KIPS Trans. Softw. and Data Eng.
Vol.9, No.6 pp.187~194 pISSN: 2287-5905

https://doi.org/10.3745/KTSDE.2020.9.6.187

Tor Network Website Fingerprinting Using Statistical-Based Feature and Ensemble Learning of Traffic Data

Junho Kim† · Wongyum Kim†† · Doosung Hwang†††

ABSTRACT

This paper proposes a website fingerprinting method using ensemble learning over a Tor network that guarantees client anonymity and personal information. We construct a training problem for website fingerprinting from the traffic packets collected in the Tor network, and compare the performance of the website fingerprinting system using tree-based ensemble models. A training feature vector is prepared from the general information, burst, cell sequence length, and cell order that are extracted from the traffic sequence, and the features of each website are represented with a fixed length. For experimental evaluation, we define four learning problems (Wang14, BW, CW_T, CW_H) according to the use of website fingerprinting, and compare the performance with the support vector machine model using CUMUL feature vectors. In the experimental evaluation, the proposed statistical-based training feature representation is superior to the CUMUL feature representation except for the BW case.

Keywords : Anonymous Network, Traffic Collection, Website Fingerprinting, Ensemble Algorithm, Machine Learning

트래픽 데이터의 통계적 기반 특징과 앙상블 학습을 이용한 토르 네트워크 웹사이트 핑거프린팅

김 준 호† · 김 원 겸†† · 황 두 성†††

요 약

본 논문은 클라이언트의 익명성과 개인 정보를 보장하는 토르 네트워크에서 앙상블 학습을 이용한 웹사이트 핑거프린팅 방법을 제안한다. 토르 네트워크에서 수집된 트래픽 패킷들로부터 웹사이트 핑거프린팅을 위한 훈련 문제를 구성하며, 트리 기반 앙상블 모델을 적용한 웹사이트 핑거프린팅 시스템의 성능을 비교한다. 훈련 특징 벡터는 트래픽 시퀀스에서 추출된 범용 정보, 버스트, 셀 시퀀스 길이, 그리고 셀 순서로부터 준비하며, 각 웹사이트의 특징은 고정 길이로 표현된다. 실험 평가를 위해 웹사이트 핑거프린팅의 사용에 따른 4가지 학습 문제(Wang14, BW, CW_T, CW_H)를 정의하고, CUMUL 특징 벡터를 사용한 지지 벡터 기계 모델과 성능을 비교한다. 실험 평가에서, BW 경우를 제외하고 제안하는 통계 기반 훈련 특징 표현이 CUMUL 특징 표현보다 우수하다.

키워드 : 익명 네트워크, 트래픽 수집, 웹사이트 핑거프린팅, 앙상블 알고리즘, 기계학습

1. 서 론

인터넷에서 개인 정보 보호에 대한 인식이 높아지면서, 사생활을 보장하는 웹 서비스의 필요성과 사용 빈도가 증가했다[1]. 토르(The second onion router)는 국가의 검열이나 해커의 공격으로부터 개인 정보 탈취를 막는 저지연(low-latency) 익명 네트워크 웹 서비스로, 히든 서비스(hidden service) 도메인을 사용한다. 토르 네트워크는 3개 이상의 노드[1]가 연결된 릴레이(relay) 구조로 구성되며, 트래픽에 128-bit AES 암호화를 적용하여 512 바이트로 구성된 셀 시퀀스(cell sequence)를 전송한다[2]. 각 노드는 SOCKS 프록시와 TCP 프로토콜을 이용한다. 암호화된 토르 셀 시퀀스는 출구 노드에서 복호화되어

※본 연구는 문화체육관광부 및 한국저작권위원회의 2019년도 저작권 기술개발사업의 연구결과로 수행되었음.
† 준 회 원 : 단국대학교 컴퓨터학과 석사과정
†† 종신회원 : (주)에이아이딥 연구소장
††† 종신회원 : 단국대학교 소프트웨어학과 교수
Manuscript Received : November 15, 2019
First Revision : February 7, 2020
Accepted : March 3, 2020
* Corresponding Author : Doosung Hwang(dshwang@dankook.ac.kr)

1) 입구, 중간, 출구(entry, middle, exit) 노드가 있으며, 중간 노드의 개수는 도메인에 따라 달라진다.

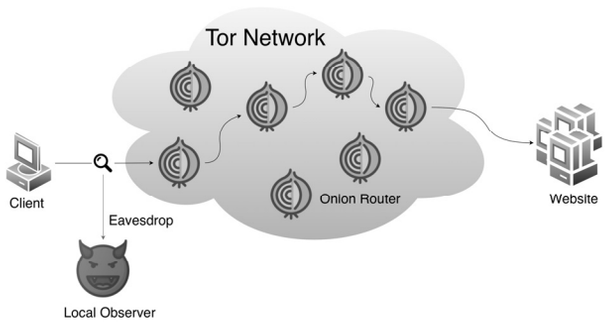

Fig. 1. Tor Network Architecture

웹 서버(web server)에 전송된다. 그러나 내부 감시자(local observer)가 클라이언트와 입구 노드 사이의 트래픽을 도청한다면, 네트워크 트래픽에 대한 분석과 패턴 도출이 가능하다[3]. Fig. 1은 클라이언트가 토르 네트워크를 사용해 웹사이트에 접속하는 과정과 트래픽 도청의 예이다.

