심층 신경망, 합성곱 신경망, 오토 인코더(AutoEncoder)를 적용한 딥러닝 기법과, 머신러닝 기법인 랜덤 포레스트와 서포트 벡터 머신을 사용한 모바일 서비스 컨텐츠 예측 방법에 대해 설명한다.

2.1 심층 신경망 (DNN)

심층 신경망 (DNN)은 입력층 (Input layer), 은닉층 (Hidden layer), 출력층 (output layer)으로 이뤄진 인공신경망 (Artificial Nebral Network, ANN)이다. 심층 신경망은 은닉층을 2개 이상 지닌 학습 방법으로 복잡한 비선형 관계들을 모델링 할 수 있다. 학습 데이터가 많아지면 성능이 좋아지는 장점과 특징점 추출 (Feature extraction)이 자동으로 이루어져 변수 선택의 번거로움이 없어진다.

본 논문에서 심층 신경망을 사용해 모바일 서비스 컨텐츠를 예측한 모델 구조는 그림 1과 같다. PDCCH의 데이터 32개 중 29개의 데이터를 입력으로 사용했는데, PDCCH 데이터에 대한 자세한 설명은 3.1절 실험환경에서 설명하도록 하겠다. 학습에 필요한 최적의 하이퍼 파라미터를 적용하기 위해 베이지안 최적화

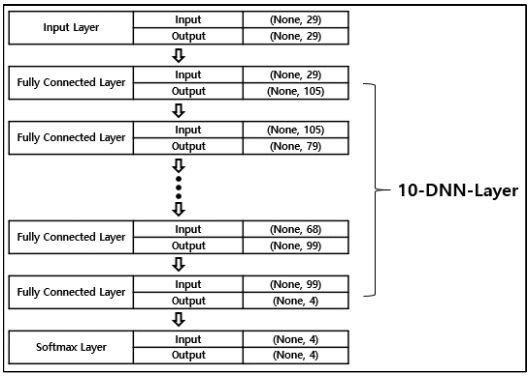


그림 1. 10계층 심층 신경망 모델 구조
Fig. 1. Model architecture of DNN

(Bayesian optimization)기법을 사용했다[7]. 학습률 (Learning rate), 은닉층 셀의 수 (Hidden Layer cell size), 에폭 (Epoch)과 같은 하이퍼 파라미터 값을 표 1과 같이 범위를 설정 한 후, 결과가 가장 좋은 하이퍼 파라미터 값을 사용했다. 마지막 계층은 소프트맥스 (softmax)로 구성해 4개의 정답 (VoD, VoIP, FTP, Web)이 각각 얼마의 확률로 정답을 예측하는지 확인했다.

표 1. 10계층 심층 신경망 하이퍼 파라미터 및 범위
Table 1. 10-Layer DNN Hyper parameter and range

Hyperparameter	Range	Value
Learning Rate	0.0001 -0.01	0.0018
Number of Hidden Layer cell size	43 - 145	105, 79, 92, 46, 63, 51, 61, 98, 68, 99
Epoch	80 - 180	129
Batch size	50 - 60	50
Dropout	0.4 - 1	0.93
Hidden Layer Learning Rate Dacay	0.6 - 1	0.6465
Weight Initializer (fix)		Xavier Initializer
Optimizer (fix)		Adam

2.2 합성곱 신경망(CNN)

합성곱 신경망 (CNN)은 전처리 (Preprocess)를 사용하여 합성곱 (Convolution) 연산을 사용하는 층이 있는 신경망이다. 합성곱 신경망은 보통 합성곱 층과 풀링(Pooling) 층으로 이루어져 있다. 합성곱과 풀링을 통해 다양한 특징들을 추출해 낼 수 있다. 심층 신경망과 비교해 볼 때 파라미터양이 적고, 특히 신호처리와 이미지 처리에 효과적인 신경망이며 데이터에 대한 특