기계학습 기반 웹사이트 핑거프린팅(website fingerprinting)은 트래픽의 시간, 순서, 크기 등의 패턴을 이용하여 사용자가 접속한 사이트를 추정하는 수동적 공격이다[4-6]. 모든 웹사이트의 패턴을 생성하는 것은 불가능하기 때문에, 닫힌 세계(closed world)와 열린 세계(open world) 시나리오 등으로 제한된 도메인에서 기계학습 알고리즘의 성능을 비교한다[6-8].

닫힌 세계 시나리오는 클라이언트가 접속한 웹사이트를 구분하는 다중 분류 문제이다. 그러므로 사용자가 접근하지 않은 웹 페이지의 트래픽 데이터는 학습 데이터에 포함되지 않는다. 열린 세계 시나리오는 웹 페이지를 구별하는 것이 아니라 사용자가 감시되고 있는 웹사이트 목록에 접속 유무를 판단하는 이진 분류 문제로 정의되며, 학습 데이터는 모니터링되는 웹 카테고리와 모니터링 되지 않는 카테고리로 구성시킨다[9].

기계학습 기반 웹사이트 핑거프린팅은 트래픽 데이터 수집과 전처리, 모델 학습과 평가 그리고 모델 선택의 단계를 거친다. 본 논문에서는 토르 브라우저(Tor browser)와 WGET[2]으로 트래픽 데이터를 수집한다. 지금까지 제안된 특징 벡터는 트래픽 시퀀스의 최소 길이를 사용하기 때문에 전체 트래픽 데이터의 분포를 반영하지 못한다는 단점이 있다[4,7,8]. 따라서 트래픽 시퀀스에 확률적 모델링을 적용한 고정 길이 특징 벡터를 제안하고, 앙상블(ensemble) 기반 알고리즘을 이용하여 웹사이트 핑거프린팅 분류 모델을 분석한다.

본 논문의 구성은 다음과 같다. 2장에서는 기계학습을 이용한 익명 네트워크 웹사이트 핑거프린팅의 관련 연구를 서술한다. 3장에서는 훈련 데이터 수집 방법을 기술하며, 특징 벡터를 제안한다. 4장은 기 연구된 모델과 제안하는 방법의 학습 성능을 비교, 분석한다. 5장에서는 결론과 향후 연구 방향에 대해 서술한다.

2. 관련 연구

Tao Wang 외 4명[6]은 TCP 패킷을 토르 셀 레이어로 변환하고 SENDME 셀을 제거했다. 그리고 토르 셀의 크기와 방향 정보를 사용해 시퀀스 특징 벡터를 구성했다. 개선된 Levenshtein distance 커널 함수를 제안하였고, 지지 벡터 기계 모델을 이용했다. Alexa Top 100 사이트에 대한 트래픽 데이터를 수집했으며, 닫힌 세계 시나리오에 91.0%의 분류 성능을 보였다.

Kota Abe 외 1명[7]은 SDAE(Stacked Denoising Auto-encoder) 모델을 사용한 웹사이트 핑거프린팅 방법을 제안했다. Tao Wang이 공개한 Wang14 데이터 세트[4]에서, 토르 셀의 방향과 길이 정보로 구성된 특징 벡터를 훈련 데이터로 사용했다. 닫힌 세계 시나리오에 88.0%, 열린 세계 시나리오는 86.0%의 정확도를 나타냈다.

Andriy Panchenko 외 6명[8]은 토르 셀 크기의 누적합을 이용한 CUMUL 특징 벡터를 제안했다. CUMUL 특징 벡터는 패킷 인스턴스 $T=(p_1, \cdots, p_N)$에 대해 수신 패킷일 경우 $p_i>0$, 송신 패킷이면 $p_i<0$으로 정의한다(p_i는 셀의 크기, N은 셀의 수이다). 패킷 인스턴스의 누적합은 $C(T)=((0,0), (a_1,c_1), \cdots, (a_N,c_N))$이며 $c_1=p_1,\ c_i=c_{i-1}+p_i,\ a_i=a_{i-1}+|p_i|$이다. $C(T)$에 부분구간 선형 보간법(piecewise linear interpolation)을 적용하여 CUMUL 벡터를 계산한다. 닫힌 세계 시나리오는 Wang14 데이터 세트를 사용해 Tao Wang의 특징 벡터와 분류 성능을 비교했다. 열린 세계 시나리오의 현실성 반영을 위해 120,000개 웹 페이지를 수집하여, 클라이언트가 접근할 수 있는 웹사이트의 수를 늘리면서 실험을 진행했다. 지지 벡터 기계를 이용하여 닫힌 세계 시나리오는 91.3%, 열린 세계 시나리오는 80.0%의 성능을 보였다.

Vera Rimmer 외 5명[10]은 SDAE와 합성곱 신경망(Convolutional Neural Network) 그리고 LSTM(Long Short Term Memory) 모델을 이용해 토르 네트워크 트래픽 데이터의 특징점을 자동으로 학습하는 웹사이트 핑거프린팅 모델을 제안했다. 훈련 데이터 세트는 900개의 웹사이트와 2,500개의 인스턴스로 구성된다. LSTM은 연산 시간의 제약으로 인해 두 모델에 비해 적은 특징을 사용했다. 닫힌 세계 시나리오에 94.2%, 91.7%, 88.0% (SDAE, CNN, LSTM)의 분류 성능을 보고했다.

Arash Habibi Lashkari 외 3명[11]은 토르를 사용하는 애플리케이션(Browsing, Email, File Transfer, VoIP, P2P)을 분류하는 시나리오를 정의했다. 트래픽 수집 시간에 따른 분류 정확도 비교를 위해 10, 15, 30, 60, 120초 간격으로 데이터를 수집했다. ISCXFlowMeter[3]를 사용해 패킷 시간에 관한 특징을 추출해 랜덤 포레스트(Random Forest), 의사 결정 트리(Decision Tree), k-최근접 이웃 알고리즘으로 분류 성능

2) https://www.gnu.org/software/wget/

3) https://github.com/ahlashkari/ISCXFlowMeter

Table 1. Feature Vector vs. Learning Algorithm

Author	Features	No. of Features	Classifier
T. Wang[6]	Tor Cell Sequences	5000	SVM
K. Abe[7]		5000	SDAE
V. Rimmer[10]		5000, 3000, 150	SDAE, CNN, LSTM
A. Panchenko[8]	No. of Incoming & Outgoing Packets, Sum of Incoming & Outgoing Packets, Cumulative Cell Array	104	SVM
A. H. Lashkari[11]	Forward/backward Interval Time, Flow Interval Time, Active, Idle, Bytes per Second, Packets per Second, Duration	23	Random Forest, Decision Tree, *k*-NN
BW, CW_T, CW_H	General Information, Cell Sequence Length, Cell Interval Time, Burst, Cell Ordering, Concentration	125	Random Forest, Extra Trees, XGBoost

을 비교, 분석했다. 15초의 트래픽 수집 시간과 랜덤 포레스트 알고리즘을 사용했을 때 84.3%의 정확도를 보였다.

Table 1은 기 연구된 특징 벡터와 제안하는 특징 벡터 그리고 학습에 사용된 알고리즘을 나타낸다. 지금까지 연구된 웹사이트 핑거프린팅은 토르 셀 시퀀스를 고정 길이 특징 벡터로 변환하기 때문에 트래픽 정보의 일부가 사라지는 단점이 존재한다. 웹사이트 핑거프린팅의 기계학습 모델 응용은 높은 일반화 성능을 내기 위해서 충분한 훈련 데이터 확보와 특징 벡터화가 선행되어야 한다. BW, CW_T, CW_H 데이터 세트는 이 연구에서 수집한 웹사이트 핑거프린팅 데이터로 3장에서 상세히 기술한다.

3. 제안 방법

3.1 트래픽 데이터 수집

데이터 수집 환경은 Ubuntu 16.04 가상머신(virtual machine)에서 토르 브라우저 7.5.4 버전과 WGET을 사용했다. 원활한 데이터 수집을 위해 토르 메트릭의 분석 데이터[1]를 이용하여 넓은 대역폭과 전송 속도가 빠른 입구 노드를 설정했다. Fig. 2는 트래픽 데이터의 수집 구성도를 나타낸다.

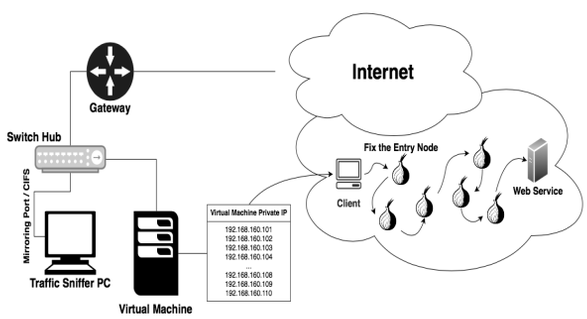

Fig. 2. Tor Traffic Data Collection Environment

토르 브라우저: Tao Wang이 제안한 방법[5]에 따라 웹페이지의 로드가 완료되면 5초, 다음에 수집하는 웹사이트가 상이하면 5초, 동일하면 3초의 시간 간격을 둔다. 10번의 배치(batch)를 두어 한 번의 배치에 20개의 인스턴스를 수집한다. 배치가 종료되면 중간 노드와 출구 노드를 무작위로 재구성한다. 브라우저의 캐시(cache)를 비활성화하여, 동일한 사이트를 재접속할 때 균일한 트래픽을 수집한다.

WGET: torsocks의 wrapper 프로그램인 torify를 사용하면, WGET으로 토르에 접근이 가능하다. 트래픽 수집은 HTML 페이지가 표현 가능한 모든 파일을 다운로드하여 브라우저와 유사한 트래픽을 생성한다.