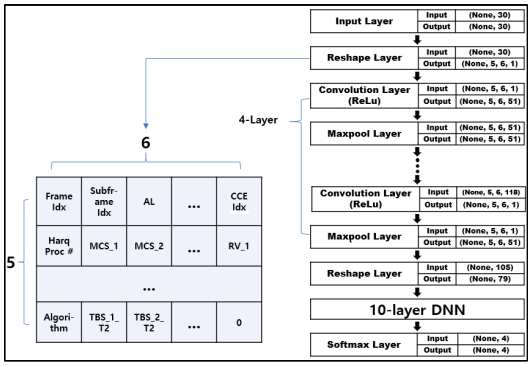


그림 2. 합성곱신경망 + 10계층 심층 신경망 모델 구조
Fig. 2. Model architecture of CNN + 10-layer DNN

정한 특징 정보를 잘 추출한다[8].

1차원 데이터로 구성된 PDCCH 데이터에 합성곱 신경망을 적용하기 위해서 마지막 컬럼에 0을 추가하여 그림 2와 같이 2차원 데이터로 만들었다. 형태 변경 (Reshape) 과정 을 거쳐 만들어진 5*6 데이터를 합성곱 계층에 입력으로 사용했다. 주변을 0으로 채우는 제로패딩 (Zero Padding) 기법을 사용했고, 필터 크기는 2*2로 구성해 합성곱을 진행했다. 활성화 함수 (Activation function)으로 ReLu (Rectified Linear Unit)를 통과시켰다. 마지막으로 그림 2와 같이 최대풀링층 (Max Polling layer)을 통과시키는 총 4층의 합성

표 2. 합성곱 신경망 + 10계층 심층 신경망 하이퍼 파라미터 및 범위
Table 2. CNN + 10-Layer DNN Hyper parameter and range

Hyperparameter	Range	Value
Learning rate	0.0001-0.01	0.008242
Number of Hidden Layer Cell Size	192 - 640	438, 396, 466, 426, 569, 597, 313, 235, 204, 235
Epoch	100 - 300	102
Batch size	100, 120, 150	100
DNN Dropout	0.4 - 1	0.951639
CNN Dropout	0.4 - 1	0.986297
Number of CNN Channel Size	30 - 128	51, 109, 118, 70
Hidden Layer Learning Rate Decay	0.6 - 1	0.7732
Weight Initializer (fix)		Xavier Inintializer
optimizer (fix)		Adam

곱 신경망 층을 만들었다. 합성곱 신경망에서 나온 출력을 다시 1차원 데이터로 형태를 변형시켜 2.1절의 10계층 심층 신경망을 통과한 후 소프트맥스 계층을 통해 각 확률 값을 확인했다. 은닉층의 셀 (cell) 개수, 채널 개수, 에폭, 드롭아웃 (dropout)과 같은 하이퍼 파라미터 값은 2.1절과 같이 베이지안 최적화를 사용해 표 2와 같이 범위를 정한 후 정확도가 가장 높은 값을 사용했다.

2.3 Stacked AutoEncoder

Stacked AutoEncoder는 인코더 (Encoder)와 디코더 (Decoder)가 합쳐진 구조로 입력과 출력이 같도록 하는 구조이고 여러 개의 은닉층을 가지는 구조이다. 노이즈 제거에 탁월하며, 인코딩 과정에서 차원을 축소하며 특징 점을 잘 뽑아낸다. 기지국에서 측정한 PDCCH 데이터는 여러 단말기를 통해 측정된 값이기 때문에 노이즈가 끼어있을 확률이 있고, 많은 데이터 중 특징 점을 잘 뽑아낼 수 있기 때문에 Stacked AutoEncoder를 사용했다[9]. Stacked AutoEncoder의 경우 정해야 할 하이퍼 파라미터수가 적어 여러 번의 실험을 통해 학습률은 0.0082, 에폭은 100으로 설정했다. 모델의 구조는 그림 3과 같이 구성했다. 29개의 입력을 은닉 셀의 개수가 각각 15개, 4개인 인코더로 구성했고, 4개의 특징 값을 은닉 셀의 개수가 각각 15개, 29개가 되게 디코더를 구성했다. 마지막은 fully connected layer로 구성했고 정답과 비교를 진행하며 학습을 진행했다.

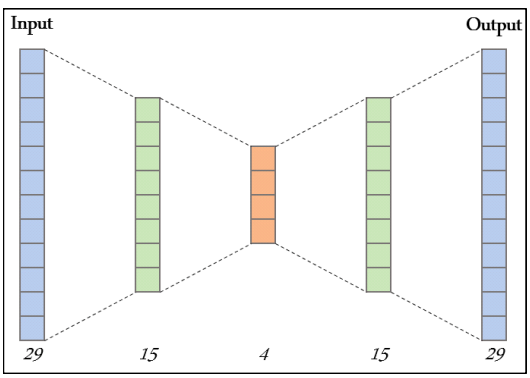


그림 3. AutoEncoder 모델 구조
Fig. 3. Model architecture of AutoEncoder

2.4 랜덤 포레스트 (Random Forest)

머신러닝 기법중 가장 잘 알려진 랜덤 포레스트는 앙상블 (Ensemble) 기법의 한 종류로 배깅 (Bagging)에 속한 대표적인 기법이다. 랜덤 포레스트는 여러 개의 의사결정 나무 (Decision Tree)를 기반으로 구성되어 있다. 의사결정 나무는 데이터의 분석을 통해 데이터 사이에 존재하는 패턴을 예측 가능한 규칙들의 집합으로 생성하는 기법이다[10]. 랜덤 포레스트를 구성하는 각각의 의사결정 나무는 전체 학습 데이터를 부트스트래핑 (Bootstraping) 방식으로 샘플링 (Sampling)하여 개별적으로 학습하고 예측을 진행한다. 의사결정 나무들이 예측한 결과는 투표 (Voting) 방식을 통해 데이터에 대한 최종적인 정답 예측을 진행한다. 앙상블 기법을 사용함으로써 개별 모델에 대한 성능을 분산시키기 때문에 과 적합 (Overfitting)이 감소되며 단일 모델보다 향상된 성능을 보인다. 딥러닝 기법과 마찬가지로 PDCCH 데이터 32개 중 정답 데이터와 학습에 불필요한 데이터 2개를 제거한 29개의 데이터를 입력으로 사용한다. 적용된 하이퍼 파라미터는 표 3과 같다.

표 3. 랜덤 포레스트 하이퍼 파라미터 정의
Table 3. Define RandomForest Hyperparameters

Hyperparameter	Value
n_estimators	30
n_jobs	16
criterion	gini
max_depth	2
min_samples_split	2
min_samples_leaf	1
min_weight_fraction_leaf	0.0
max_features	auto
max_leaf_node	none

본 실험에는 30개의 의사결정 나무 (Decision Tree)로 구성되어있는 랜덤 포레스트를 사용하였다. 기준 (Criterion)의 경우는 지니 불순도 (Gini Impurity)를 사용했다. 의사 결정 나무의 과적합 문제를 해결하기 위해 트리의 최대 깊이는 2, 자식노드가 없는 노드는 1, 자식노드를 갖기 위한 최대한의 데이터 개수는 1로 지정했다.

2.5 SVM (Support Vector Machine)

서포트 벡터 머신은 서포트 벡터 (Support Vector)와, 분류를 위해 최대한의 마진 (Margin)을 갖는 결정 경계 (Decision Boundary)인 초평면 (Hyperplane)을 정의하여 균등한 위치에 분류 기준을 세운 후 데이터를 분류하는 기법이다. 서포트 벡터 머신은 분류나 예측문제에 뛰어난 성능 을 가지고 있으며, 다른 기법과 비교하여 과 적합이 발생할 확률이 적다[11]. 적용된 하이퍼 파라미터는 표 4와 같다. 최적의 비선형 커널