3.2 데이터 전처리와 제안하는 특징 벡터

훈련 데이터 준비를 위한 전처리 과정은 다음과 같다. 페이로드(payload)의 정보가 없는 패킷은 분류 성능을 감소시키며, 데이터 세트의 크기를 증가시키기 때문에 제거한다[12]. 토르 셀은 512 바이트의 고정 길이로 구성시키기 때문에, TCP 패킷은 토르 셀 패킷으로 변환된다.[4] Fig. 3은 TCP 패킷과 토르 셀 시퀀스의 변환 과정을 나타낸다.

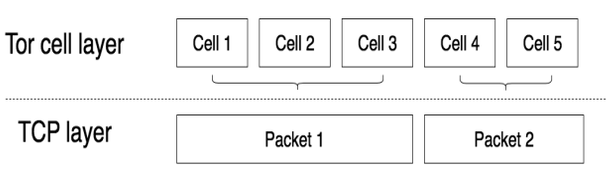

Fig. 3. Packet Transformation for Information Extraction

내부 감시자가 토르 셀 시퀀스에서 관찰 가능한 정보는 셀의 방향, 도착 시간 그리고 순서 정보이다. 클라이언트에서 서버로 전송하는 셀을 송신 셀, 서버에서 클라이언트로 전송되는 셀을 수신 셀로 정의한다. Table 2는 특징 집합과 개수를 나타낸다.

제안하는 특징 벡터는 셀 시퀀스 길이, 도착 시간 간격, 버스트, 셀 순서 정보, 밀집도의 통계치와 범용 정보로 구성된다.

4) TCP packet list [+1514, -609, +2962] → Tor cell list [+512, +512, +512, -512, -512, +512, +512, +512, +512, +512], 양수는 송신 패킷, 음수는 수신 패킷을 의미한다.

Table 2. Feature Type

Feature Type	Number
범용 정보(General Information)	44
셀 시퀀스 길이(Cell Sequence Length)	4
도착 시간 간격(Cell Interval Time)	27
버스트(Burst)	24
셀 순서 정보(Cell Ordering)	18
밀집도(Concentration)	8
Total	125

범용 정보: 일반적인 트래픽에서 추출 가능한 정보로 셀의 개수, 셀의 송/수신 구성 비율, 웹 페이지 로딩 완료 시간 등이다. 초기 30개의 셀은 웹사이트 메인 페이지의 정보를 제공하며 서로 다른 웹사이트 간에 차이가 나타나 보편적 특징으로 사용된다[4].

토르 셀 시퀀스 길이 정보: 전체 셀 시퀀스 길이의 평균과, 표준 편차, 분산, 왜도로 구성된다. 웹사이트 내 콘텐츠의 종류에 따라 송신 셀과 수신 셀의 개수가 달라지기 때문에 전체 셀 시퀀스에 차이를 보인다.

도착 시간 간격 정보: 셀 도착 시간 간격의 정보로 릴레이 노드와 웹 서버의 상태에 따라 차이를 나타낸다.

버스트: 셀의 방향이 바뀌기 전까지 나타나는 연속적인 셀들의 길이를 나타낸다. 셀 시퀀스 “1, -1, -1, -1, 1, 1”의 버스트 패턴은 “1, -3, 2”이다. 버스트는 특정 시간에 집중되는 트래픽의 패턴을 포착한다.

밀집도: HTTP 프로토콜은 데이터를 전송할 때, 청크(chunk) 단위로 송신한다[13]. 밀집도는 송신 셀이 어떤 위치의 청크에 집중되었는지를 나타낸다. 웹 서버마다 사용하는 청크의 단위가 다르기 때문에 본 논문에서는 청크를 20개의 셀이라고 가정한다. 전체 셀 시퀀스를 20개 셀 세그먼트로 구성하고, 세그먼트 안의 송신 셀의 개수를 특징으로 사용한다.

제안하는 특징은 토르 네트워크의 접근 시간에 따라 영향을 받을 수 있다. Fig. 4는 수집한 웹사이트 인스턴스의 평균 RTT(Round Trip Time)를 보여준다. 이 분석으로부터 각 웹사이트의 데이터 수집 시간에 차이가 미미하여 유사한 특징을 추출할 수 있었다.

3.3 훈련 데이터

훈련 데이터는 Wang14, BW, CW_T, CW_H의 4가지 데이터 세트로 구성된다. Table 3은 데이터 세트의 수집 방법과 클래스 그리고 클래스의 인스턴스를 나타낸다.

Wang14는 중국과 영국, 그리고 사우디아라비아에서 금지된 도메인 100개로 구성되었다. 각 웹사이트는 90개의 인스턴스를 갖는다. BW는 클라이언트가 이용한 애플리케이션의 트래픽 데이터이며, 수집 방법(토르 브라우저 또는 WGET)을 구분하는 이진 분류 문제이다. CW_T는 Moz Top 500에서 중복을 제외한 100개 웹사이트의 트래픽 데이터이다. CW_H는 Ahmia.fi에서 총기, 마약, 그리고 불법 저작물과 관련된 히든 서비스 도메인 100개에 대한 트래픽 데이터이다. CW_T, CW_H 데이터 세트는 수집이 불완전하게 종료된 50개의 트래픽을 제거한 150개의 인스턴스로 구성된다.

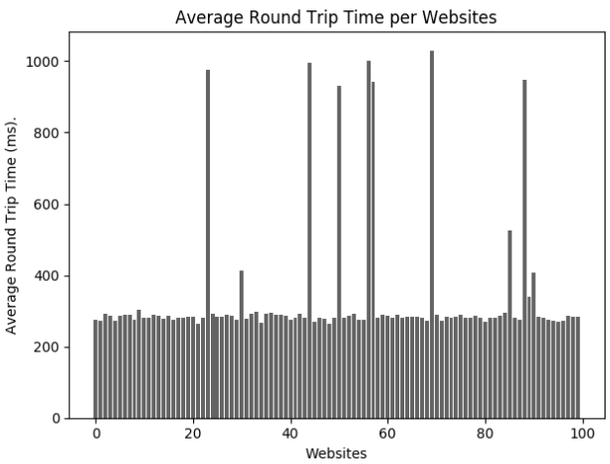

Fig. 4. Average RTT per Website

Table 3. Training Datasets

Dataset	Methods	# of Classes	Size
Wang14	Browser	90	9,000
BW	Browser	2	30,000
	WGET		
CW_T	Browser	100	15,000
	WGET	100	15,000
CW_H	Browser	100	15,000

3.4 학습 알고리즘

랜덤 포레스트는 다수의 독립적인 의사 결정 트리를 학습한다[14]. 훈련 데이터는 배깅(bagging)을 통해 임의로 선택하여 새로운 훈련 데이터를 준비하며, 부분 특징들을 선택하여 의사 결정 트리 분류기를 학습한다. 의사 결정 트리의 노드 분할은 선택한 샘플에 따라 다르며 훈련된 독립된 트리는 전체 훈련 데이터 공간의 일부에서 잘 동작하는 것으로 분석된다. 의사 결정 트리의 수가 증가함에 따라 랜덤 포레스트에서 학습된 모든 분류 트리들은 전체 학습 공간의 일반화 성능을 개선하는데 이용된다. 랜덤 포레스트 학습은 데이터를 반복적으로 샘플링하고 각 부분 집합에 대해 새 트리 분류기를 독립적으로 학습하여 과적합의 영향을 최소화시킨다. 각 테스트 데이터에 대해 랜덤 포레스트의 예측 결과는 다수의 투표로 예측한다.

XGBoost는 Newton 최적화를 사용하여 일련의 회구 트리를 학습하고 연속 트리는 임의의 차등 손실 함수가 제공되는 이전 학습된 트리의 오류를 줄이는 방식으로 학습을 진행한다[15]. 따라서 손실 함수를 사용하려면 해당 음의 기울기를 계산하는 함수 외에 손실 함수를 지정해야 한다. XGBoost 모델을 학습할 때는 적절한 트리 수를 선택하는 것이 중요하다. 실험 분석으로부터 앙상블의 트리 수가 너무 많으면 과적합이 발생하였으며, 너무 낮게 설정하면 과소적합이 보고되었다. 테스트 데이터의 최종 예측은 추가된 각 모델이 손실 함수를 최소화하도록 훈련되기 때문에 학습된 모든 분류기의 출력 합계이다.

엑스트라 트리(Extra trees)는 다양한 앙상블을 생성하기 위해 훈련 과정에 무작위성을 사용한다[16]. 다른 앙상블 모델과 다르게 최상의 임의 노드 분할 및 임계값 선택에 임의성을 이용한다. 랜덤 포레스트와 달리 엑스트라 트리는 부트스트랩(bootstrap) 샘플링을 사용하지 않으며, 임의의 이산화 임의 임계값을 적용하여 의사 결정 경계를 부드럽게 하도록 설계되었다. 일반적으로 엑스트라 트리의 학습에서 높은 편향과 분산을 보이는 현상은 트리 깊이를 적절히 설정하면 이 문제를 해결할 수 있다고 실험을 통해 나타났다.

4. 실험 및 결과

모델 성능 평가 지표로 정확도와 f1-score를 사용하며, 실험은 10식 교차 검증(10-way cross-validation)으로 진행했다. 그리드 탐색(grid search)을 진행하여 최적의 하이퍼파라미터를 선택했다. 제안하는 특징 벡터를 사용한 앙상블 기반 모델(엑스트라 트리, 랜덤 포레스트, XGBoost)과 기 연구된 특징 벡터(CUMUL)를 사용한 지지 벡터 기계(Support Vector Machine)와 학습 성능을 비교, 분석한다.

Table 4. Test Accuracy Comparison for Wang14

Model	90 Instances	60 Instances	40 Instances
Extra Trees	**0.922**	0.914	**0.912**
Random Forest	0.920	0.908	0.902
XGBoost	0.919	0.904	0.888
SVM(CUMUL)	0.922	**0.919**	0.905

Table 4는 Wang14 데이터 세트 중 닫힌 세계 시나리오의 테스트 데이터에 대한 분류 성능 비교이다. 인스턴스의 수를 다양화하여 분류 정확도를 평가했다. 클래스별 90개의 인스턴스를 사용한 경우 엑스트라 트리와 지지 벡터 기계의 분류 성능이 92.2%로 가장 높았다. 60개의 인스턴스만 학습할 때 지지 벡터 기계가 91.9%의 분류 성능을 얻었고 엑스트라 트리 모델이 91.4%로 근소한 차이를 보였다. 가장 적은 인스턴스를 사용했을 경우 엑스트라 트리 모델이 91.2%로 가장 높은 학습 결과를 나타냈다. 이는 연구된 웹사이트 핑거프린팅 방법과 동일한 분류 성능을 가지며 적은 클래스 인스턴스를 가지더라도, 전체 클래스 데이터를 사용한 것과 유사한 성능을 낼 수 있다고 분석된다.