표 4. 서포트 벡터 머신 하이퍼 파라미터
Table 4. SVM Hyperparameters

Hyperparameter	Value
box constraint level	1
kernel	Gaussian
kernel scale	1.3
multiclass method	one-vs-one
standardize data	true

(Kernel)을 찾기 위해 MATLAB을 사용하였다. 선형 (Linear), 조밀 가우시안 (Fine Gaussian), 중간 가우시안 (Medium Gaussian), 성간 가우시안 (Coarse Gaussian) 커널을 적용해 본 결과 조밀 가우시안이 가장 높은 정확도를 보였다. 커널 스케일은 1.3을 적용하였으며, 마진을 위반하는 예측에 최대 벌점을 제어하고, 과 적합을 방지하는 상자 제약 조건(Box Constraint Level)은 1로 설정하였다.

Ⅲ. 실 험

본 장에서는 실험 데이터를 포함한 전체적인 실험환경, 실험결과 및 예측에 많은 영향을 주는 주요 컬럼에 대해 설명한다.

3.1 실험환경

본 실험에서는 LTE (Long Term Evolution)기지국에서 측정한 PDCCH 데이터로 컨텐츠 예측 실험을 진행했다. 전체 866,779개의 데이터로, 각각 VOD 38507개, VOIP 20738개, FTP 773393개, WeP 34141개로 이루어져 있다. PDCCH의 각 데이터에 대한 설명은 표 5에 정리되어 있고, 총 32개의 컬럼으로 구성되어 있다. 2번째 데이터인 PCI 값은 구별 값으로 사용이 불가하여 학습에서 제외시키고, 3번째 데이터인 Total RB의 값은 모두 100으로 되어있어 학습에서 제외시켰다. 6번째 데이터인 DCI Format은 문자열 데이터 타입을 가지고 있어, 숫자로 변환하기 위해 Format0을 1, Format1A를 2, Format2A를 3, Format1을 4, Format2를 5, 4B를 6으로 각각 대응시켜 변환했다. 5번째 데이터인 RNTI값은 구별될 클래스의 값을 가지고 있어 정답으로 사용해 총 30개의 데이터를 학습에 사용했다.

딥러닝 기법을 사용한 실험의 경우 GPU GeForce GTX 1060이 장착된 워크스테이션에서 python 기반 Tensorflow를 사용해 실험을 진행했다. 전체 데이터의 80%(693,423개)를 학습데이터로 사용했고, 나머지 20% (173,356개)를 테스트 데이터로 사용해 예측 정확도를 확인했다. 학습에 입력으로 넣어지는 데이터는 셔플하여 랜덤으로 데이터가 들어가게 학습을 진행했다.

표 5. PDCCH 구성 데이터
Table 5. PDCCH Configuration Data

Number	Item	Description
0	Frame Idx	Frame Number(10ms)
1	Subframe Idx	SFN(1ms)
2	PCI	Pysical Cell ID
3	# Total RB	Number of Total RB
4	AL	Aggregation Level: number of using CCE(1, 2, 4, 8)
5	RNTI	Radio Network Temporary Indicator
6	DCI Format	Format0: UL, The rest Format1~2: DL
7	DCI Length	DCI Length
8	CCE Idx	CCE Idx
9	Harq Proc #	DL retransmission execution processor ID (total number: 8)
10	MCS_1	MCS Index(codeword 1)
11	MCS_2	MCS Index(codeword 2)
12	TBS_1_T1	Transmission Block Size_Table1(64QAM)-codeword1
13	TBS_2_T2	Transmission Block Size_Table1(64QAM)-codeword2
14	RV_1	HARQ_Redundancy version-codeword1
15	RV_2	HARQ_Redundancy version-codeword2
16	NDI_1	New Data Indicator-codeword1
17	NDI_2	New Data Indicator-codeword2
18	# Sch. RB	Allocation RB
19	CFI	Conrol format Indicator: Number of symbols assigned to Control Channel(1~3)
20	# Tx. Ant	Number of transmit antenna
21	Precode Idx	Precode Index
22	Rank	Rank Indicator
23 ~ 27	Value of algorithm	Value of algorithm
28	TBS_1_T2	Transmission Block Size_Table1(256QAM)-codeword1
29	TBS_2_T2	Transmission Block Size_Table1(256QAM)-codeword2
30	SNR 0	S/N ratio 0
31	SNR 1	S/N ratio 1