Table 5. Accuracy Comparison for BW

Model	Train	Test
Random Forest	1.0	0.999
Extra Trees	1.0	0.999
XGBoost	1.0	0.999
SVM(CUMUL)	1.0	1.0

Table 5는 BW 데이터 세트에 대한 모델별 학습 성능을 나타낸다. 모든 모델이 브라우저와 WGET 트래픽을 완벽하게 구별한다. WGET 트래픽의 경우 브라우저와 다르게 메인 페이지를 랜더링(rendering)하지 않아 웹페이지 로딩 완료 시간에 차이를 나타낸다. 따라서 내부 감시자는 클라이언트가 어떤 애플리케이션을 이용하는지 구분할 수 있다.

Table 6은 CW_T 데이터 세트의 모델별 학습 성능과 f1-score를 나타낸다. WGET 시나리오는 엑스트라 트리 모델이 96.2%로 분류 성능이 가장 뛰어났으며, 높은 일반화 성능을 나타냈다. 제안하는 방법은 지지 벡터 기계 모델보다 약 5.2%의 성능 향상을 보였다. Browser 시나리오는 엑스트라 트리 모델이 91.7%의 분류 정확도로 지지 벡터 기계 모델보다 약 0.7%의 근소한 성능 향상이 나타났다. 그러나 모든 모델이 낮은 일반화 성능을 보였다.

Table 6. Accuracy and f1-score Comparison for CW_T

Model	Methods	Accuracy		f1-score	
		Train	Test	Train	Test
Random Forest	WGET	0.999	0.961	0.999	0.950
	Browser	1.000	0.916	1.000	0.916
Extra Trees	WGET	0.995	**0.962**	0.995	**0.962**
	Browser	1.000	**0.917**	1.000	**0.919**
XGBoost	WGET	0.999	0.959	0.999	0.959
	Browser	0.996	0.901	0.996	0.903
SVM (CUMUL)	WGET	0.931	0.910	0.931	0.910
	Browser	0.971	0.910	0.972	0.910

Table 7은 히든 서비스 도메인 사이트의 학습 결과를 나타낸다. XGBoost 모델이 64.8%로 가장 높은 분류 성능을 보였으나, 낮은 일반화 성능을 나타냈다. 지지 벡터 기계 모델은 테스트 데이터에 대해 높은 과적합 현상을 보였다. CW_H 데이터 세트의 경우 CW_T 데이터 세트에 비해 낮은 정

Table 7. Accuracy and f1-score Comparison for CW_H

Model	Accuarcy		f1-score	
	Train	Test	Train	Test
Random Forest	0.799	0.624	0.735	0.628
Extra Trees	0.793	0.646	0.722	0.622
XGBoost	0.825	**0.648**	0.822	**0.648**
SVM(CUMUL)	0.977	0.624	0.977	0.621

확도를 보였다. 이는, 히든 서비스 사이트를 접속할 경우 중간 노드의 개수가 많아져 RTT의 지연을 야기해 분류 정확도가 낮은 것으로 분석된다.

실험 분석 결과로부터 CW_H는 낮은 분류 결과를 보였다. 이러한 이유는 유사한 콘텐츠와 단순한 구조를 갖는 불법 히든 서비스 사이트가 존재하기 때문이다. Fig. 5는 유사한 콘텐츠와 구조를 보이는 CW_H 데이터의 예이다.