머신러닝 기법을 사용한 실험의 경우 고성능 CPU(Intel i9-9900K)가 장착된 워크스테이션에서 Python 기반의 scikit-learn 오픈소스 기계 학습 라이브러리와 MATLAB를 사용하여 학습 및 실험 환경을 구성하였다. 딥러닝과 마찬가지로 전체 데이터 중 80%를 학습데이터로, 나머지 20%를 테스트 데이터로 사용했다. 모델 학습 시 테스트 데이터에 과 적합이 이루어지는 현상을 방지하기 위해 테스트 데이터를 하나로 고정하지 않고 데이터의 모든 부분을 사용하여 모델을 검증하는 교차 검증 (Cross Validation)을 5회 적용하였다.

3.2 실험결과

딥러닝 기법(심층 신경망, 합성곱 신경망, AutoEncoder)과 머신러닝 기법(랜덤 포레스트, 서포트 벡터 머신)을 사용해 실험을 진행한 결과는 표 6과 같다. 딥러닝 기법 중 합성곱 신경망과 10-계층 심층 신경망을 사용한 기법이 97.72%로 높게 나왔고, 머신러닝 기법 중 랜덤 포레스트를 사용한 기법이 99.02%로 가장 높게 나왔다. 10계층 심층 신경망만을 사용했을 때 보다 합성곱 신경망을 같이 사용했을 때 성능이 약 5.67% 올랐다. 합성곱 신경망을 사용할 경우 합성곱을 통해 특징점을 뽑아내고, 최대풀링을 통해 한번 더 특징점을 뽑아내기 때문에 성능이 좋아진 것이라 판단된다. 그림 4와 5는 혼돈행렬 (Confusion Matrix)을 나타냈다. X축은 정답에 대한 항목이며 Y축은 예측에 대한 항목이다. 특정 정답 라벨 (label)의 항목과 예측 라벨의 항목이 일치하는 구간이 예측에 성공한 경우에 대한 지표이다. 항목이 일치하는 구간이 1에 수렴하며 예측을 잘하는 모습을 볼 수 있다. 랜덤 포레스트를 사용

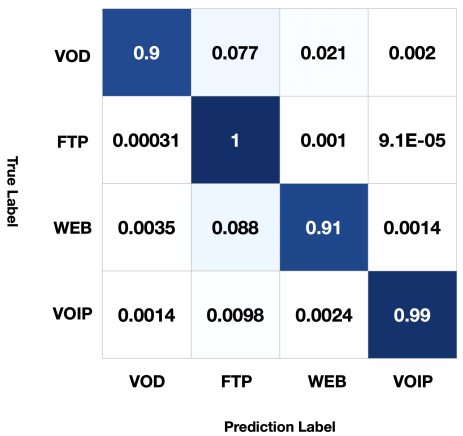


그림 4. 랜덤 포레스트의 혼동 행렬 (단위 : %)
Fig. 4. Confusion Matrix of RandomForest (Unit : %)

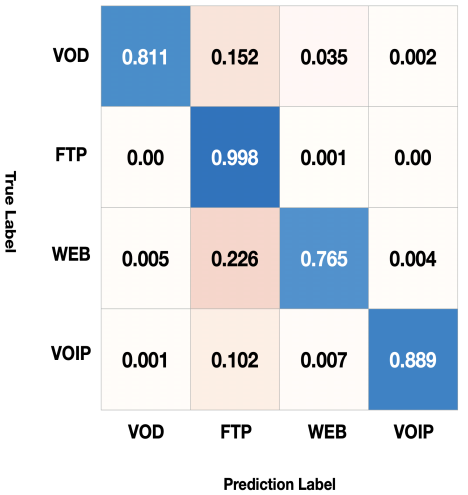


그림 5. 서포트 벡터 머신 의 혼동 행렬 (단위 : %)
Fig. 5. Confusion Matrix of SVM (Unit : %)

표 6. 실험결과
Table 6. experiment results

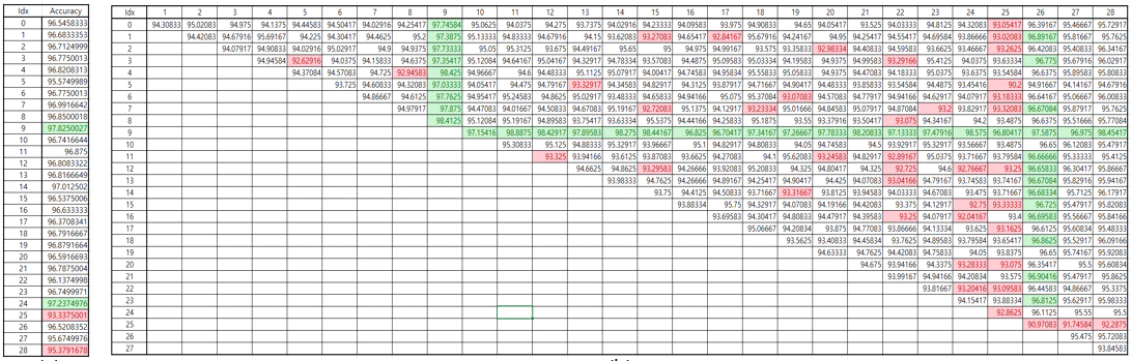
Method	Accuracy
DNN	92.05%
CNN	**97.72%**
AutoEncoder	95.40%
SVM	97.80%
Random Forest	**99.02%**

한 경우 각각의 연관성이 낮은 의사결정 나무가 서로 다른 관점으로 데이터를 분류한 후 배깅을 통해 결과 예측을 조합함으로써 새로운 데이터에 대한 일반화 (Generalize)가 잘되어 정확도가 높은 것이라 판단된다.

3.3 주요 컬럼 확인

모바일 컨텐츠 예측에 중요도가 높은 데이터를 확인하기 위해 딥러닝 기법 중 정확도가 가장 높았던 합성곱신경망 + 10계층 심층 신경망기법을 사용해 입력 컬럼을 1개, 2개씩 제거해 컨텐츠를 예측하는 실험과, 머신러닝 기법 중 랜덤 포레스트 기법을 사용해 중요도를 확인하는 실험을 진행했다. 그림 6의 (a)는 입력 컬럼이 하나씩 빠졌을 때의 정확도이다. (b)는 입력 컬럼이 두 개 씩 빠졌을 때의 결과 값이다. 초록색으로 표시한 부분은 상위 10%의 정확도를 가지는 값이고, 빨간색으로 표시한 부분은 하위 10% 정확도를 가지는 값이다.

(a)와 (b)를 보면 9번 컬럼 (TBS_1_T1)을 제거했음



(a) (b)

그림 6. (a): 컬럼을 하나씩 제거했을 때 정확도, (b): 컬럼을 두 개씩 제거했을 때 정확도 (단위 : %)
Fig. 6. (a): Accuracy when columns are removed one by one, (b): Accuracy when two coms are removed. (Unit : %)

에도 상위 10%의 정확도를 보이며 학습에 중요도가 낮음을 확인할 수 있다. 25번 컬럼 (TBS_1_T2)을 제거했을 경우 하위 10%의 정확도를 가장 많이 보였고, 28번 컬럼 (SNR 1)을 제거했을 때 가장 낮은 정확도를 보이며 상대적으로 중요도가 높음을 확인했다. 랜덤 포레스트 학습 후 각 컬럼별 중요도 확인 결과는 그림 7과 같이 SNR1, algorithm monitoring2, #Sch. RB 데이터가 높은 중요도를 보이는 것을 확인할 수 있다.