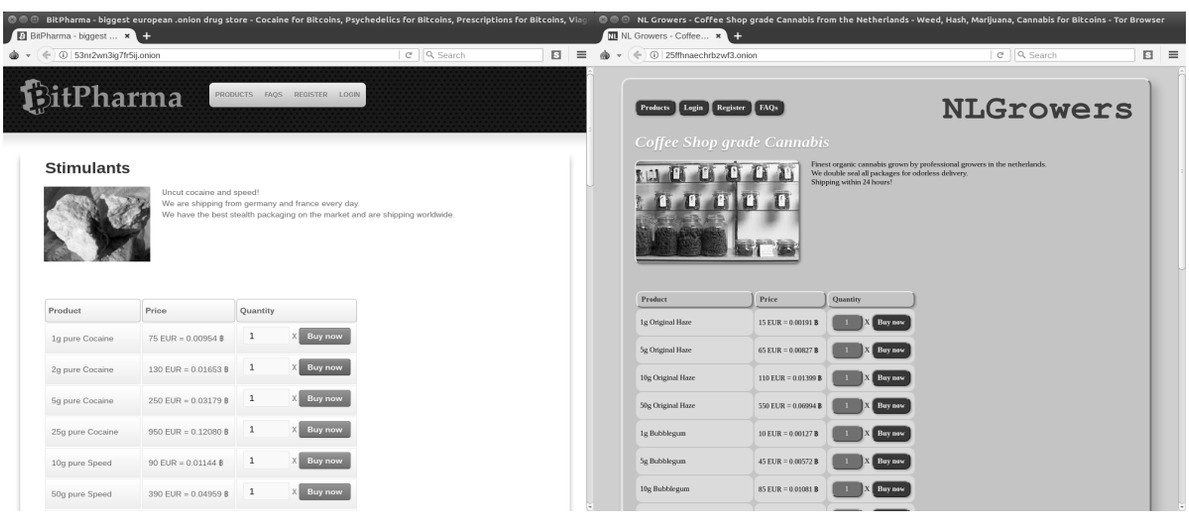

Fig. 5. Two Different Hidden Service Sites

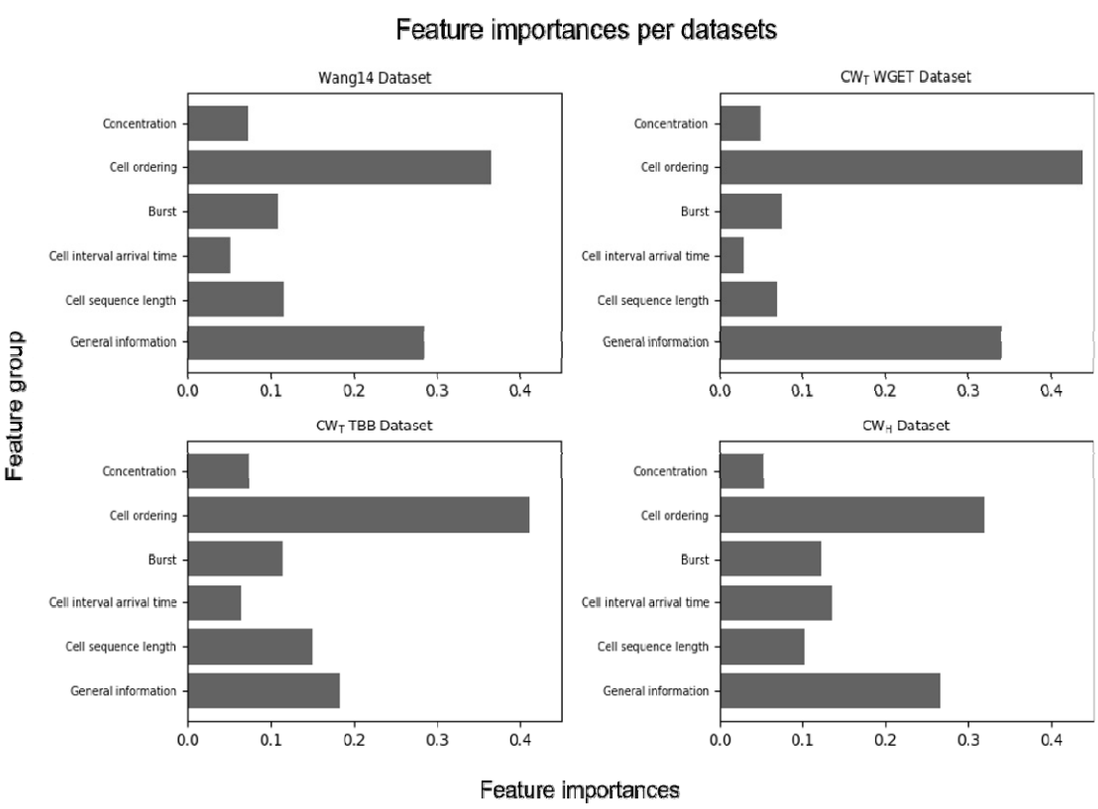

Fig. 6. Feature Importance

Fig. 6은 사용한 데이터 세트의 특징 중요도를 산출한 결과이다. 가장 높은 중요도를 나타낸 특징은 셀 순서 특징이다. 도착 시간 간격 정보는 CW_H 데이터 세트를 제외하고 낮은 중요도를 보인다. 이는 히든 서비스 사이트의 경우 Fig. 5와 같이 비슷한 콘텐츠를 제공하는 사이트가 다수 존재하여 셀 순서 정보, 셀 시퀀스 길이 정보의 중요도가 상대적으로 낮아졌다고 분석되었다.

5. 결 론

본 논문은 실험실 환경에서 토르 브라우저와 WGET을 이용해 일반 최상위 도메인과 히든 서비스 도메인에 대한 트래픽 데이터를 수집하고 앙상블 알고리즘을 이용해 분류 평가를 진행했다. Wang14, BW, CW_T, CW_H 데이터 세트를 통해 제안하는 특징 벡터의 분류 성능을 비교하였다.