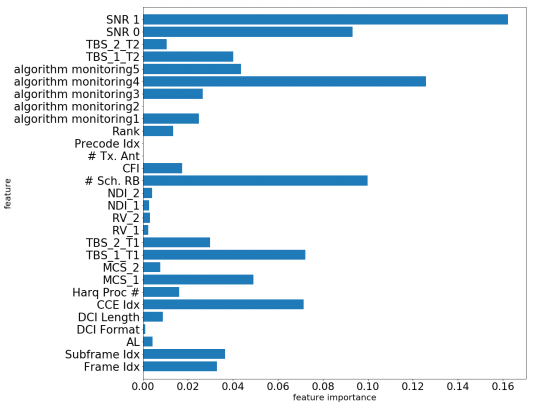


그림 7. 랜덤 포레스트의 특성 중요도 시각화 (단위 : %)
Fig. 7. Feature importance of RandomForest (Unit : %)

Ⅳ. 결 론

본 논문에서는 물리계층의 PDCCH 데이터와 딥러닝의 심층 신경망, 합성곱 신경망, AutoEncoder, 머신러닝의 서포트 벡터 머신, 랜덤 포레스트 기법을 사용하여 모바일 서비스 컨텐츠 예측을 제안했다. 실험 결과 모두 92% 이상의 정확도를 얻었고, 최대 99%의 정확도를 얻어 물리계층의 PDCCH 데이터를 이용해 모바일 서비스 컨텐츠 예측이 가능하다는 것을 확인했다. 예측에 있어 데이터의 중요도를 평가하기 위해 입력 컬럼을 제거하는 실험과 랜덤 포레스트를 이용한 실험을 진행한 결과 TBS_1_T2, SNR 1, algorighm Monitoring4, #Sch. RB 데이터가 분류하는데 있어 높은 중요도를 보이며, algorithm monitoring2, TBS_1_T1 데이터는 중요도가 낮음을 확인했다.

추 후 연구로 머신러닝의 경우 관련도가 적은 데이터 제거와 정규화 (regularization)를 통해 모델 학습 최적화를 진행할 계획이다. 딥러닝의 경우 PDCCH의 시간적인 특성을 고려하여, 딥러닝 모델 중 RNN (Recurrent Neural Network)계열의 LSTM (Long Short-Term Memory Network), GRU (Gated Recurrent Unit), Transformer와 같은 모델을 사용해 컨텐츠 예측 실험을 진행 할 계획이다.

References

[1] Thuy T.T. Nguyen and G. Armitage, "A survey of techniques for internet traffic classification using machine learning," *IEEE Commun. Surv. & Tuts.*, Fourth Quarter, vol. 10, no. 4, pp. 56-76, 2009.

[2] A. W. Moore and K. Papagiannaki, "Toward the accurate identification of network applications," *Int. Wkshp. Passive and Active Netw. Measurement,* pp. 41-54, Springer, Berlin, Heidelberg, 2005.

[3] Y.-H. Goo, S. Lee, K.-S. Shim, and M.-S. Kim, "A traffic-classification method using the correlation of the network flow," *J. KIISE*, vol.

44, no. 4, pp. 433-438, Apr. 2017.

[4] S.-H. Yoon and M.-S. Kim, “Research on signature maintenance method for internet application traffic identification using header signatures,” *J. Internet Comput. and Serv.*, vol. 12. no. 6, pp. 19-33 Dec. 2011.

[5] J.-S. Park, S.-H. Yoon, and M.-S. Kim, “Performance improvement of the payload signature based traffic classification system using application traffic locality,” *J. KICS*, vol. 38, no. 7, pp. 519-525, Jul. 2013.