제안하는 특징 벡터는 가변 길이의 트래픽 시퀀스에서 범용 정보, 셀 시퀀스 길이 정보, 도착 시간 간격 정보, 버스트, 셀 순서 정보, 밀집도를 추출하고 통계적 모델링을 이용하여 고정 길이의 특징 벡터를 구성했다. Wang14 데이터 세트에서 학습에 사용되는 인스턴스의 수가 적더라도 유사한 결과를 보였고, 기 연구된 모델과 동일한 학습 성능을 나타냈다. 모든 웹사이트 핑거프린팅 모델이 클라이언트가 사용하는 애플리케이션의 유형을 구분했다. 엑스트라 트리 모델은 WGET과 토르 브라우저 시나리오에서 96.2%, 91.7%의 분류 성능을 보였다. XGBoost 모델은 히든 서비스 트래픽에서 64.8%의 정확도를 보였으나, 높은 과적합 현상이 나타났다.

토르에서 일반 최상위 도메인 사이트에 접근할 경우, 연구된 방법과 제안하는 방법 모두 높은 분류율을 보였으나, 히든 서비스 사이트에서 낮은 성능을 보였다. 추후, 히든 서비스 사이트의 분류 성능을 높일 수 있는 새로운 트래픽 특징 표현과 기계학습 응용에 관한 연구가 필요하다.

References

[1] Tor Project Metrics [Internet], https://metrics.torproject.org.

[2] Onion Service Protocol [Internet], https://www.torproject.org.

[3] R. Dingledine, N. Mathewson, and P. Syverson, "Tor: The second-generation onion router," *Usenix Security*, pp. 303-320, 2004.

[4] T. Wang, X. Cai, R. Nithyanand, R. Johnson, and I. Goldberg, "Effective attacks and provable defenses for website fingerprinting," *Proceedings of 23rd USENIX Security Symposium*, pp.143-156, 2014.

[5] M. S. I. Mamun, A. A. Ghorbani, and N. Stakhanova, "An entropy based encrypted traffic classifier," *International Conference on Information and Communications Security*, pp.282-294, 2015.

[6] T. Wang and I. Goldberg, "Improved website fingerprinting on tor," *Proceedings of 12th ACM Workshop on Workshop on Privacy in the Electronic Society*, pp.201-212, 2013.

[7] K. Abe and S. Goto, "Fingerprinting attack on tor anonymity using deep learning," *Proceedings of the Asia-Pacific Advanced Network*, pp.15-20, 2016.

[8] A. Panchenko, F. Lanze, J. Pennekamp, T. Engel, A. Zinnen, M. Henze, and K. Wehrle, "Website Fingerprinting at Internet Scale," NDSS, 2016.

[9] X. Cai, X. C. Zhang, B. Joshi, and R. Johnson, "Touching from a distance: Website fingerprinting attacks and defenses," *Proceedings of the 2012 ACM Conference on Computer and Communications Security*, pp.605-616, 2012.

[10] V. Rimmer, D. Preuveneers, M. Juarez, T. V. Goethem, and W. Joosen, "Automated website fingerprinting through deep learning," *arXiv preprint arXiv*, 2017.

[11] A. H. Lashkari, G. Draper-Gil, M. S. I. Mamun, and A. A. Ghorbani, "Characterization of Tor Traffic using Time based Features," *3rd International Conference on Information Systems Security and Privacy*, pp.253-262, 2017.

[12] A. Pescape, A. Montieri, G. Aceto, and D. Ciuonzo, "Anonymity services tor, i2p, jondonym: Classifying in the dark (web)," *IEEE Transactions on Dependable and Secure Computing*, 2018.

[13] L. Lu, E. C. Chang, and M. C. Chan, "Website fingerprinting and identification using ordered feature sequences," *European Symposium on Research in Computer Security*, pp.199-214, 2010.

[14] L. Breiman, "Random forests," *Machine Learning*, pp.5-32, 2001.

[15] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," *Proceddings of the 22nd acm SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp.785-794, 2016.

[16] P. Geurts, D. Ernst, and L. Wehenkel, "Extremely randomized trees," *Machine Learning*, pp.3-42, 2006.

김 준 호

https://orcid.org/0000-0002-8557-4446

e-mail : 72181469@dankook.ac.kr

2018년 단국대학교 소프트웨어학과(학사)

2018년 ~ 현 재 단국대학교 컴퓨터학과 석사과정

관심분야 : Machine Learning, Deep Learning, Website Fingerprinting,

김 원 겸

https://orcid.org/0000-0003-3022-6230
e-mail : wgkim@aideep.ai
1997년 현대하이닉스(구, LG반도체) 생산기술연구소 주임연구원
2001년 충남대학교 컴퓨터과학과(박사)
2007년 한국전자통신연구원 선임연구원
2009년 마크애니(주) 부장
2016년 (사)한국저작권단체연합회 팀장
2016년 ~ 현 재 (주)에이아이딥 연구소장
관심분야 : 저작권보호, 인공지능(딥러닝), 모바일 보안

황 두 성

https://orcid.org/0000-0003-1840-9296
e-mail : dshwang@dankook.ac.kr
1986년 충남대학교 계산통계학과(학사)
1990년 단국대학교 전자계산학과(석사)
2003년 Wayne State Univ. 컴퓨터학과 (박사)
2003년 ~ 현 재 단국대학교 소프트웨어학과 교수
관심분야 : Machine Learning, Parallel processing, Computer Vision