[6] H.-M. An, J.-H. Ham, and M. S. Kim, “Performance improvement of the statistical information based traffic identification system,” *KIPS Trans. Comput. and Commun. Syst.*, vol. 2, no. 8, pp. 335-342 Aug. 2013.

[7] J. Snoek, H. Larochelle, and R. P. Adams, “Practical bayesian optimization of machine learning algorithms,” in *Advances in NIPS*, pp. 2951-2959, 2012.

[8] P. Y. Simard, D. Steinkraus, and J. C. Platt, “Best practices for convolutional neural networks applied to visual document analysis,” *Icdar.*, vol. 3, no. 2003, Aug. 2003.

[9] Y. Qi, Y. Wang, X. Zheng, and Z. Wu, “Robust feature learning by stacked autoencoder with maximum correntropy criterion,” *IEEE ICASSP*, pp. 6716-6720, 2014.

[10] L. Xu, X. Zhou, Y. Ren, and Y. Qin, “A traffic classification method based on packet transport layer payload by ensemble learning,” *ISCC* 2019, pp. 1-6, Barcelona, Spain, Jul. 2020.

[11] Z. Li, R. Yuan, and X. Guan, “Accurate classification if the internet traffic based on the svm method,” *IEEE Int. Conf. Commun. 2007*, pp. 1373-1378, Glasgow, UK, Jun. 2007.

문 성 우 (Sung-woo Moon)

2020년 2월 : 광운대학교 로봇학부 졸업 공학사

2020년 3월~현재 : 광운대학교 로봇학과 석사과정

<관심분야> 통신 네트워크, 음성인식 및 합성, 주식투자 알고리즘

[ORCID:0000-0002-8002-1754]

김 성 현 (Sung-hyun Kim)

2020년 2월 : 경기과학기술대학교 컴퓨터모바일융합과 전문학사

2020년 8월 : 국가평생교육진흥원 학점은행제 컴퓨터공학 학사

2020년 9월~현재 : 광운대학교 로봇학과 석사과정

<관심분야> 통신 네트워크, 머신러닝, 음성인식 및 합성

신 홍 기 (Hong-gi Shin)

2017년 2월 : 경기과학기술대학교 컴퓨터모바일융합과 전문학사

2017년 8월 : 국가평생교육진흥원 학점은행제 컴퓨터공학 학사

2019년 8월 : 광운대학교 로봇학과 공학석사

2019년 12월~현재 : (주)네오위즈 연구원

<관심분야> 통신 네트워크, 머신러닝, 실내 측위

천 경 열 (Kyung-yul Cheon)

1998년 2월 : 고려대학교 전자공학과 공학사
2000년 2월 : 서울대학교 전자공학과 공학석사
2000년 2월~2004년 2월 : ㈜현대전자 주임연구원
2004년 2월~2005년 3월 : 고등기술연구원 주임연구원
2005년 4월~현재 : 전자통신연구원 책임연구원
<관심분야> 통신 시스템, 무선자원관리

윤 현 구 (Hyungoo Yoon)

1995년 2월 : 연세대학교 전자공학과 (공학사)
1997년 2월 : 연세대학교 전자공학과 (공학석사)
2002년 8월 : 연세대학교 전기전자공학과 (공학박사)
2002년 2월~2004년 2월 : (주)현대시스콤 선임연구원
2004년 3월~현재 : 명지전문대학 전자공학과 교수
<관심분야> 통신시스템, 무선자원관리, 간섭회피방안

최 용 훈 (Yong-Hoon Choi)

1995년 2월 : 연세대학교 전자공학과 공학사
1997년 2월 : 연세대학교 전자공학과 공학석사
2001년 2월 : 연세대학교 전기전자공학과 공학박사
2001년 4월~2002년 3월 : (미)매릴랜드 주립대 Postdoctoral Research Associate
2002년 6월~2005년 8월 : LG전자 책임연구원
2005년 9월~현재 : 광운대학교 로봇학부 교수
<관심분야> 통신 네트워크, 음성인식 및 합성, 머신 러닝